نظرية الحاسبات
المحاضرة التاسعة
الزمن: ساعة

Example 37. Draw the transition diagram of a finite-state automaton A that accepts the given set of strings over $\{a, b\}$

(*i*) even number of a's (*ii*) exactly one b.

Solution. (*i*) The automaton A must have two states

s_0 : state when the input string contain even number of a's. and also initial state

s_1 : state when the input string contain odd number of a's.

The transition diagram is shown in Fig. 16.17.

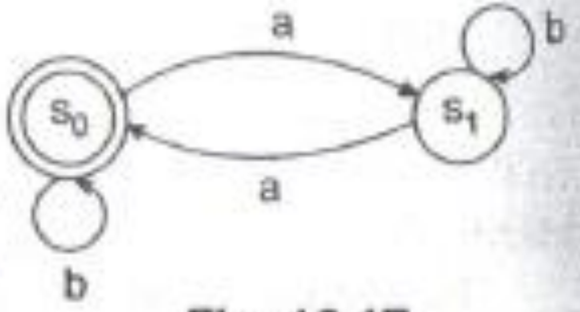


Fig. 16.17

If A is in state s_0 and input is b, A will stay in s_0 but as soon as input is a, A moves to s_1. It comes back to accepting state if s_1 has an input a. So, the string consisting even number of a's is accepted by A.

(*ii*) The automaton A must contain three distinct states s_0, s_1 and s_2 as shown in Fig. 16.18.

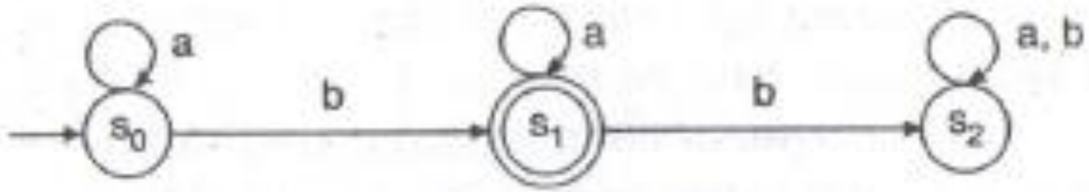


Fig. 16.18

If A is in state s_0 and a is an input, it will remain in s_0. It moves to s_1 if b is an input. Now, when A is in state s_1 and a is an input, the input string still has a single b. So A stays in accepting

ate s_1. But if b is an input string then it contains more than one b, then it must leave s_1 and go to ird state s_2 from which there is no return to s_1

Example 38. Design a finite state automata that will accept the set of natural numbers x which are divisible by 3.

Solution. On dividing by 3, we have three possible classes of numbers. These are with remainder 0, 1 are 2 respectively. Let the corresponding states are s_0, s_1 are s_2.

Let $M = (A, S, f, Y\ s_0)$ where $A = \{0, 1, ..., 9\}$.

