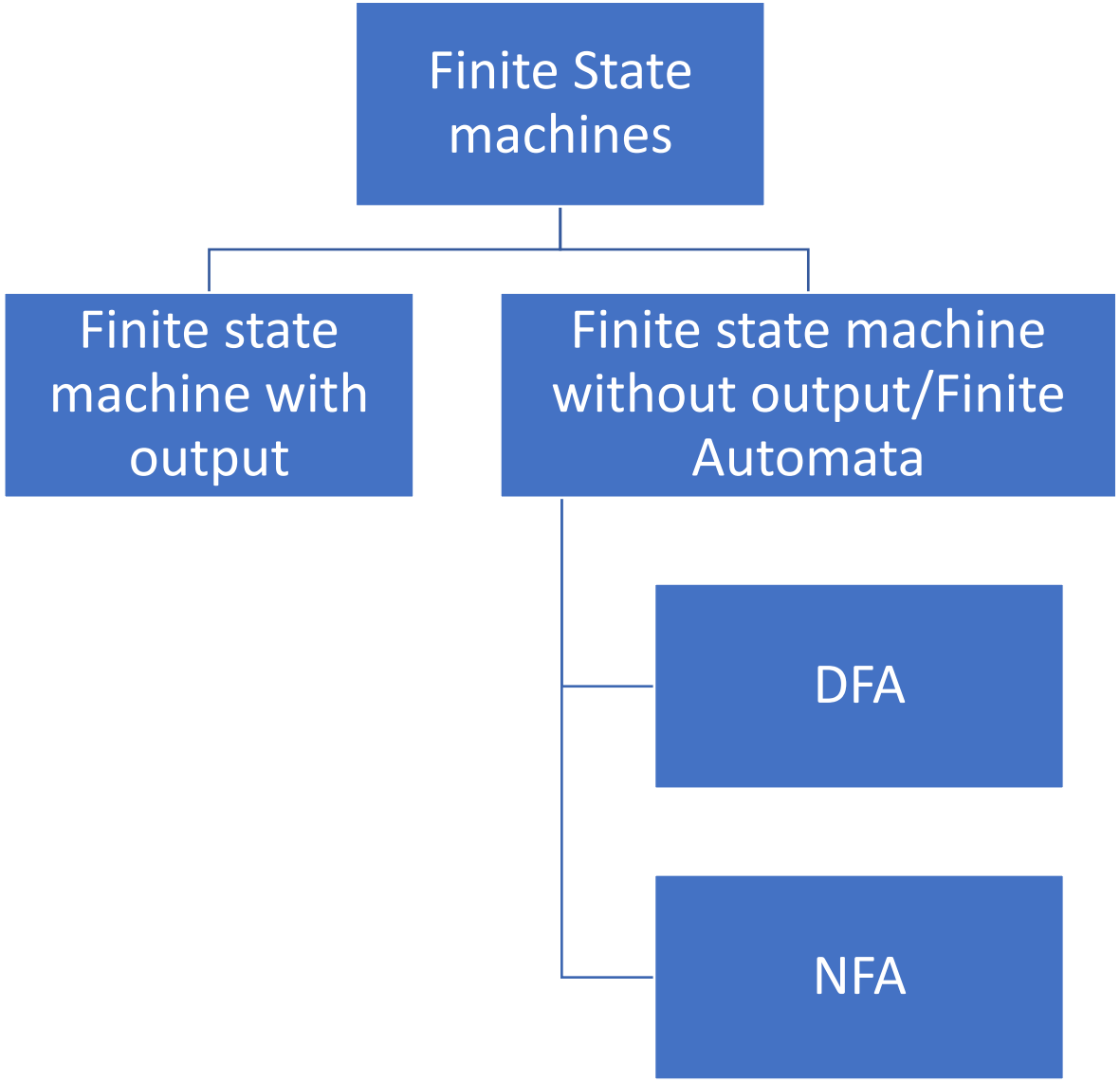
Unit VI: Language and grammar

FINITE STATE MACHINES

- **FSM = Transducer = Finite automata with output = Finite automata + Output Capability”**



Theory of Automata

- Theory of automata is a theoretical branch of computer science and mathematical.
- It is the study of abstract machines and the computation problems that can be solved using these machines.
- The abstract machine is called the automata.
- The main motivation behind developing the automata theory was to develop methods to describe and analyse the dynamic behaviour of discrete systems.

