نظرية الحاسبات

المحاضرة الثامنة

الزمن: ساعة

16.7. Finite-State Automata (FSA)

A finite-state automaton is a special kind of finite-state machine. It differs from an ordinary finite-state machine in two respects. It has no output, some of its states are distinguished as accepting states.

Definition A finite-state automaton or, simply, an automaton M consists of

1. A finite set A of input symbols.
2. A finite set S of states.
3. A **next-state function** f from S × A into S. This is also called the **transition function.** This is the function which describes the change of states during the transition. This mapping is usually represented by a **state (transition) table** or a **state (transition) diagram.**

4. A subset Y of set S of accepting states.
5. An initial state $s_0 \in S$.

It is written as $M = (A, S, f, Y, s_0)$

This special kind of finite-state machine is known as **deterministic finite automata** (*dfa*).

The finite automaton is called "finite" because the number of possible states and number of elements in the alphabet are both finite, automation because the change of the states is totally governed by the input and "deterministic" refers to the fact that on each input there is one and only one state to which the automaton can have transition from the current state. Finite–state automata are of special interest because of their relationship to languages. (note : The singular of automata is automaton.) We can represent finite state automata using either state tables or state diagrams. The accepting states are indicated in state diagram by using double circles.

Example 31. A finite-state automaton A is defined by a transition diagram shown in Fig. 16.10

(*a*) Find its states $S = \{s_0, s_1, s_2\}$

(*b*) Find its input symbols $A = \{0, 1\}$

(*c*) Find its initial state s_0

(*d*) Find its accepting states s_2

(*e*) Find $f(s_1, 1)$ s_2

(*f*) Write its next – state table.

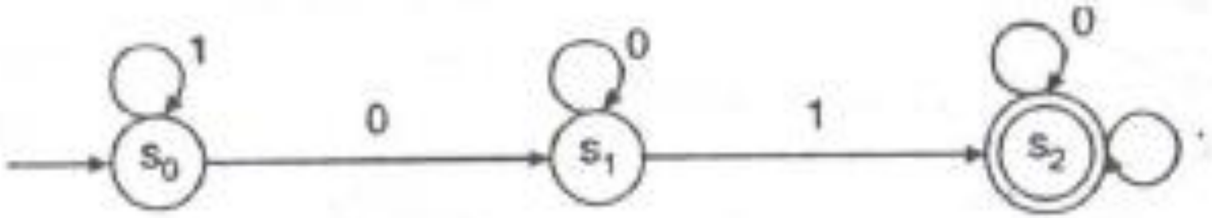


Fig. 16.10

Solution. (*a*) The states of A are s_0, s_1 and s_2, since these are the labels of the circles.

(*b*) The input symbols of A are 0 and 1 since they are the labels of the arrows.

(*c*) The initial state of A is s_0 since the unlabelled arrow points to s_0.

(*d*) The accepting state is s_2 since this state is marked by double circle.

(*e*) $f(s_1, 1) = s_2$ since there is arrow from s_1 to s_2 labelled 1.

(*f*) The next-state table is given below

		f	
	S \ A	0	1
→	s_0	s_1	s_0
	s_1	s_1	s_2
⊙	s_2	s_2	s_2

Table 16.6

Example 32. Consider the finite-state automaton B defined by the following next-state table.

(*a*) What are the states of B? $S = \{s_0, s_1, s_2, s_3\}$

(*b*) What are the input symbols of B? $A = \{a, b, c\}$

(*c*) What is the initial state of B? s_0

(*d*) What are the accepting states of B? s_3

(*e*) Find $f(s_0, c)$ s_2

(*f*) Draw the transition diagram of B.

	f		
S \ A	a	b	c
→ s_0	s_1	s_0	s_2
s_1	s_0	s_3	s_0
s_2	s_3	s_2	s_0
(o) s_3	s_1	s_0	s_1

Table 16.7

Solution. (*a*) The states of B are s_0, s_1, s_2 and s_3.