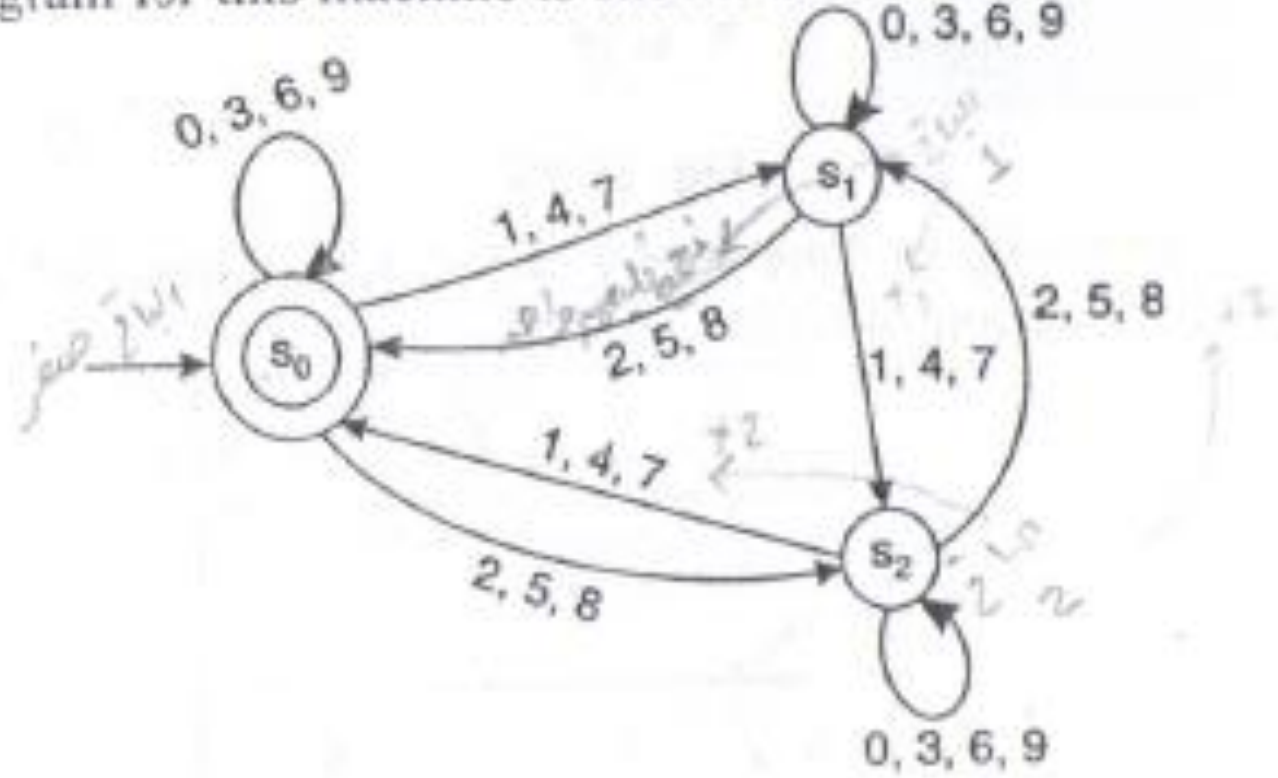
$S = \{s_0, s_1, s_2\}$ $Y = \{s_0\}$, f is defined by

$f(s_0, a) = s_0, f(s_1, a) = s_1, f(s_2, a) = s_2$ for $a \in \{0, 3, 6, 9\}$

$f(s_0, b) = s_1, f(s_1, b) = s_2, f(s_2, b) = s_0$ for $b \in \{1, 4, 7\}$

$f(s_0, c) = s_2, f(s_1, c) = s_0, f(s_2, c) = s_1$ for $c \in \{2, 5, 8\}$

The transition diagram for this machine is shown below



[Since s_0 is the state corresponding to class of string divisible by 3, s_0 is also final state]

Let $w = 117$, the computation is

$$f(s_0, 117) = f(s_1, 17) = f(s_2, 7) = s_0 \text{ (accetpting state)}$$

Let $w = 125$

$$f(s_0, 125) = f(s_1, 25) = f(s_0, 5) = s_2 \text{ (non-occepting state)}$$

Nondeterministic Finite-state Automation

There is another important type of finite state automation in which there may be several possible next states for each of input value and state. Such machines are called nonderministic.

Definition. A nondeterministic finite-state automation (*nfa*) A consists of

1. A finite set A of input symbols.
2. A finite set S of states.
3. A next-state function f from $S \times A$ into P (S) which is the power set of S, the set of all subsets of S.
4. A subset Y of set of accepting states.
5. An initial state $s_0 \in S$.

It is written as $M = \{A, S, f, Y, s_0\}$

We can represent nondeterministic finite-state automata using next-state table or transition diagram similar to that of finite-state automata.

We note that the difference between the deterministic and nondeterministic automata is only in *f*. For deterministic finite automata (*dfa*) the outcome is a state *i.e.*, an element of S ; for nondeterministic automaton the outcome is a subset of S *i.e.*, in deterministic automata, the next – state function takes us to a uniquely defined state, whereas a nondeterminstic finite-state automaton the next-state function takes us to a set of states. Each *nfa* accepts a language that is also accepted

... convert *nfa* to *dfa*.

A string is accepted by an nfa if there is some sequence of possible moves that will put the machine in a final state at the end of the string. A string is rejected only if there is no possible sequence of moves by which a final state can be reached.

Example 39. For nondeterministic finite–state automation whose transition diagram is shown in Fig. 16.19, find (*a*) initial state (*b*) state set (*c*) input set (*d*) state table defining the next-state function.

