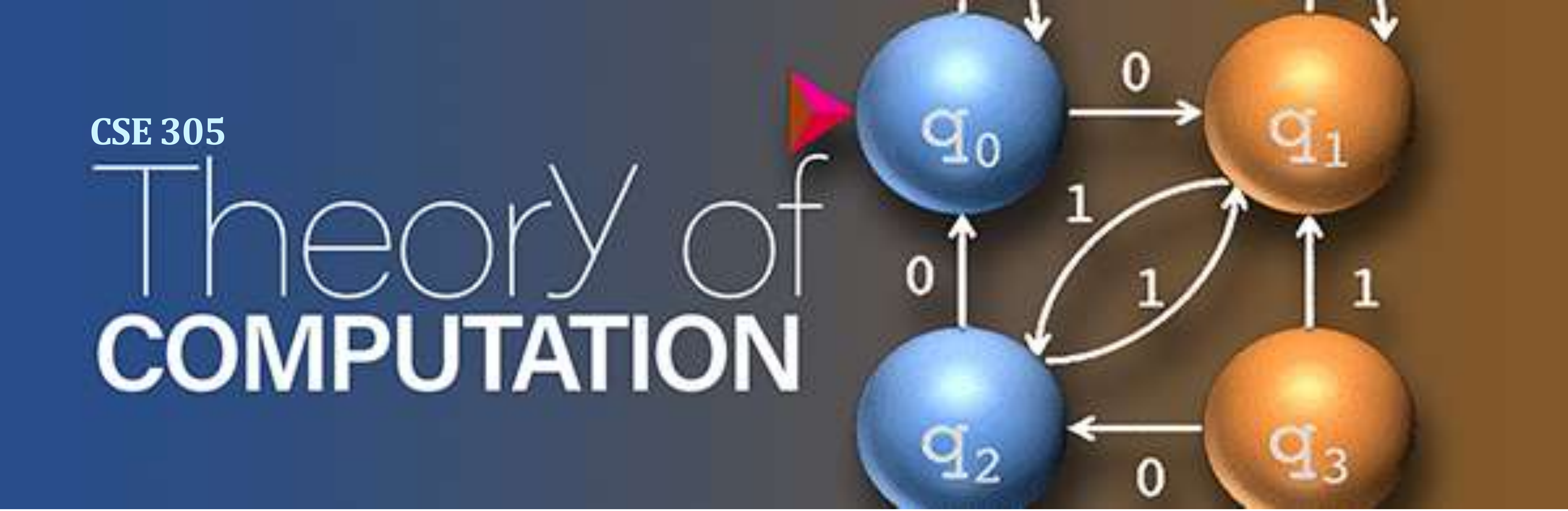


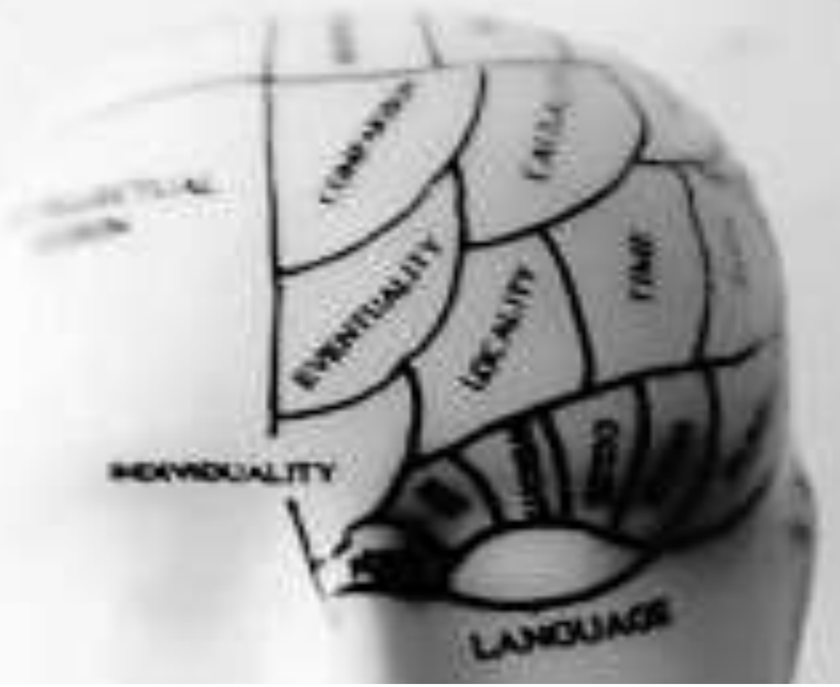
Lecture 15

Finite-State Automata (2)