Type of automata

- There are four specific areas or types of automaton. They include the following:
  - **Finite State Machine** – Finite state machines are computation models that are ideal for a small amount of memory. They also don't maintain memory. The primary application is in regards to mathematical problem analysis. A finite state machine is considered the simplest of all types of automata.
  - **Pushdown Machine** – These are finite state machines that are augmented with extra memory in what is called a stack. New elements are pushed into the stack. These machines can do more than a finite machine, but not as much as a Turing machine.
  - **Turing Machine** – This type of automata is the most advanced. It is capable of changing symbols and simulating computer storage and execution. High performance computing, machine learning, and software testing are examples that involve the complexity of Turing machines. Turing machines were one of the first foundational models that led to what is modern computer science.
  - **Linear Bounded** – This is considered a restricted type of Turing machine. Computation is restricted and operates within a finite, bounded area.

- This automaton consists of states and transitions. The State is represented by circles, and the Transitions is represented by arrows.
- Automata is the kind of machine which takes some string as input and this input goes through a finite number of states and may enter in the final state.

Finite State Machine with Output

- A finite-state machine M = (S, I, O, f, g, s_0) consists of a
  - finite set S of states,
  - a finite input alphabet I,
  - a finite output alphabet O,
  - a transition function f that assigns to each state and input pair a new state,
  - an output function g that assigns to each state and input pair an output, and
  - an initial state s_0

Representation

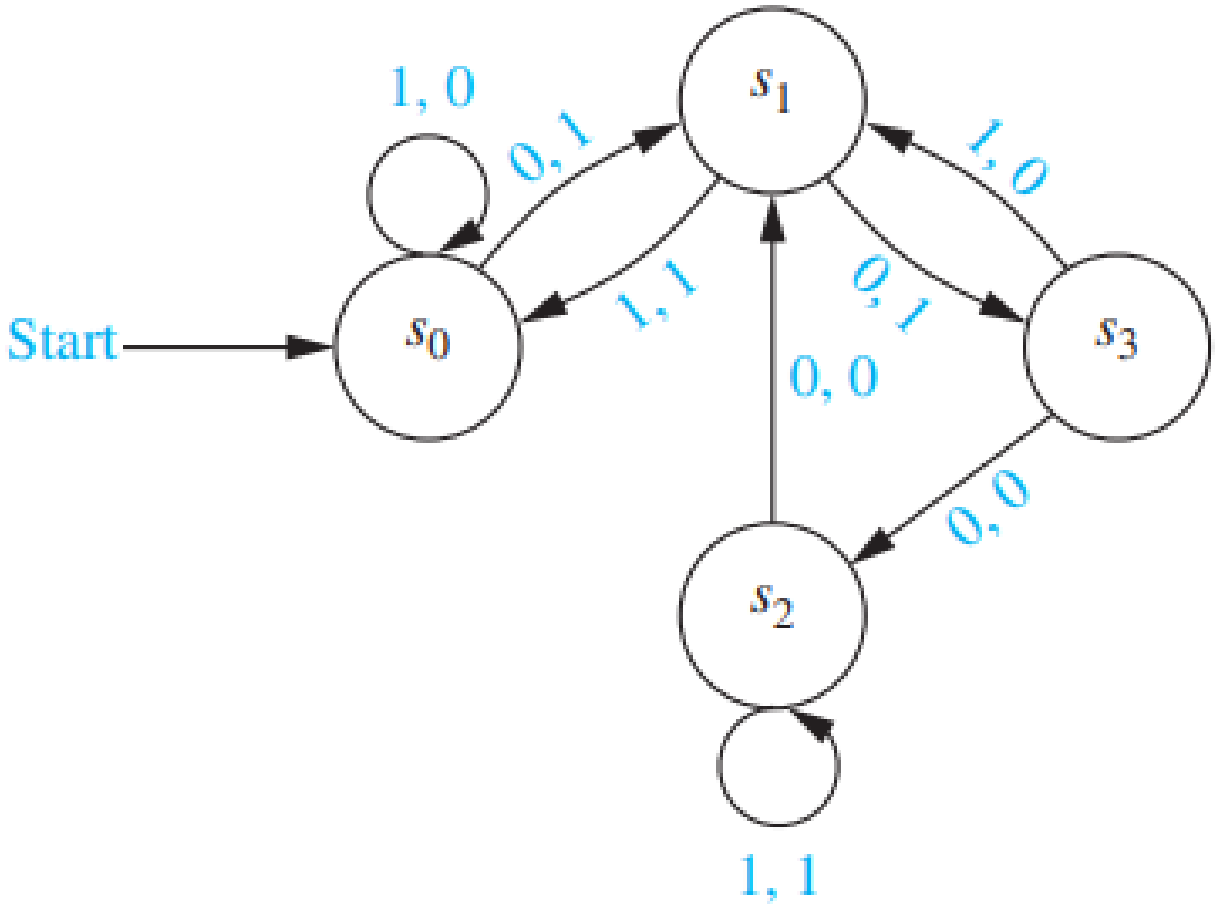
- State table
  - represent the values of the transition function f and the output function g for all pairs of states and input
- State diagram
  - a directed graph with labeled edges. In this diagram, each state is represented by a circle. Arrows labeled with the input and output pair are shown for each transition