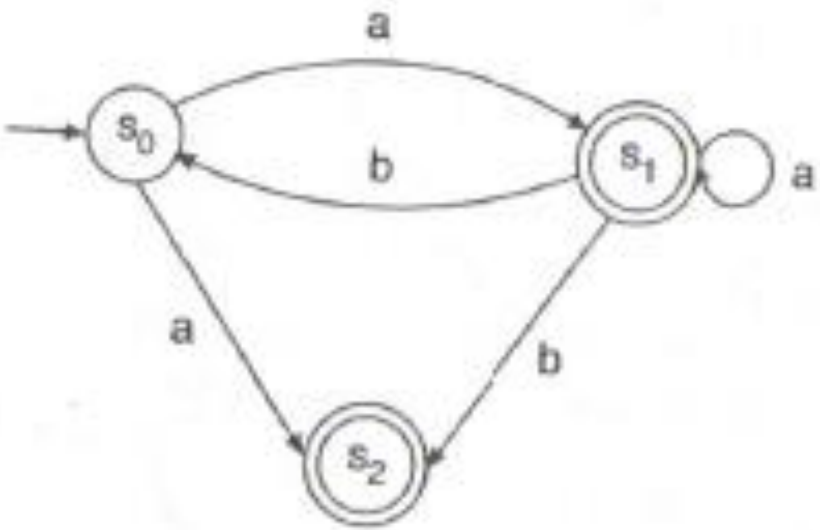


Fig. 16.19

Solution. (*a*) Initial state is s_0 (*b*) State set $s = \{s_0, s_1, s_2\}$ (*c*) Input set A = $\{a, b\}$ (*d*) The state table defining the next-state function is given below.

		f	
	S \ A	*a*	*b*
→	s_0	$\{s_1, s_2\}$	ϕ
⊙	s_1	$\{s_1\}$	$\{s_0, s_2\}$
⊙	s_2	ϕ	ϕ

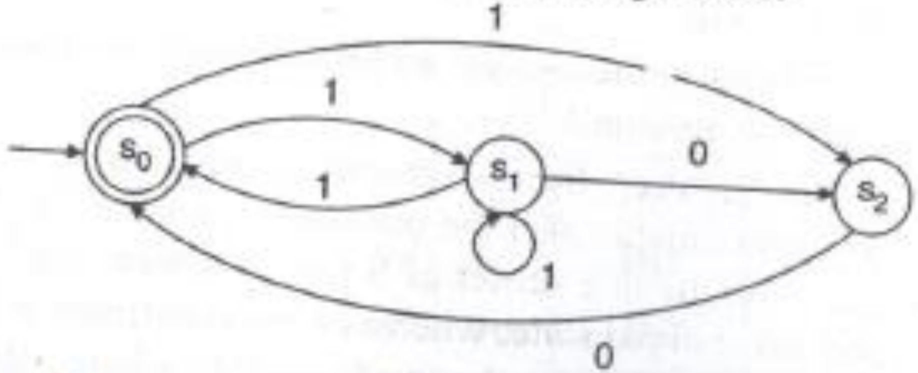
Example 40. Find the transition diagram of the nondeterministic finite state automata M with the state table shown below

$A = \{0, 1\}$, $S = \{s_0, s_1, s_2\}$, $O = \{s_0\}$

S \ a	0	1
s_0	ϕ	$\{s_1, s_2\}$
s_1	$\{s_2\}$	$\{s_0, s_1\}$
s_2	$\{s_0\}$	ϕ

Table 16.9

Solution. The transition diagram M is shown in Fig. 16.20.



Example 41. Find the language recognized by the given nondeterministic finite-state automaton M whose transition diagram is shown in Fig. 16.21.