Md. Mijanur Rahman, Prof. Dr.
Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh.
www.mijanrahman.com

Contents

Finite State Automata



- ❑ Finite State Machine
- ❑ Types of Finite Automata
- ❑ Transition Function, Diagram and Table
- ❑ DFA, NFA, e-NFA with Definitions and Examples
- ❑ **Extended Transition Function**
- ❑ **The Equivalence of DFA's and NFA's**
- ❑ **The Equivalence of NFA's with and without e-moves**
- ❑ **Conversion of e-NFA into NFA (without e)**
- ❑ **Two-way FA**
- ❑ **FA with Output: Moore machine, Mealy machine, Equivalence**
- ❑ **Applications of FA**

DFA (Deterministic Finite Automata)

- DFA refers to deterministic finite automata. Deterministic refers to the uniqueness of the computation. The finite automata are called deterministic finite automata if the machine is read an input string one symbol at a time.
- In DFA, there is only one path for specific input from the current state to the next state.
- DFA does not accept the null move, i.e., the DFA cannot change state without any input character.
- DFA can contain multiple final states. It is used in Lexical Analysis in Compiler.

Formal Definition of DFA

- A DFA is a collection of 5-tuples same as we described in the definition of FA.

Q: finite set of states

∑: finite set of the input symbol

q0: initial state

F: **final** state

δ: Transition function

Transition function can be defined as:

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

Graphical Representation of DFA

- **A DFA can be represented by digraphs called state diagram. In which:**
  1. The state is represented by vertices.
  2. The arc labeled with an input character show the transitions.
  3. The initial state is marked with an arrow.
  4. The final state is denoted by a double circle.
- In the following diagram, we can see that from state q0 for input a, there is only one path which is going to q1. Similarly, from q0, there is only one path for input b going to q2.

Transition Diagram:

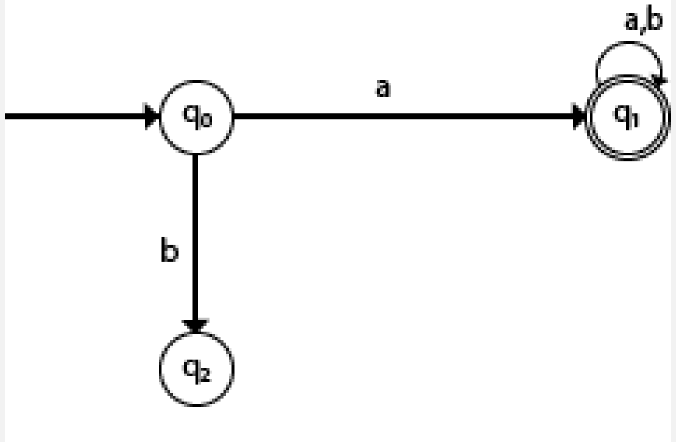


Fig:- DFA

Examples of DFA

Example 1:

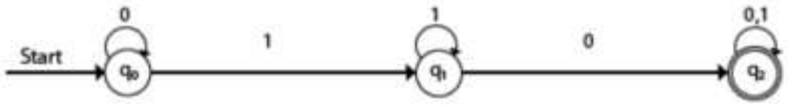
- DFA with ∑ = {0, 1} accepts all strings ending with 10.

Solution:

Q = {q0, q1, q2}
∑ = {0, 1}
q0 = {q0}
F = {q2}

Solution:

Transition Diagram:



Transition Table:

Present State	Next state for Input 0	Next State of Input 1
→q0	q0	q1
q1	q2	q1
*q2	q2	q2

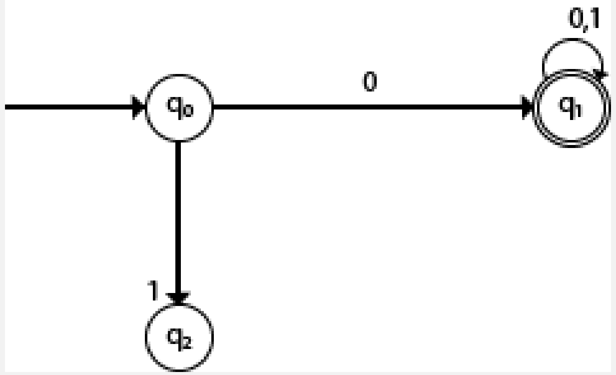
Examples of DFA

Example 2:

- DFA with ∑ = {0, 1} accepts all starting with 0.