Example 1

- Construct state diagram for state table provided below

TABLE 2

State	f Input 0	f Input 1	g Input 0	g Input 1
s_0	s_1	s_0	1	0
s_1	s_3	s_0	1	1
s_2	s_1	s_2	0	1
s_3	s_2	s_1	0	0



Example 2

- Construct state table for state diagram given below
- Find output string generated if input string is 101011

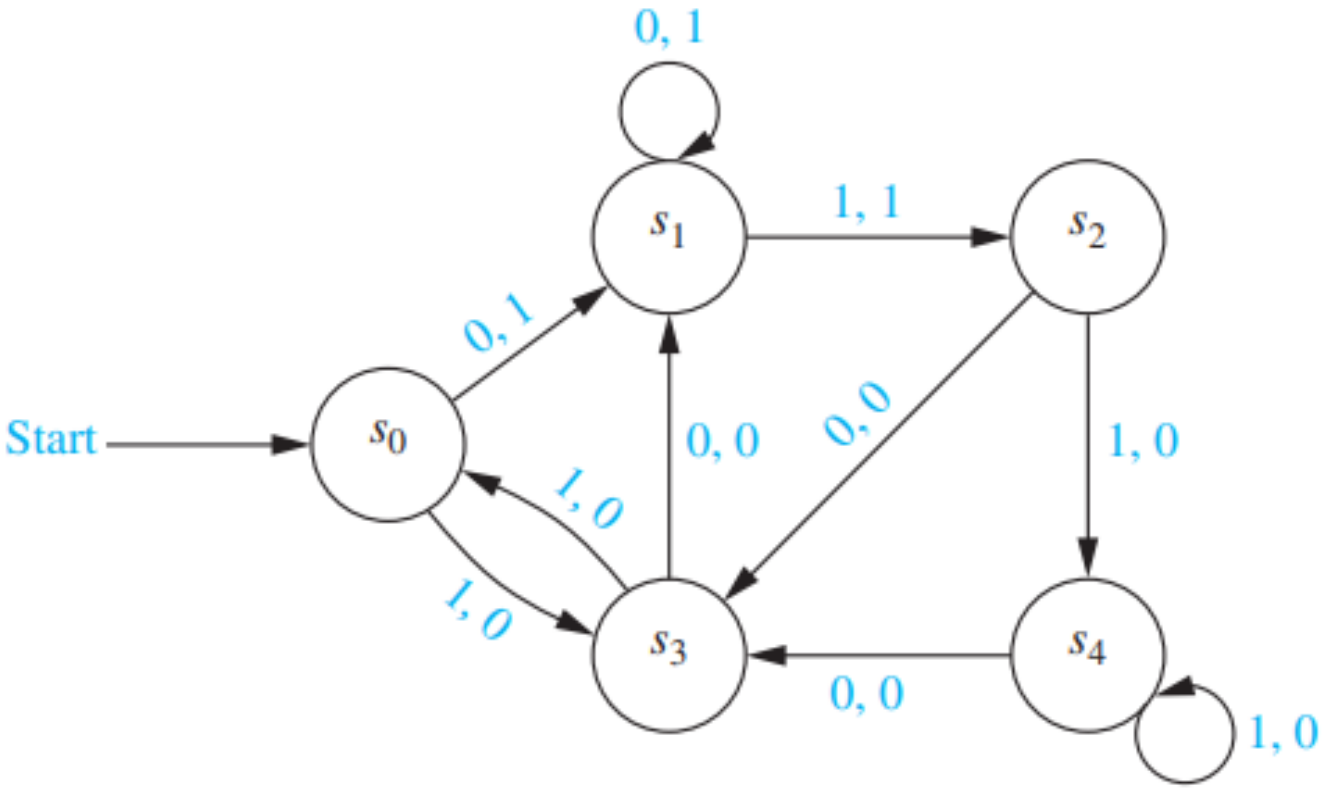
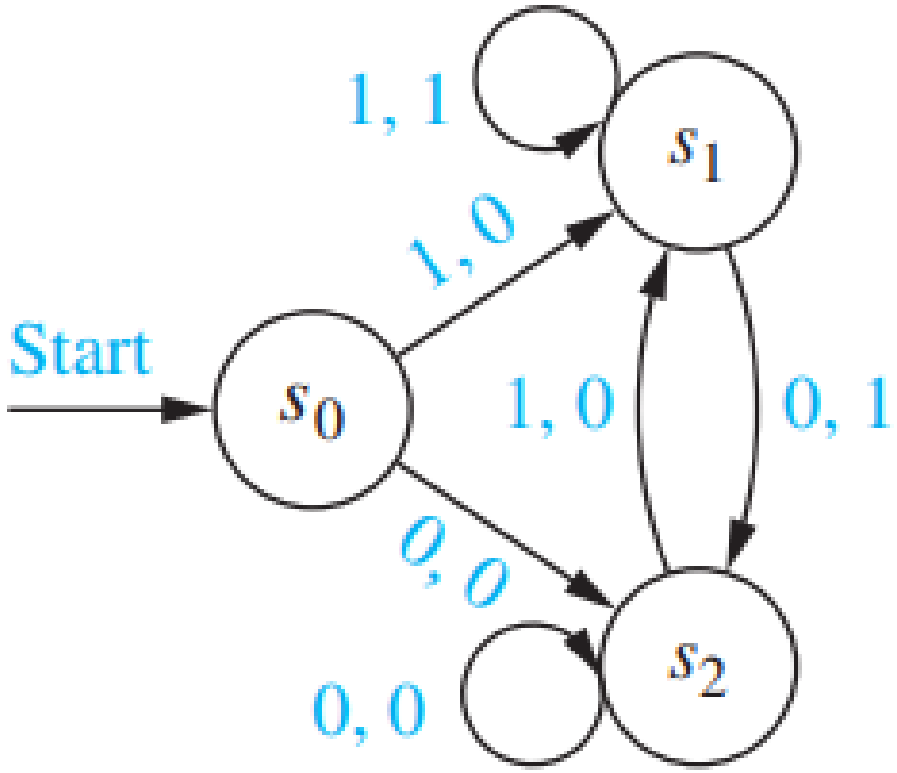