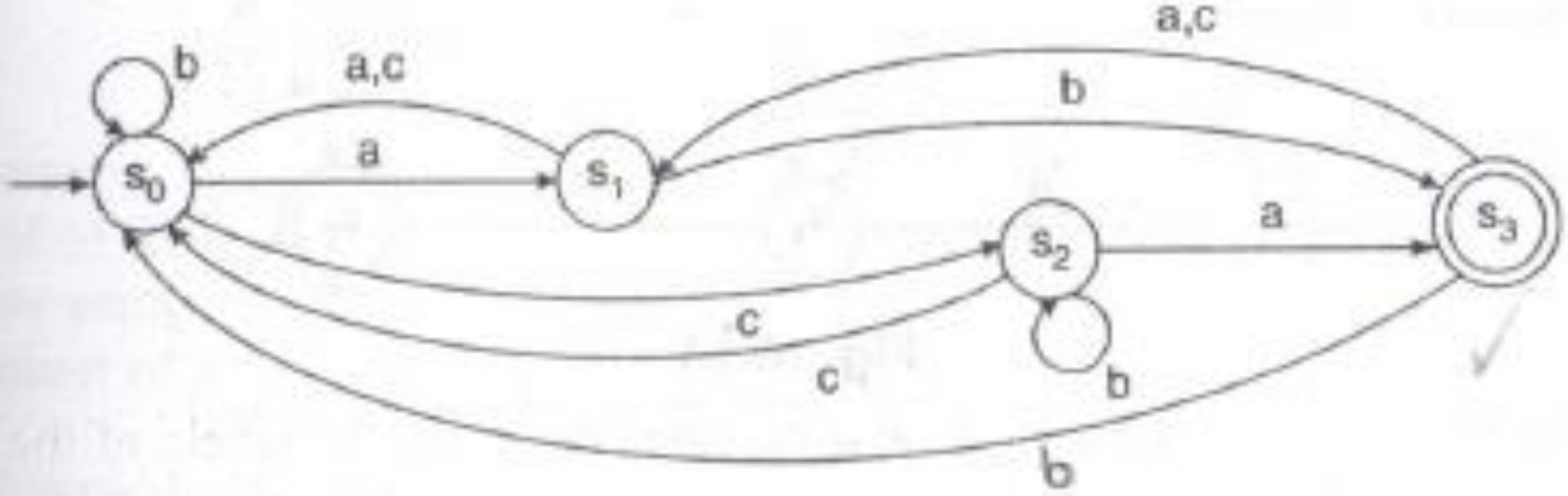
(*b*) The input symbols of B are *a*, *b* and *c*.

(*c*) The initial state of B is s_0.

(*d*) The accepting state of B is s_3.

(*e*) $f(s_0, c) = s_2$.

(*f*) The transition diagram of B is shown in fig 16.11.



Example 33. Redraw the transition diagram of the finite-state automation of Fig. 16.12 as a ansition diagram of a finite-state machine.

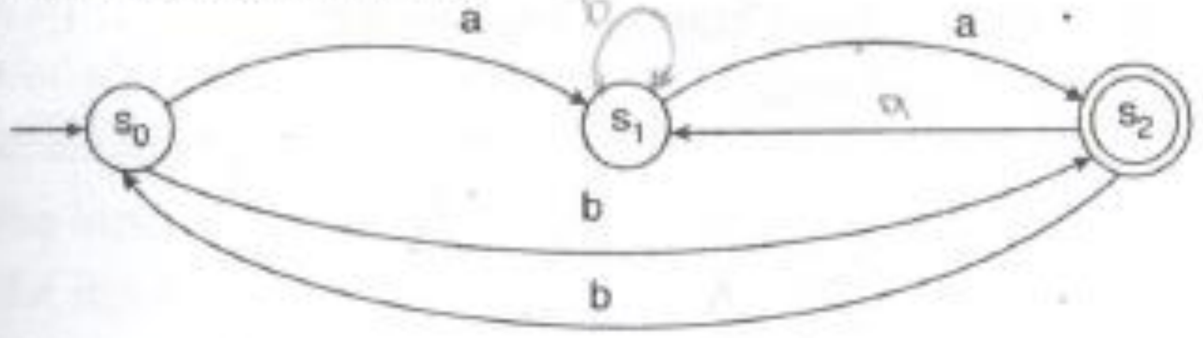


Fig. 16.12

Solution. Since s_2 is an accepting state, we lebel all its incoming edges with output 1. The states s_0 and s_1 are not accepting states, so we label all their incoming edges with output 0. We obtain the required transition diagram as shown in Fig. 16.13.