Solution:

Transition Diagram:



Transition Table:
?

Explanation:

- In the above diagram, we can see that on given 0 as input to DFA in state q0 the DFA changes state to q1 and always go to final state q1 on starting input 0. It can accept 00, 01, 000, 001....etc. It can't accept any string which starts with 1, because it will never go to final state on a string starting with 1.

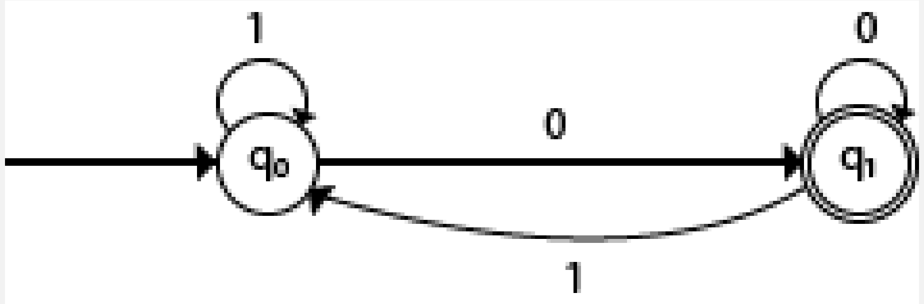
Examples of DFA

Example 3:

- DFA with ∑ = {0, 1} accepts all ending with 0.

Solution:

Transition Diagram:



Transition Table:
?

Explanation:

- In the above diagram, we can see that on given 0 as input to DFA in state q0, the DFA changes state to q1. It can accept any string which ends with 0 like 00, 10, 110, 100....etc. It can't accept any string which ends with 1, because it will never go to the final state q1 on 1 input, so the string ending with 1, will not be accepted or will be rejected.

Examples of DFA

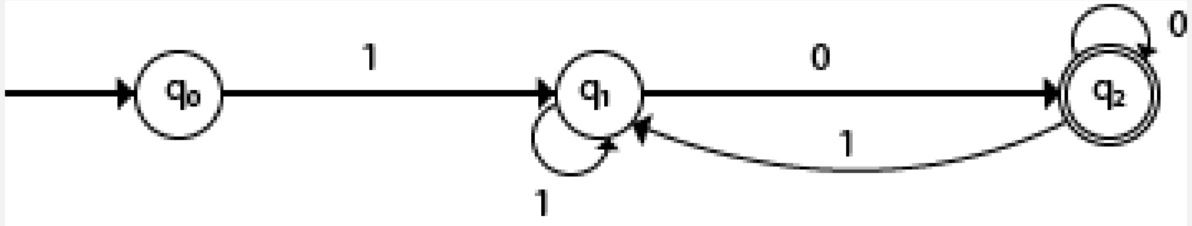
Example 3:

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

Solution:

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

Transition Diagram:



Transition Table:
?

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.

Examples of DFA

Example 4:

- Design a DFA with $\sum = \{0, 1\}$ accepts even number of 0's and even number of 1's.

Transition Table:

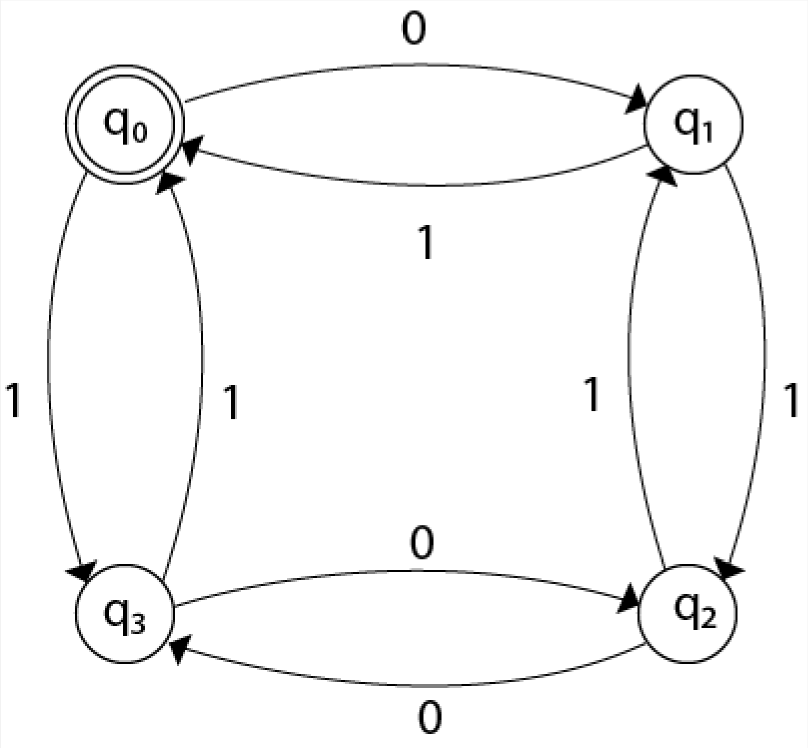
?