TABLE 3

State	f Input 0	f Input 1	g Input 0	g Input 1
s_0	s_1	s_3	1	0
s_1	s_1	s_2	1	1
s_2	s_3	s_4	0	0
s_3	s_1	s_0	0	0
s_4	s_3	s_4	0	0

Example 3

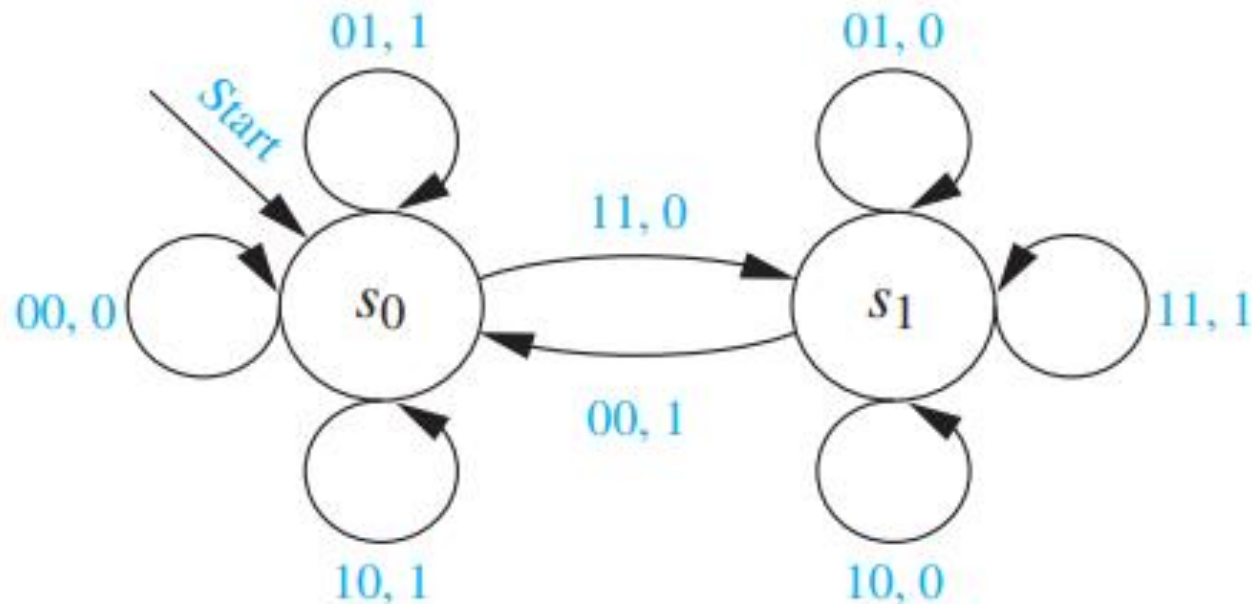
- A delay machine can be constructed that has two possible inputs, namely, 0 and 1. The machine must have a start state s_0