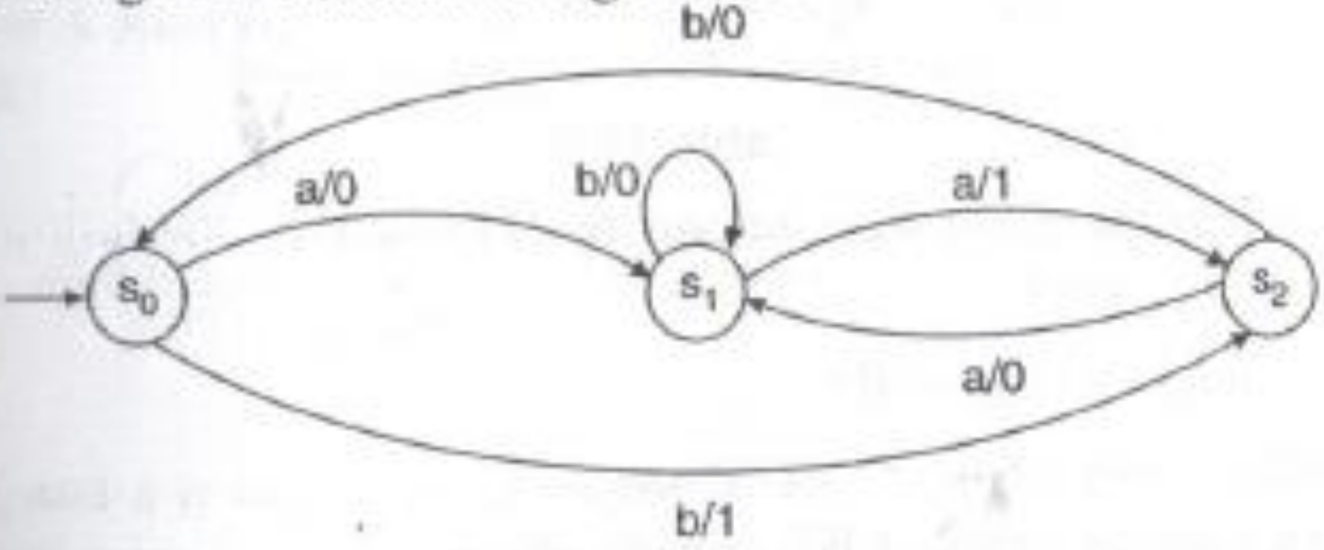


Fig. 16.13

The Language accepted by an automaton

Suppose a string of input symbols is fed into finite-state automaton in sequence. After each successive input symbol has changed the state of automaton, the automaton ends up to a certain state (accepting or a non accepting). The status of this final state determines whether the string is accepted by the finite-state automaton. These strings that send the automaton to an accepting state are said to be **accepted (or language recognized)** by the automaton. If a string is not accepted, we say it is **rejected.** The language accepted by a finite–state automaton M, denoted by L (M), is the set of all strings that are accepted by M *i.e.*,

$$L(M) = \{x \in I^* \mid x \text{ is accepted by } M\}$$

where I^* be the set of all strings over I.

The language accepted by a finite state automaton is termed as regular language. If we can construct a *dfa* for a given language, then the language is called a regular.

The power of a machine does not lie in the number of strings it accepts, but in its ability to discriminate to accept some and reject others.

Example 34. Consider the finite-state automaton A shown in Fig. 16.14.

