Chapter 1

Finite State Machines

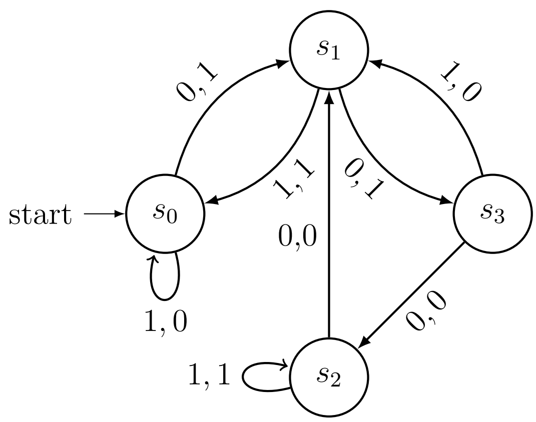
1.1 Finite state machines

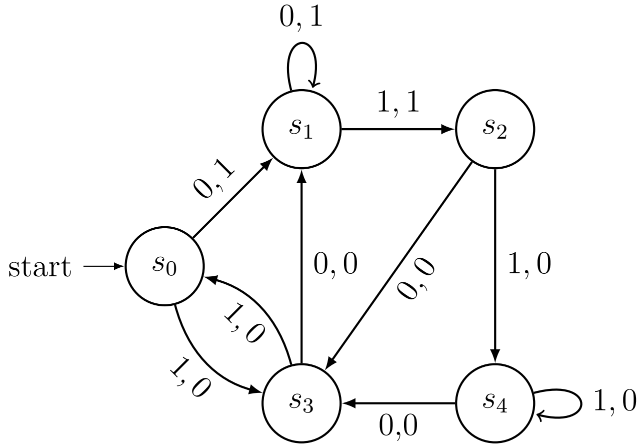
A finite state machine $M = (S, I, O, f, g, s_0)$ consists of a finite set S of states, a finite input alphabet I, a finite output alphabet O, a transition function f that assigns to each state and input pair a new state, an output function g that assigns to each state and input pair an output and an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

- We can use a state table to represent the values of the transition function f and the output function g for all pairs of states and input.

- A state diagram is a directed graph with labeled edges that represents a finite state machine.

	f		g	
	Input		Input	
State	0	1	0	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





	f		g	
	Input		Input	
State	0	1	0	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

- If the input string is 101011, the output string is 001000.

- Let $M = (S, I, O, f, g, s_0)$ be a finite state machine and $L \subseteq I^*$ (the set of all input string over I). We say that M recognizes (or accepts) L if an input string x belongs to L if and only if the last output bit produced by M when given x as input as a 1.

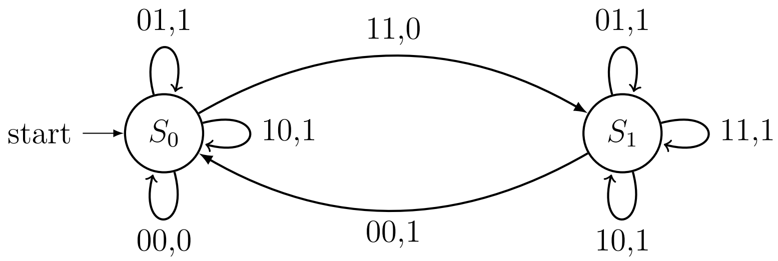
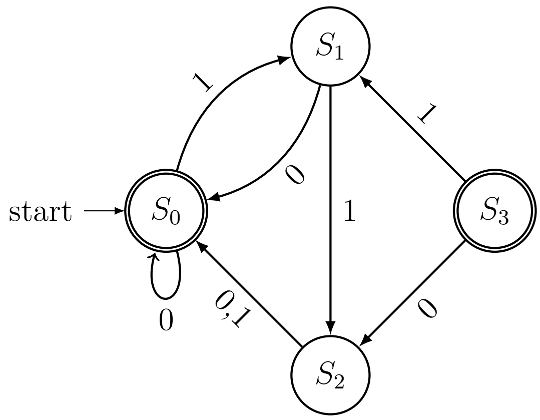


Figure 1.1: A finite state machine for binary addition

1.2 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 that assigns a next state to every pair of state and input (so that $f : S \times I \to S$), an initial or state s_0 and a subset F if S consisting of final (or accepting states).

	f	
	Input	
State	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



$M = (S, I, f, s_0, F)$, $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0, 1\}$, $F = \{s_0, s_3\}$ and f is given in the table.