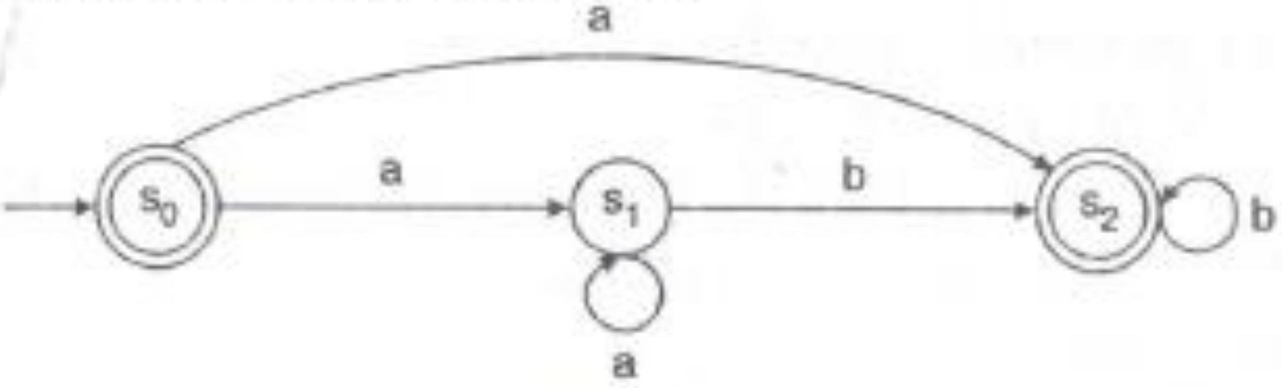


Fig. 16.21

Solution. Here initial state s_0 and the states s_0 and s_2 are both accepting states. Since the initial and the accepting state s_0, the language recognized by M is λ. There are two ways to reach the state s_2, one from s_0 to s_2 and other from s_0 to s_1, then s_2. Thus the language accepted by M is

$$L(M) = \{\lambda, a\} \cup \{a^m b^n \mid m, n \geq 1\}$$

Conversion of NFA to DFA

Although there are many languages for which an nfa is easier to construct than a dfa, such as the language of strings that end in 01, it is a fact that both *nfa* and *dfa* have identical capabilities. For every language accepted by some *nfa*, there is a *dfa* that accepts the same language. We shall describe a procedure to convert a given *nfa* to an equivalent *dfa*.

Given a *nfa* $M = \{A, S, f, Y, s_0\}$ then corresponding *dfa* $M^d = \{A, S^d, f^d, Y^d, s_0^d\}$ is defined as follows:

$$S^d = p(S), \text{ the power set of S.}$$
$$s_0^d = \{s_0\}$$
$$Y^d = \{B \in S^d : B \cap Y \neq \phi\}.$$

That is Y^d is all sets of M's states that include at least one accetping state of M

and $f^d : Y^d \times A \rightarrow Y^d$ is defined by

$$f^d(B, a) = \bigcup_{s \in B} f(s, a)$$

for all $B \in Y^d$, $a \in A$

Every state of M^d whose total contains a final state of M is defined as a final vertex.

The next two examples illustrate step by step how to construct a dfa equivalent to a given *nfa*.

Example 42. Find a state transition table for the given nfa of Fig 16.22 obtain a *dfa* equivalent to nfa.

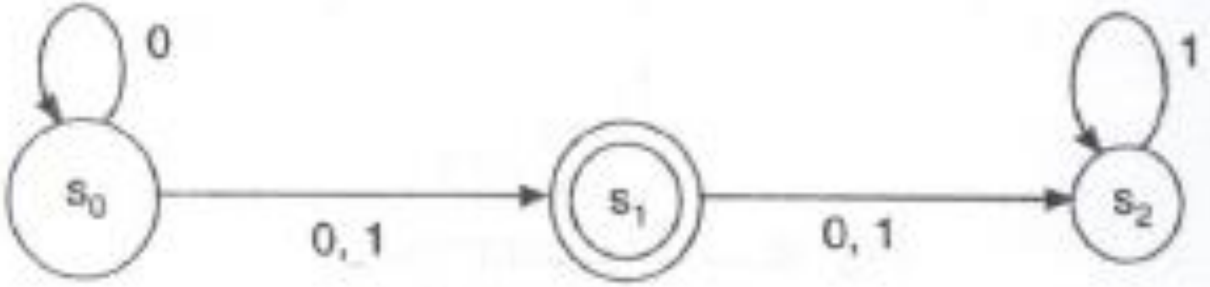


Fig. 16.22

Solution. The state transition table for the given *nfa* is

	f	
S \ *A*	0	1
s_0	$\{s_0, s_1\}$	$\{s_1\}$
s_1	$\{s_2\}$	$\{s_2\}$
s_2	ϕ	$\{s_2\}$

The state labeled ϕ represents an impossible move for the nfa and, therofore, means non acceptance of the string.

Here, we construct the corresponding *dfa*.

$$M^d = \{ A, S^d, f^d, Y^d, s_0^d \}$$

$$S^d = P(s) = \{\phi, \{s_0\}, \{s_1\}, \{s_2\}, \{s_0, s_1\}, \{s_0, s_2\}, \{s_1, s_2\}, \{s_0, s_1, s_2\}\}$$

$$Y^d = \{\{s_1\}, \{s_0, s_1\} \ \{s_1, s_2\} , \{s_0, s_1, s_2\}\}$$

$$s_0^d = \{s_0\}$$

and $f^d : s^d \times A \to s^d$ is defined by the following transition table.

