Finite Automata (FA)

Lecture 3.2

Strathmore
UNIVERSITY

Lecture Outline

- This lecture covers:
  - Finite automata
  - Finite automata classes
  - Deterministic finite automata
  - Nondeterministic finite automata
  - Finite automata formal definition
  - Applications of finite automata

Finite Automata

- Finite Automata, FA (also called Finite State Machine) is a model of computation with **finite set of states** and which acts as a ***language acceptor***
- FA state can change from one to another upon receiving some input symbol
- FA has fixed memory capacity to handle information and does not have **auxiliary memory**

Finite Automata (II)

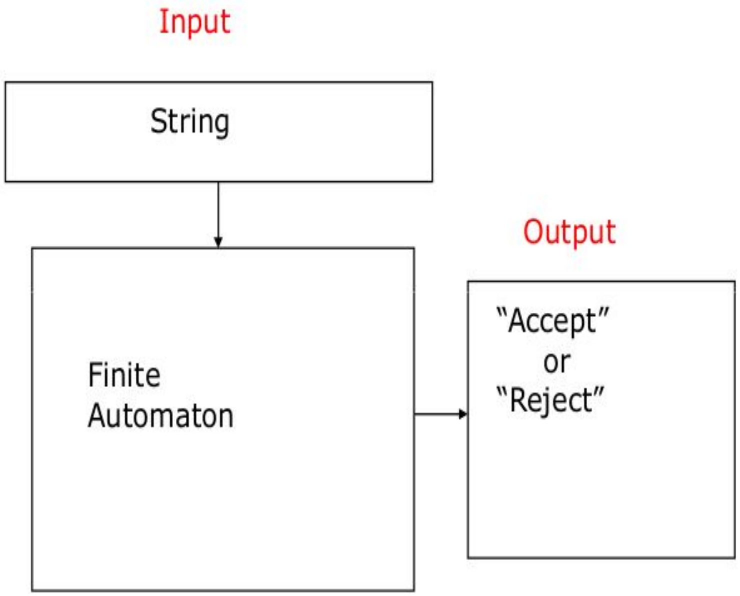
- When FA receives a string input, it can either accept or reject it hence generally regarded as ***language acceptors*** or **language recognition device**
- Any computer whose outputs are either “yes” or “no” acts as a language acceptor
- The language the computer accepts is the set of input strings that cause it to produce the answer yes

Finite Automata (III)

- A language is called a **regular language** if some finite automaton recognizes it
- Due to memory limitation, FA's can only accept **simple languages** which does not require ***remembering more than the current state***

Finite Automata (IV)

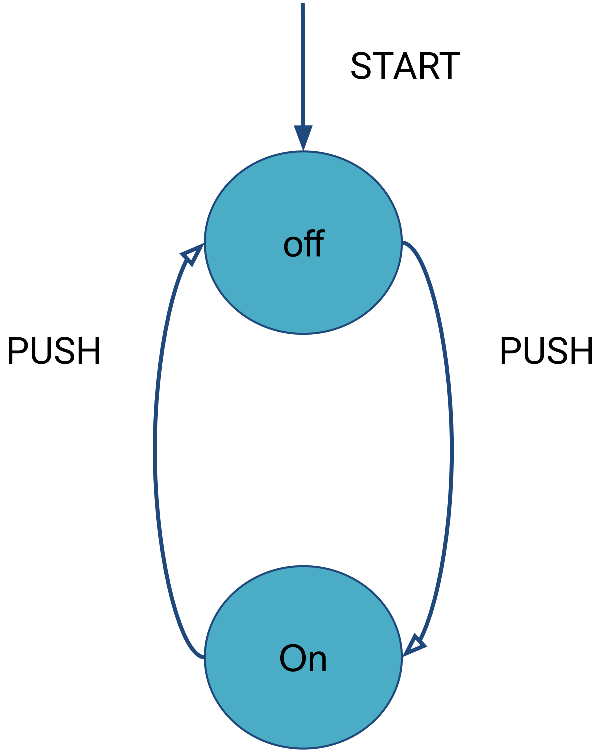
- The machine reads the input string one at a time until the end of string
- If it winds up in one of the set of **final states** the input string is considered to be accepted otherwise **rejected**
- The **language accepted** by the machine is the set of strings it accepts



FA Basic Design

Finite Automata (V)

- FA is the simplest computational model, with very limited memory, but highly useful
- Example: *a finite automaton modeling an on/off switch*



Finite Automata Classifications

- Finite automata can be classified into two major categories
  1. Deterministic Finite Automata (DFA)
  2. Nondeterministic Finite Automata (NFA)

Deterministic Finite Automata (DFA)

Formal Definition of DFA

Strathmore UNIVERSITY

- DFA is defined by a 5 -tuple:

  $(Q, \Sigma, \delta, q_0, F)$

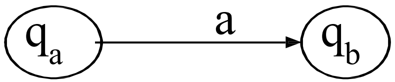
1. Q: finite set of states
2. Σ: finite alphabet
3. δ: transition function, $\delta : Q \times \Sigma \rightarrow Q$, takes a state and input symbol as arguments, and returns a state
4. $q_0 \in Q$: start state
5. $F \in Q$: set of accept states

Finite State Diagram

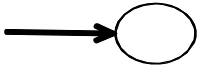
- A graphic representation of a finite automaton
- A finite state diagram is a directed graph, where **nodes represent elements** in Q (i.e., states) and **arrows are characters in Σ** such that:

Finite State Diagram (II)

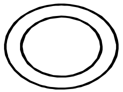
- $((q_a,a),q_b)$ is a transition in δ:



- The initial state is marked with:



- The final state(s) are marked with:



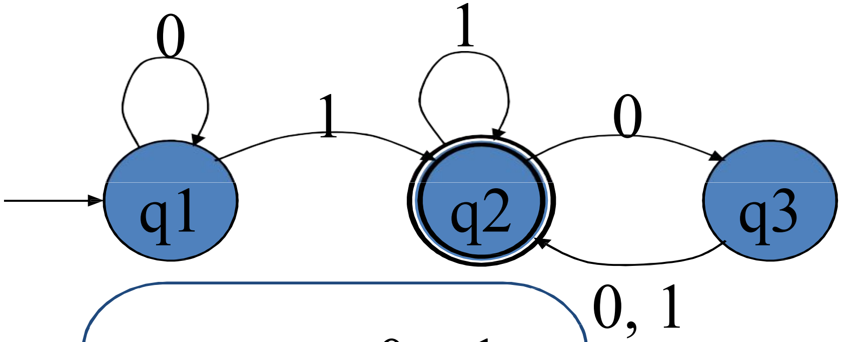
Formal **Description** of DFA - Example

Strathmore UNIVERSITY

FORMAL DESCRIPTION

- $M = (Q, \Sigma, \delta, q_0, F)$, where
  - $Q = \{q1, q2, q3\}$
  - $\Sigma = \{0, 1\}$
  - $\delta(q1,0)=q1$, $\delta(q1,1) = q2$, $\delta(q2,0) = q3$, $\delta(q2,1) = q2$, $\delta(q3,0) = q2$, $\delta(q3,1) = q2$
  - q1 is the start state
  - $F = \{q2\}$

