13.3 Finite-State Machines with No Output

Concatenation

Suppose that A and B are subsets of V^*, where V is a vocabulary. The *concatenation* of A and B, denoted by AB, is the set of all strings of the form xy, where x is a string in A and y is a string in B.

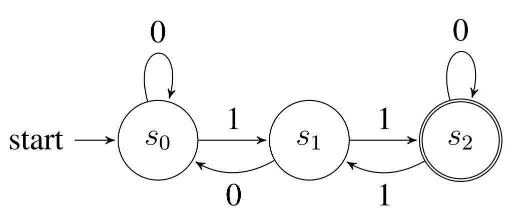
Kleene closure

Suppose that A is a subset of V^*. Then the *Kleene closure* of A, denoted by A^*, is the set consisting of concatenations of arbitrarily many strings from A. That is, $A^* = \bigcup_{k=0}^{\infty} A^k$.

Finite-state Automata

Finite-state automata are finite-state machines with no output.
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 $(f : S \times I \to S)$
- an *initial state* s_0
- a finite set F of *final states* (or *accepting states*)



State	Input	
	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_2	s_1

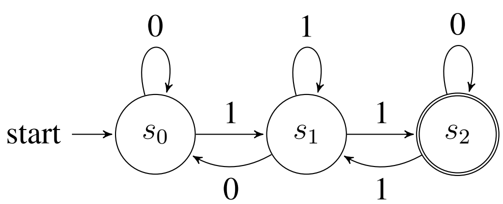
Language Recognition by Finite-State Machines

A string x is said to be *recognized* or *accepted* by the machine $M = (S, I, f, s_0, F)$ if it takes the initial state s_0 to a final state, that is $f(s_0, x)$ is a state in F. The *language recognized* or *accepted* by the machine M, denoted by $L(M)$, is the set of all strings that are recognized by M. Two finite-state automata are called *equivalent* if they recognize the same language.

Nondeterministic Finite State Automata

So far we have only discussed *deterministic* finite state automata because each pair of state and input value has a unique next state given by the transition function.
We will now discuss *nondeterministic* finite state automata where there can be several possible next states for each pair of state and input value.
A *nondeterministic finite-state automaton* $M = (S, f, I, s_0, F)$ consists of

- a finite set S of *states*
- a finite *input alphabet* I
- a *transition function* f $(f : S \times I \rightarrow P(S))$
- an *initial state* s_0
- a finite set F of *final states*



State	Input	
	0	1
s_0	s_0	s_1
s_1	s_0	s_1, s_2
s_2	s_2	s_1

Theorem 1

If the language L is recognized by a nondeterministic finite-state automaton M_0, then L is also recognized by a deterministic finite-state automaton M_1.

13.3 pg. 975 # 1

Let $A = \{0, 11\}$ and $B = \{00, 01\}$. Find each of these sets.

a) AB

$AB = \{000, 001, 1100, 1101\}$

b) BA

$BA = \{000, 0011, 010, 0111\}$

c) A^2

$A^2 = \{00, 011, 110, 1111\}$

13.3 pg. 975 # 5

Describe the elements of the set A^* for these values of A.

a) $\{10\}$

The set of strings where there are zero or more copies of 10, defined as $\{(10)^n | n = 0, 1, 2, \ldots\}$

b) $\{111\}$

The set of strings where there are zero or more copies of 111, defined as $\{1^{3n} | n = 0, 1, 2, \ldots\}$

c) $\{0, 01\}$

The set of strings where every 1 is immediately preceded by a 0.

13.3 pg. 975 # 9

Determine whether the string 11101 is in each of these sets.

a) $\{0,1\}^*$

Yes.

b) $\{1\}^*\{0\}^*\{1\}^*$

Yes.

c) $\{11\}\{0\}^*\{01\}$

No.

d) $\{11\}^*\{01\}^*$

No.

e) $\{111\}^*\{0\}^*\{1\}$

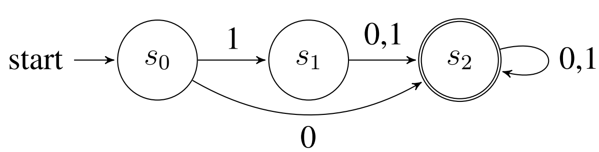
Yes.

f) $\{11,0\}\{00,101\}$

Yes.

13.3 pg. 876 # 17

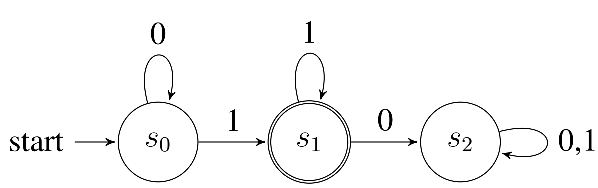
Find the language recognized by the given deterministic finite-state automaton.



The language is $\{0,10,11\}\{0,1\}^*$

13.3 pg. 876 # 19

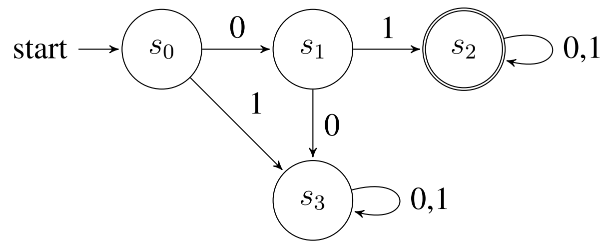
Find the language recognized by the given deterministic finite-state automaton.



The language is $\{0\}^*\{1\}\{1\}^*$

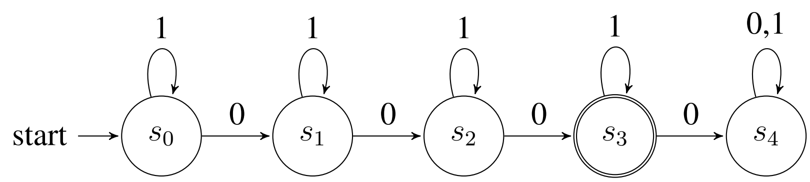
13.3 pg. 876 # 23

Construct a deterministic finite-state automaton that recognizes the set of all bit strings beginning with 01.



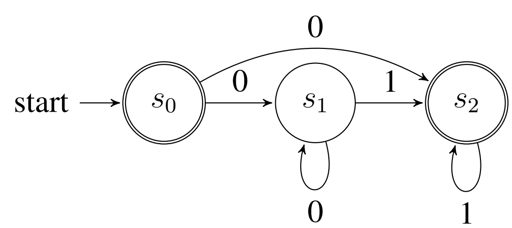
13.3 pg. 876 # 27

Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain exactly three 0s.



13.3 pg. 877 # 45

Find the language recognized by the given nondeterministic finite-state automaton.



Since the initial state is an accepting state, we know that the automaton accepts the empty string, λ. We now need to figure how to enter the second accepting state, s_2. By inspection, we can see that we have two ways to reach there, by s_0 or by going through s_1. Let us first consider going through s_1. We can only reach s_2 by inputing one or more 0s followed by one or more 1s. The other way to reach s_2 is to skip s_1 by inputing one 0 and zero or more 1s. Thus, the language recognized is $\{\lambda\} \cup \{0^n 1^m | n, m \geq 1\} \cup \{01^m | m \geq 0\}$.