Example 4

- Produce a finite-state machine that adds two positive integers using their binary expansions
  - When $(x_n \ldots x_1 x_0)$ and $(y_n \ldots y_1 y_0)$ are added, the following procedure is followed.
  - First, the bits x_0 and y_0 are added, producing a sum bit z_0 and a carry bit c_0. This carry bit is either 0 or 1. Then, the bits x_1 and y_1 are added, together with the carry c_0. This gives a sum bit z_1 and a carry bit c_1. This procedure is continued until the nth stage, where x_n, y_n, and the previous carry c_{n-1} are added to produce the sum bit z_n and the carry bit c_n, which is equal to the sum bit z_{n+1} .
  - Assume Initial x_n and y_n are zero

- The inputs to the machine are pairs of bits, there are four possible inputs. We represent these possibilities by 00 (when both bits are 0), 01 (when the first bit is 0 and the second is 1), 10 (when the first bit is 1 and the second is 0), and 11 (when both bits are 1). The transitions and the outputs are constructed from the sum of the two bits represented by the input and the carry represented by the state. For instance, when the machine is in state s1 and receives 01 as input, the next state is s1 and the output is 0, because the sum that arises is $0 + 1 + 1 = (10)2$


01, 1
01, 0
Start
11, 0
00, 0
s_0
s_1
11, 1
00, 1
10, 1
10, 0

- Finite state machines
  - Mealy machines
    - Outputs correspond to transition between states
  - Moore machines
    - Output is determined only by state

Finite state machine without output/Finite state automata

- Finite state machine without output is called finite state automata
- Recognize a string if and only if it takes start state to final state
- We can represent finite-state automata using either state tables or state diagrams. Final states are indicated in state diagrams by using double circles.

Finite Automata

- 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.

Finite Automata

- Finite automata can be represented by input tape and finite control and read only head
- **Input tape**: It is a linear tape having some number of cells. Each input symbol is placed in each cell.
- **Read only head**: The tape has read only head that examines one square at time and can move one square either to left or to right. At the beginning of operation, the head is always at leftmost square of the input tape.
- **Finite control**: The finite control decides the next state on receiving particular input from input tape. The tape reader reads the cells one by one from left to right, and at a time only one input symbol is read.

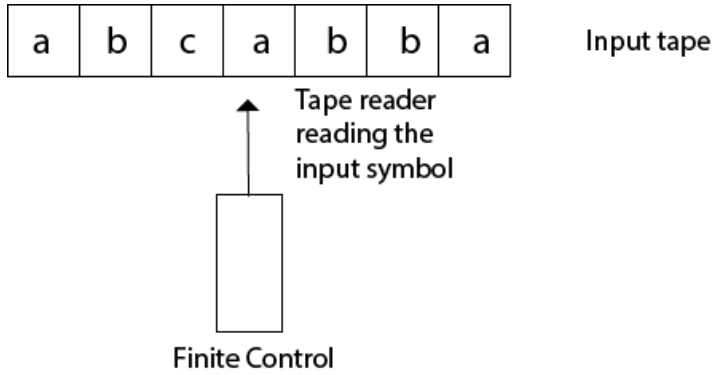
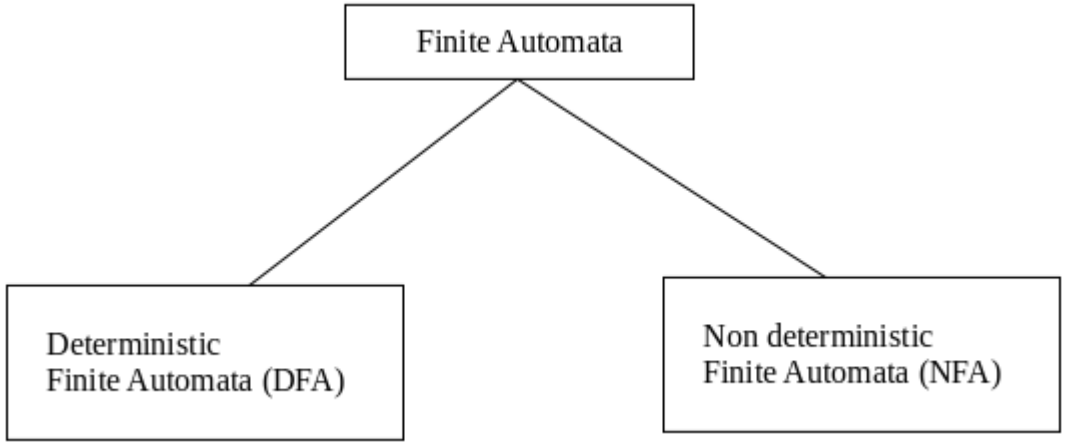


Fig :- Finite automata model

Finite Automata

- DFA refers to deterministic finite automata. Deterministic refers to the uniqueness of the computation. In the DFA, the machine goes to one state only for a particular input character. DFA does not accept the null move
- NFA stands for non-deterministic finite automata. It is used to transmit any number of states for a particular input. It can accept the null move