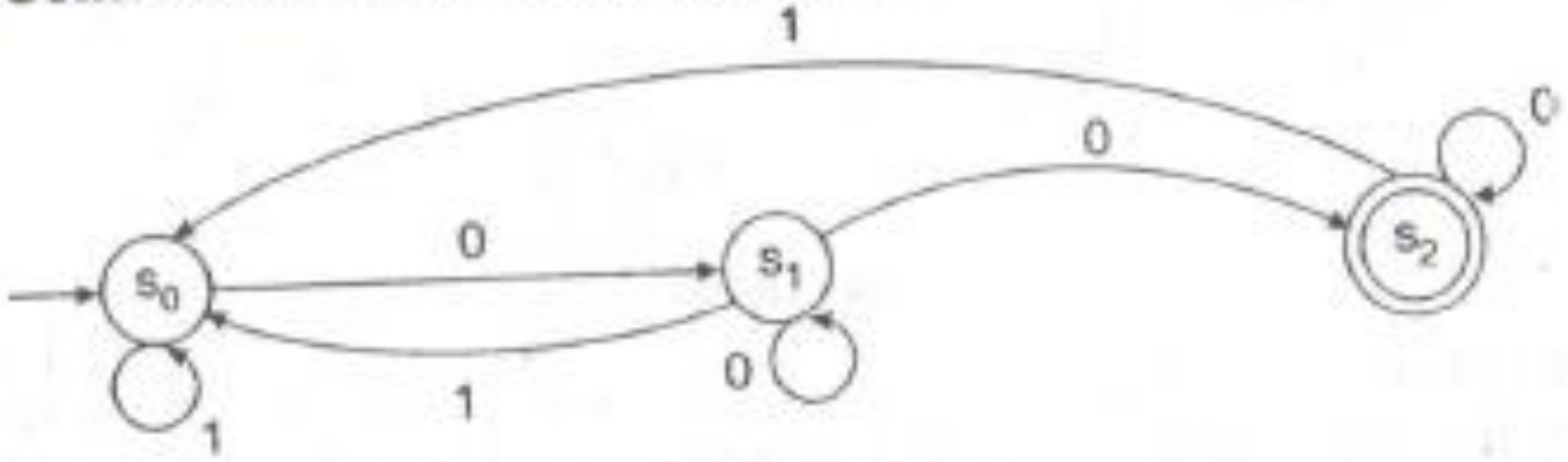


Fig. 16.14

(*a*) To what state does the automaton go when the symbols of the following strings are input to it in sequence starting from the initial state?

(*i*) 00 (*ii*) 0010 (*iii*) 10101 (*iv*) 010100

(*b*) Which of the strings in part (*a*) are accepted by A.

(*c*) What is the language accepted by A? any string of 0's and 1's that ends in 00

Solution. (*a*) (*i*) $s_0 \xrightarrow{0} s_1 \xrightarrow{0} s_2$

(*ii*) $s_0 \xrightarrow{0} s_1 \xrightarrow{0} s_2 \xrightarrow{1} s_0 \xrightarrow{0} s_1$

(*iii*) $s_0 \xrightarrow{1} s_0 \xrightarrow{0} s_1 \xrightarrow{1} s_0 \xrightarrow{0} s_1 \xrightarrow{1} s_0$

(*iv*) $s_0 \xrightarrow{0} s_1 \xrightarrow{1} s_0 \xrightarrow{0} s_1 \xrightarrow{1} s_0 \xrightarrow{0} s_1 \xrightarrow{0} s_2$

Hence (*i*) s_2 (*ii*) s_1 (*iii*) s_0 (*iv*) s_2

(*b*) Since s_2 is an accepting state, the string 00 and 010100 send A to an accepting state and hence they are accepted by A.

(*c*) We note that if x is any string that ends in 00, then x is accepted by A. For if the length of $x \geq 2$, then the first n-2 symbols of x have been input, A is in one of the three states s_0, s_1 and s_2. From any of these three states, the input of 00 in sequence sends A first s_1 and then the accepting state s_2. Hence, any strings of 0*s* and 1*s* that ends in 00 is accepted by *A*..

... whose transition diagram is shown

Example 35. Describe the language accepted by dfa A whose transition diagram is shown in Fig. 16.15.