... by the following transition table.

S^d \ A	f^d	
	0	1
ϕ	ϕ	ϕ
$\{s_0\}$	$\{s_0, s_1\}$	$\{s_1\}$
$\{s_1\}$	$\{s_2\}$	$\{s_2\}$
$\{s_2\}$	ϕ	$\{s_2\}$
$\{s_0, s_1\}$	$\{s_0, s_1, s_2\}$	$\{s_1, s_2\}$
$\{s_0, s_2\}$	$\{s_0, s_1\}$	$\{s_1, s_2\}$
$\{s_1, s_2\}$	$\{s_2\}$	$\{s_2\}$
$\{s_0, s_1, s_2\}$	$\{s_0, s_1, s_2\}$	$\{s_1, s_2\}$

The set of final states which contains the final state s_1 in M is $\{\{s_1\}, \{s_0, s_1\}\ \{s_1, s_2\}, \{s_0, s_1, s_2\}\}$ the state transition diagram of M^d is

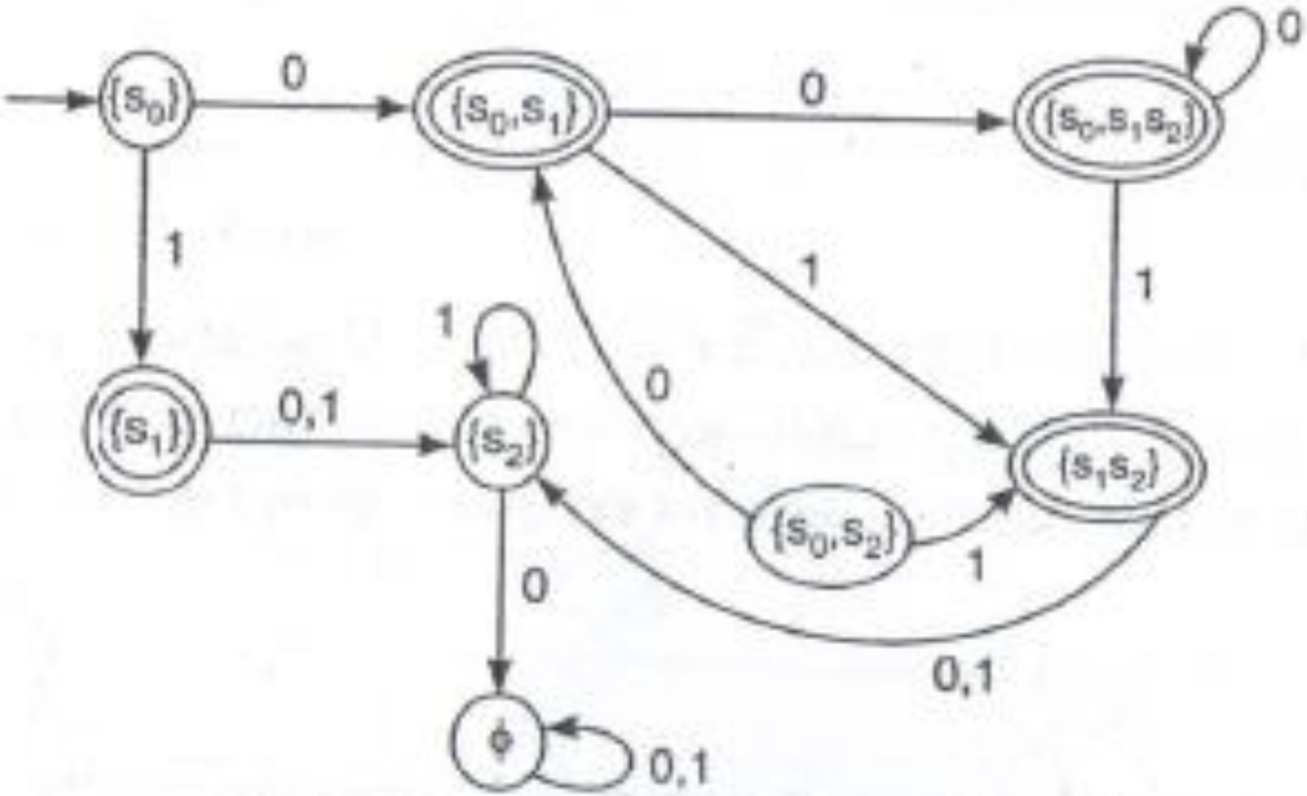


Fig 16.23. Transition diagram fo M^d.