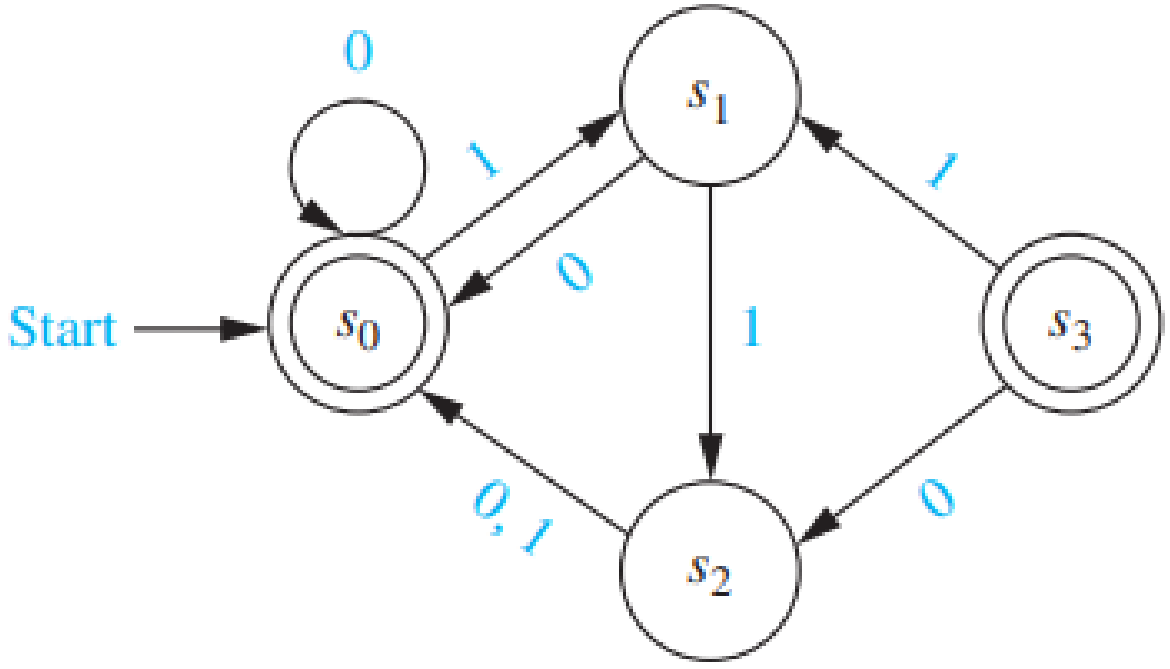


Deterministic Finite state automata

- A finite-state automaton M = (S, I, f, s_0,F) consists of a
  - finite set S of states,
  - a finite input alphabet I,
  - a transition function f that assigns a next state to every pair of state and input (so that f : S × I → S),
  - an initial or start state s_0, and
  - a subset F of S consisting of final (or accepting states)

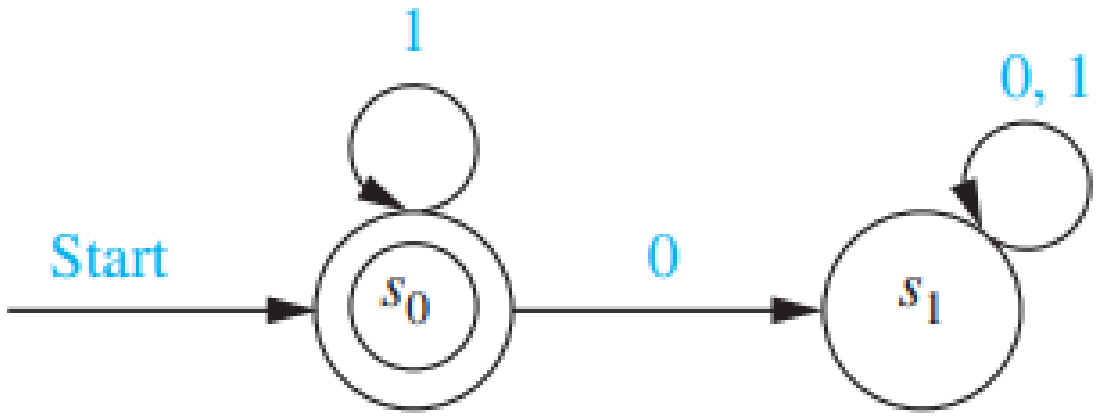
Example

State	f	
	Input	
	0	**1**
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1



Example

- Determine language recognized by finite state automata.