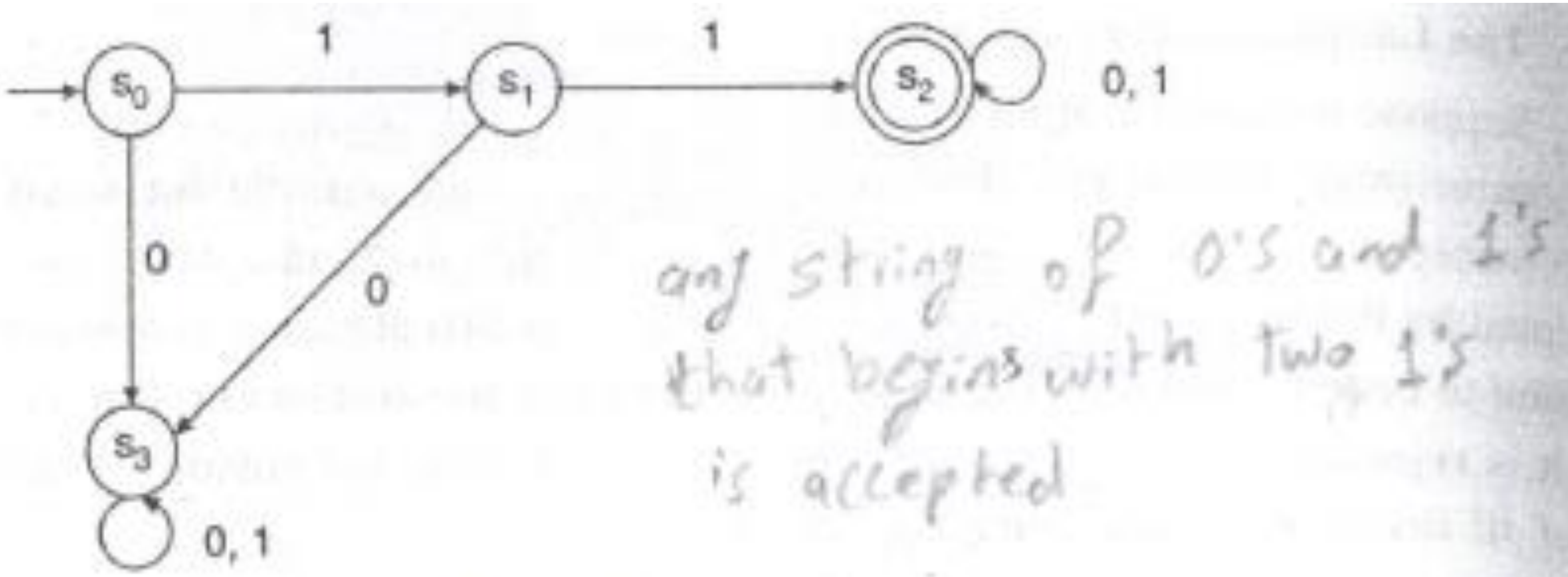


Fig. 16.15

Solution. If the input string begins with 0, A moves to s_3, a nonaccepting state from which it can never exist. If the input string begins with *a* 1, A moves to s_1. If the next input is a 0, A goes to s_3, and can never leave. But if the next input is also 1, a enters s_2 which is the accepting state and it remains there. Hence the strings of 0 and 1 symbols that begin with two 1s is accepted by A.

Example 36. Consider the dfa B in Fig 16.16. Find the Language accepted by the following dfa.

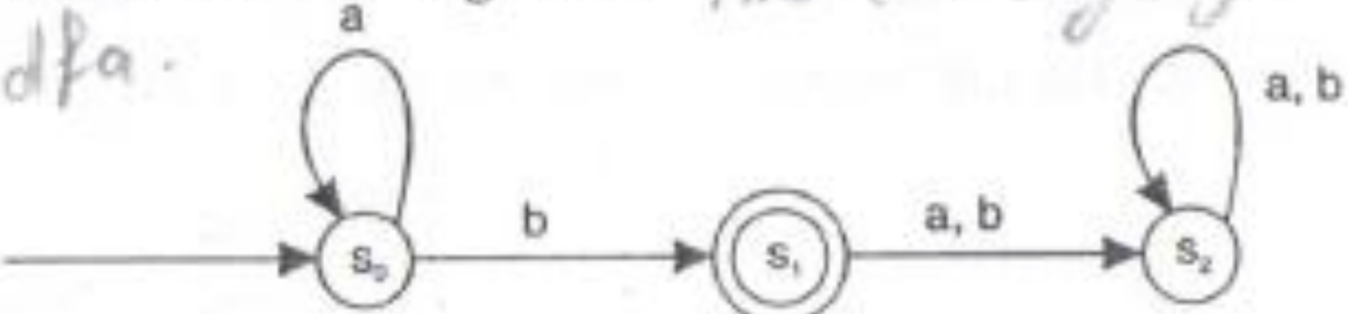


Fig. 16.16

The dfa shown in above Fig. remains in its initial state s_0 until the first b is encountiud. If this is the last symbol of the input, then the string is accepted. If not the, dfa goes into state s_2 from which it can never escape such a state is called a **trap state**. The automaton accepts only all these strings which consist of an arbitrary number of a's, followed by a single b, *i.e.*

$$L(B) = \{ a^n b : n \geq 0 \}$$