Solution:

This DFA will consider four different stages for input 0 and input 1, as shown in the figure.

Here q0 is a start state and the final state also. Note carefully that a symmetry of 0's and 1's is maintained. We can associate meanings to each state as:

- q0: state of even number of 0's and even number of 1's.
  q1: state of odd number of 0's and even number of 1's.
  q2: state of odd number of 0's and odd number of 1's.
  q3: state of even number of 0's and odd number of 1's.



Transition Diagram

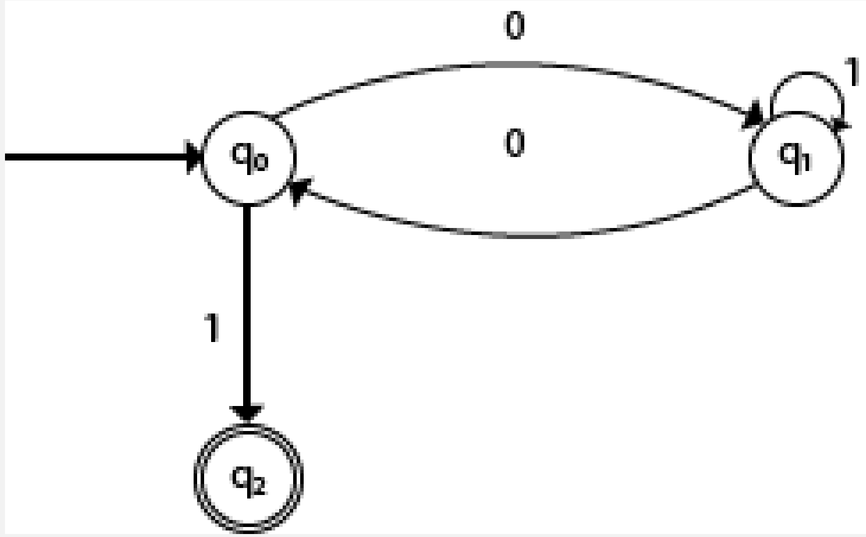
Examples of DFA

Example 5:

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

Solution:

- This DFA can be shown by the following transition diagram:



Transition Diagram

Transition Table:
?

Examples of DFA

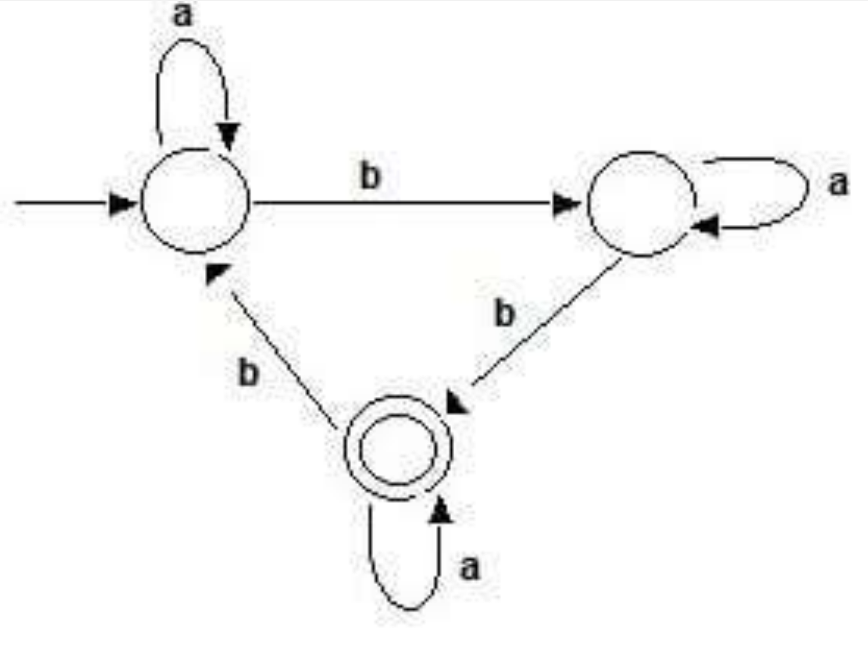
Example 6:

- Design a DFA for the language L = {w ϵ (a,b)*: n_b mod 3 > 1}.

Solution:

- n_b represents the number of b in the string. n_b mod 3 gives the remainder when n_b is divided by 3. n_b mod 3 > 1 implies that the remainder is 2. That means the number of b can be 2, 5, 8, …
- Let M = (Q, Σ, δ, F q_0) be the DFA; Σ = {a, b} is given.
- For the automaton
  Q = {q_0, q_1, q_2}
  F = {q_2}
- This DFA can be shown by the following transition diagram:

Transition Diagram:



Transition Table:
?

How do a DFA Process Strings?

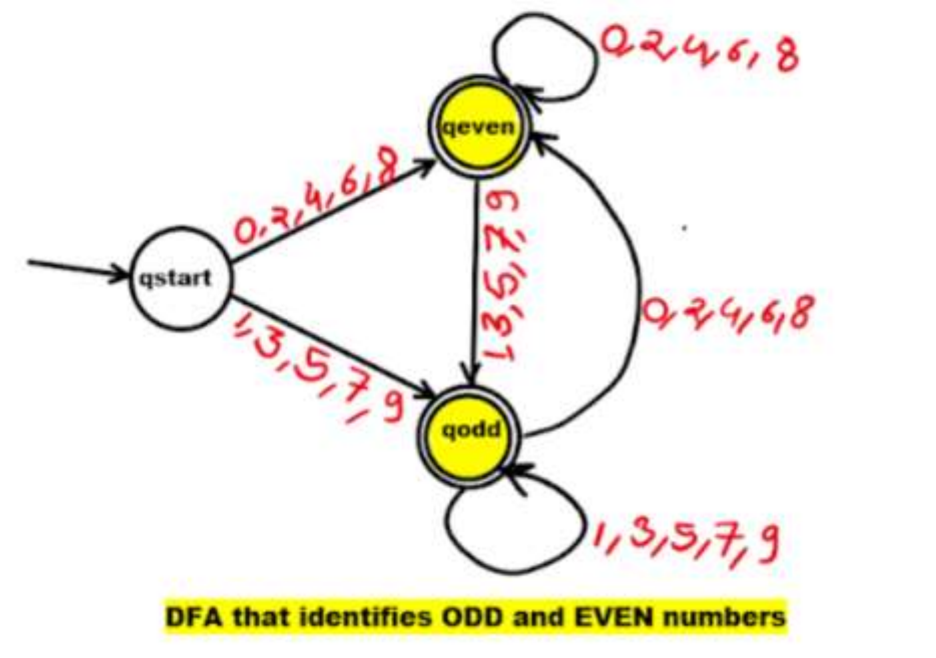
- **The important thing about DFA is to know that it identifies the acceptance of strings.**
- **The language of the DFA is the set of all strings that the DFA accepts.**
- Assume that S1, S2, S3, ….. Sn is a sequence of input symbols. q0 is the starting states of DFA.
- Then first we shall check the transition δ (q0, S1) = q1 where q1 is the state where DFA reaches from q0 by input of S1 where DFA reaches from q0 by input.
- Then we apply δ(qi-1, Si) = qi for each i.
- **If qn ∈ F then the input S1, S2, S3, ….. Sn is accepted otherwise the string is rejected.**

How do a DFA Process Strings?

- **For example,** Consider the DFA that identifies whether the given decimal is even or odd.
- Here we consider 3 states, one start state **qstart**, one even state **qeven** and one odd state **qodd**.

1. If the machine stops at **qeven** the given number is even.
2. If it stops at **qodd** the given number is odd.

Transition Diagram:



Transition Table:

State	Input Symbol	
	0, 2, 4, 6, 8	1, 3, 5, 7, 9
qstart	qeven	qodd
qeven	qeven	qodd
qodd	qeven	qodd

THE END