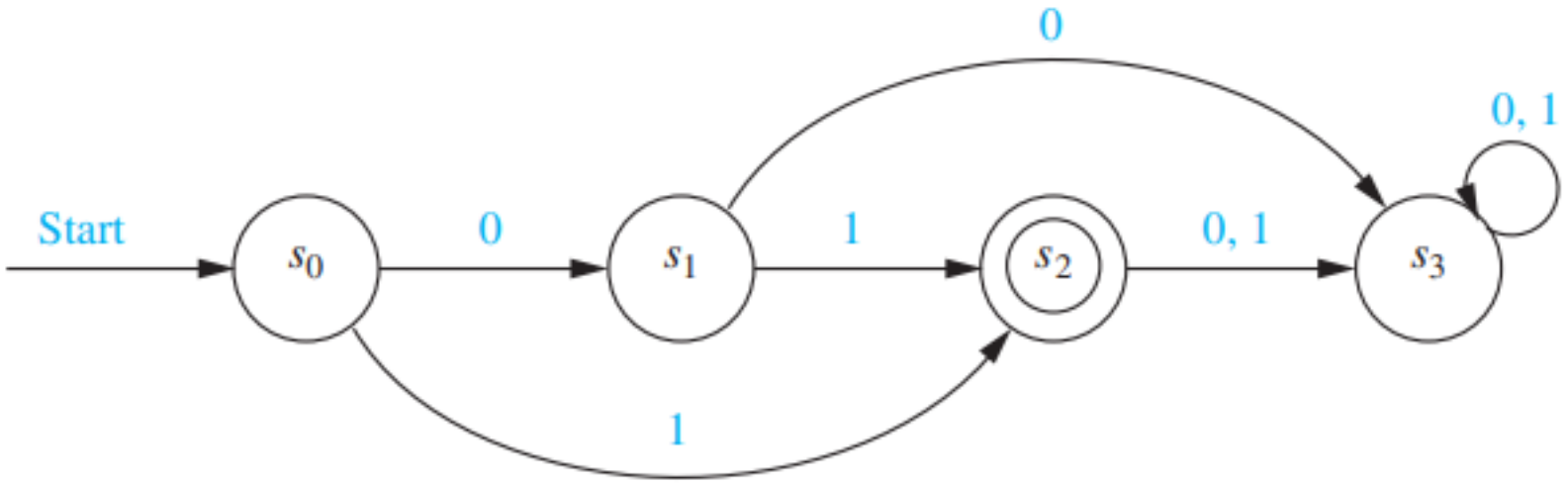


M_1

- Solution: The only final state of M_1 is s_0. The strings that take s_0 to itself are those consisting of zero or more consecutive 1s. Hence, $L(M_1) = \{1^n \mid n = 0, 1, 2,...\}$.

Example

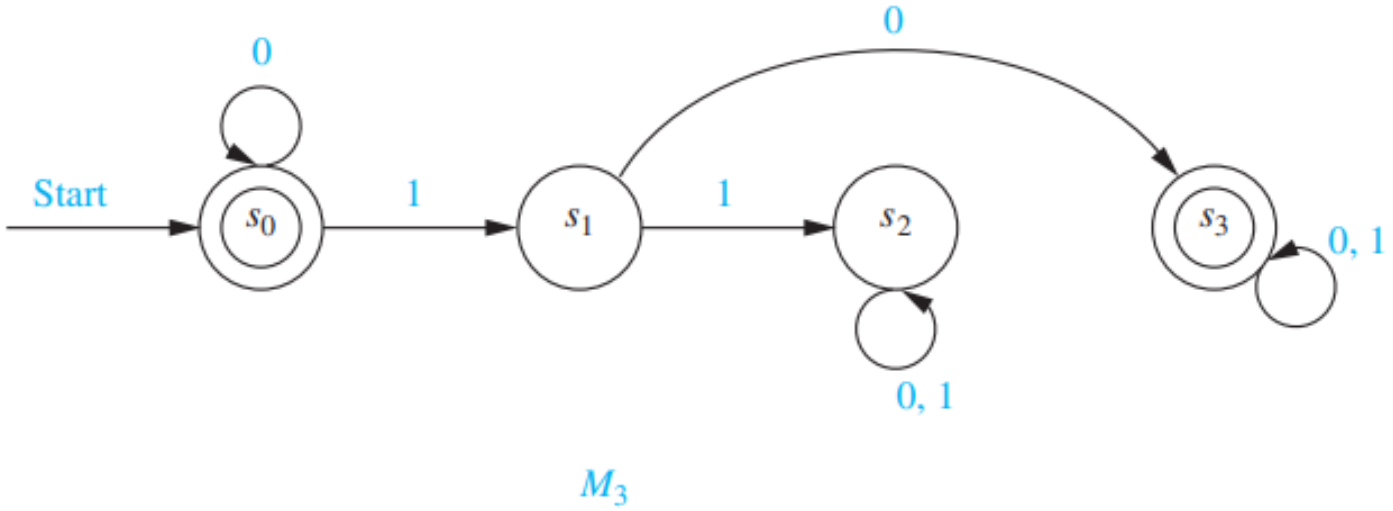
- Determine language recognized by finite state automata.