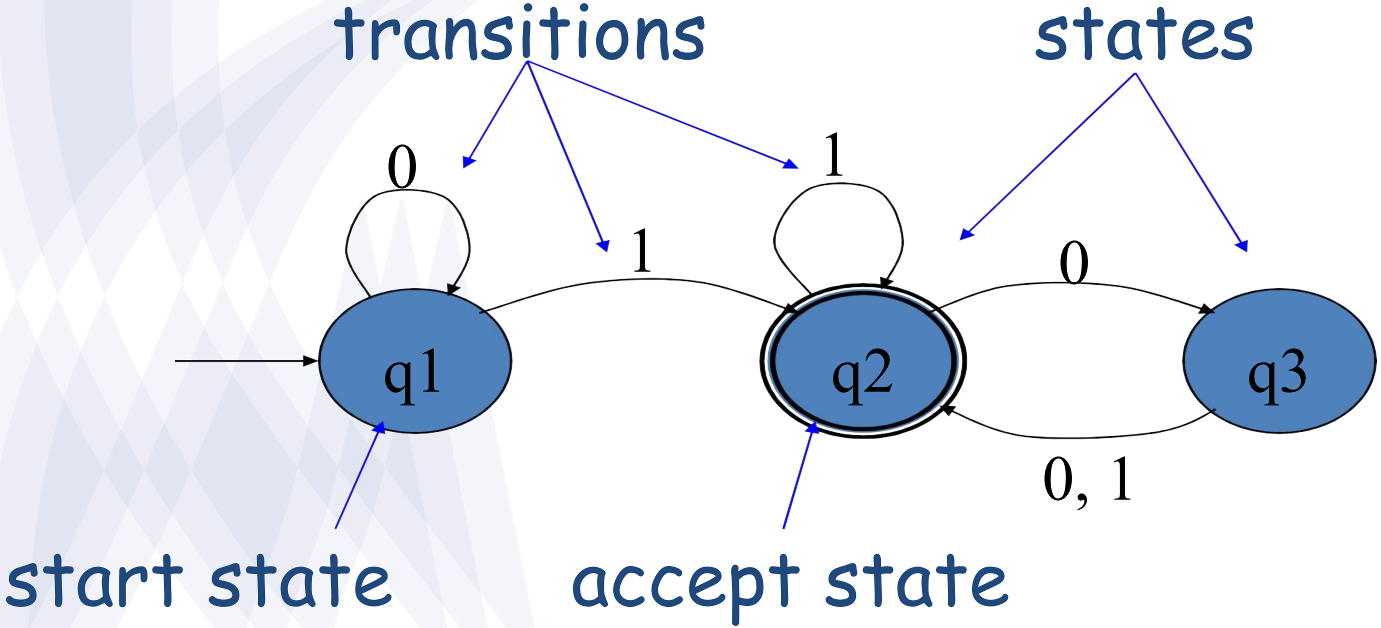
STATE TRANSITION DIAGRAM



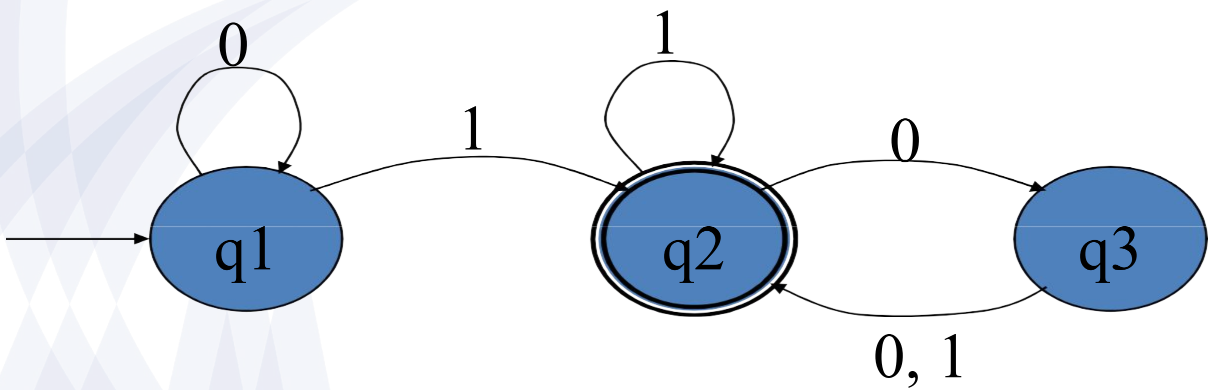
	0	1
q1	q1	q2
q2	q3	q2
q3	q2	q2

STATE TRANSITION TABLE

Finite State Diagram(III)



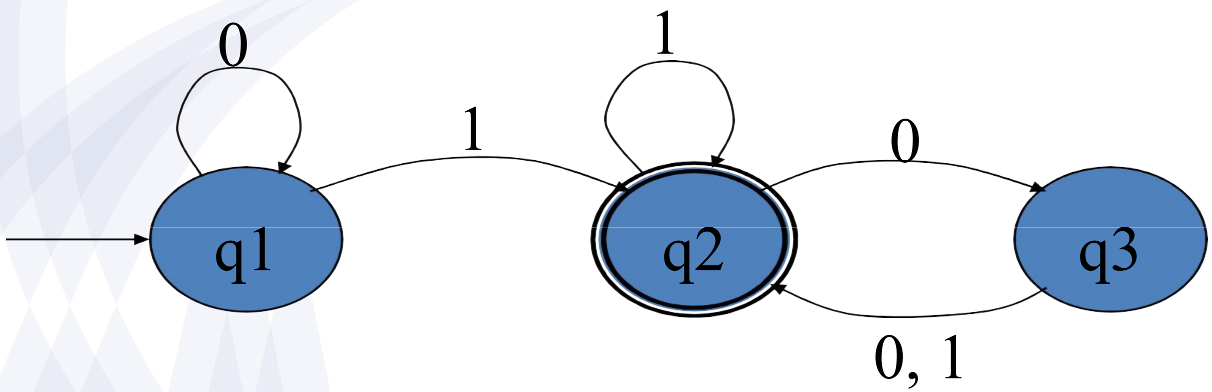
Finite State Diagram(III) - Example 1



- on input “0110”, the machine goes:

  q1→q2→q2→q3 = “reject”

