13.3 Finite-State Machines with No Output

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, $L(M_1) = \{1^n | n = 0, 1, 2, ...\}$.

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, $L(M_2) = \{1, 01\}$.

The final states of $M_3$ are $s_0$ and $s_3$. The only strings that take $s_0$ to itself are $\lambda$, 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, $L(M_3) = \{0^n, 0^n10x | n = 0, 1, 2, ..., and x is any string\}$.

DESIGNING FINITE-STATE AUTOMATA

We can often construct a finite-state automaton that recognizes a given set of strings by carefully adding states and transitions and determining which of these states should be final states. When appropriate we include states that can keep track of some of the properties of the input string, providing the finite-state automaton with limited memory. Examples 6 and 7 illustrate some of the techniques that can be used to construct finite-state automata that recognize particular types of sets of strings.

EXAMPLE 6

Construct deterministic finite-state automata that recognize each of these languages.

(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

Solution:

(a) Our goal is to construct a deterministic finite-state automaton that recognizes the set of bit strings that begin with two 0s. Besides the start state $s_0$, we include a nonfinal state $s_1$; we move to $s_1$ from $s_0$ if the first bit is a 0. Next, we add a final state $s_2$, which we move to from $s_1$ if the second bit is a 0. When we have reached $s_2$ we know that the first two input bits are both 0s, so we stay in the state $s_2$ no matter what the succeeding bits (if any) are.

We move to a nonfinal state $s_3$ from $s_0$ if the first bit is a 1 and from $s_1$ if the second bit is a 1.

The reader should verify that the finite-state automaton in Figure 3(a) recognizes the set of bit strings that begin with two 0s.