  - Solution: The only final state of M_2 is s_2. The only strings that take s_0 to s_2 are 1 and 01. Hence, $L(M_2) = \{1, 01\}$

Example

- Determine language recognized by finite state automata.



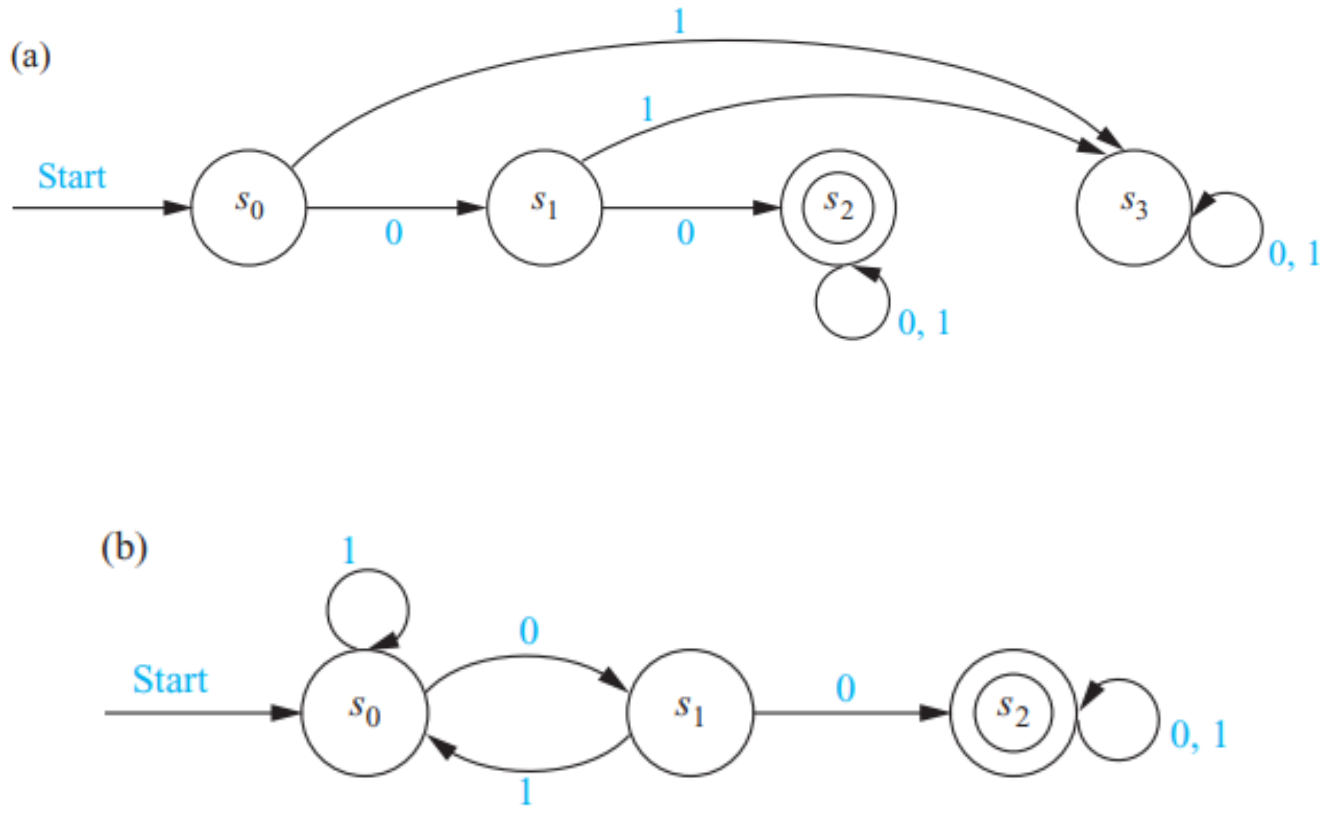
M_3

- Solution: The final states of M_3 are s_0 and s_3.The only strings that take s_0 to itself are λ, 0, 00, 000,..., that is, any string of zero or more consecutive 0s. The only strings that take s_0 to s_3 are a string of zero or more consecutive 0s, followed by 10, followed by any string. Hence, $L(M_3) = \{0^n, 0^n10x \mid n = 0, 1, 2,...,$ and x is any string$\}$.

Construct DFSA

- a) the set of bit strings that begin with two 0s
- (b) the set of bit strings that contain two consecutive 0s
- (c) the set of bit strings that do not contain two consecutive 0s
- (d) the set of bit strings that end with two 0s
- (e) the set of bit strings that contain at least two 0s

Construct DFSA



DFSA example