Finite State Diagram(IV) - Example 2

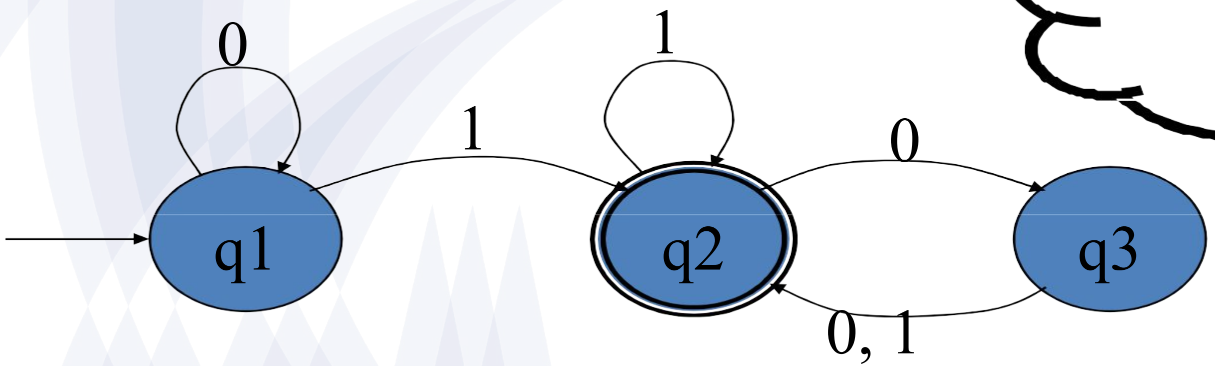


on input “101”, the machine goes:

q1 → q2 → q3 → q2 = “accept”

Finite State Diagram(V) - Example 3

What is the language accepted by this Machine?



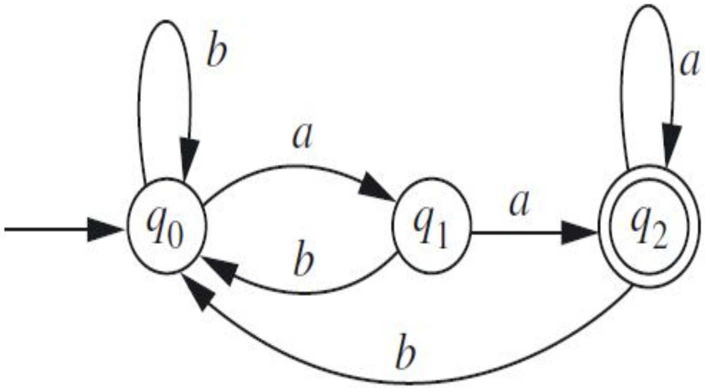
010:	reject
11:	accept
0110100:	accept
010000010010:	reject

Finite State Diagram(V) - Example 4

- Given language L_1 =

$\{x \in \{a, b\}^* \mid x \text{ ends with aa}\}$

- DFA to recognise L_1 (accepting the strings ending with aa)
- What is the transition from start to accept state on
  - aabbaa
  - ababaa
  - aaaaaa

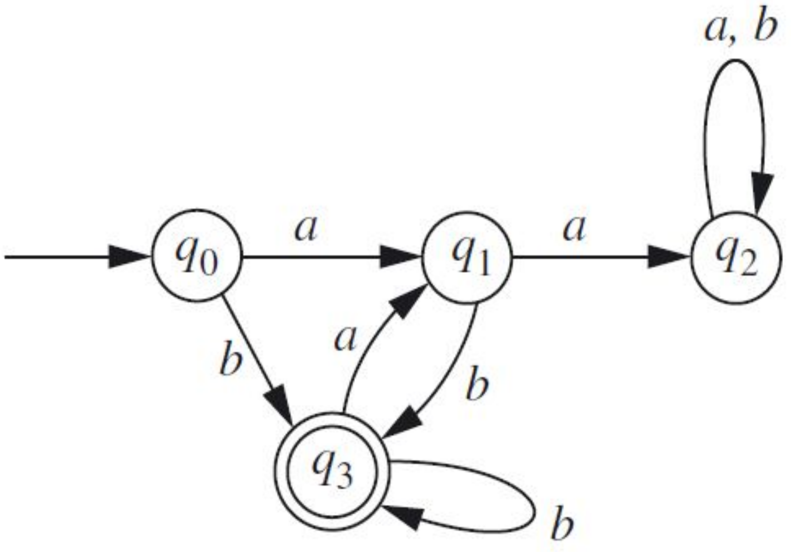


Finite State Diagram(V) - Example 6

- Given language L_2 =

$\{x \in \{a, b\}^* \mid x$ ends with b and does not contain the substring aa$\}$

- What is the transition from start to accept state on
  - abab
  - abbbbab



Finite State Machine Language Acceptance

- If A is the set of all strings that finite automaton machine **M** accepts, we say that A is the language of machine M and write **L(M) = A.**
- We can also say that *M* ***recognizes*** *A or that M* ***accepts*** *A*
- A machine may accept several strings, but it always *recognizes only one language.*
- If the machine accepts no strings, it still recognizes one language namely, the empty language $\varnothing$

Class Exercise

Strathmore UNIVERSITY

- SU cafeteria has installed an ice cream vending machine to automatically dispense ice cream to students and staff. The cost of a can of ice cream is Kshs 60 and the machine only accepts coins in the denomination of 20 & 40 only and the machine does not give change.

Exercise (II)

a) Formally define this machine as a finite automaton. i.e. in terms of ($Q, \Sigma, \delta, q_0, F$)

b) Draw the state transition diagram for the machine defined in (a) above

c) Draw the state transition table for the machine defined in (a) and (b) above