- 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.

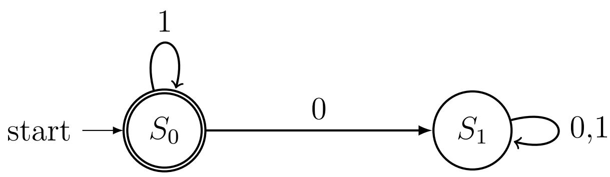


Figure 1.2: M_1

$$L(M_1) = \{1^n \mid n = 0, 1, 2, \dots\}$$

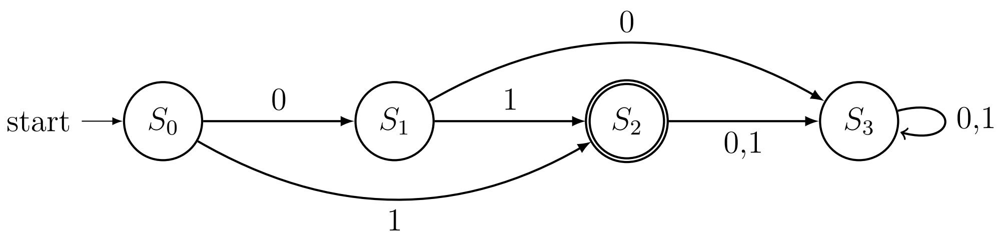


Figure 1.3: M_2

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

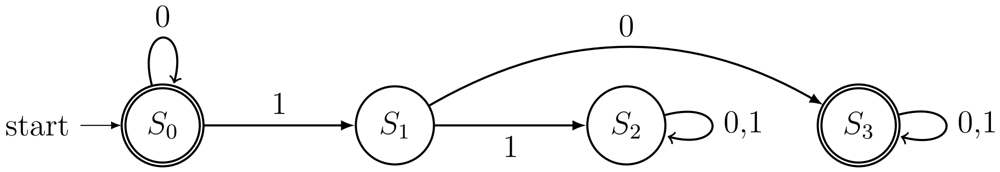


Figure 1.4: M_3

$$L(M_3) = \{0^n,\ 0^n 10x \mid n = 0, 1, 2, \ldots \text{ and } x \text{ is any string}\}$$

Problem 1.2.1. Construct deterministic finite set automata that recognize each of these languages:

(a) The set of bit strings that begin with two 0's.

(b) The set of bit strings that contains two consecutive 0's.

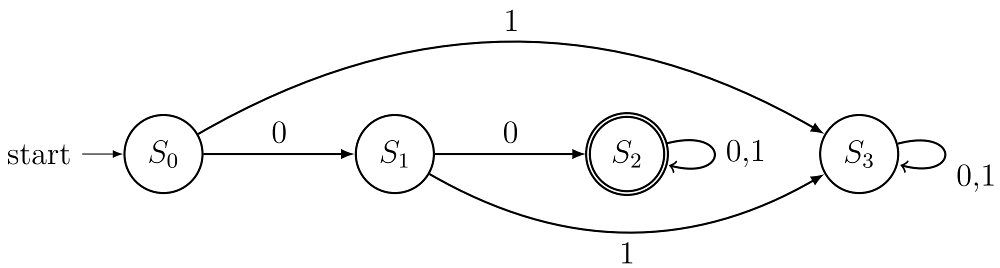
(c) The set of bit strings that do not contain two consecutive 0's.

(d) The set of bit strings that end with two consecutive 0's.

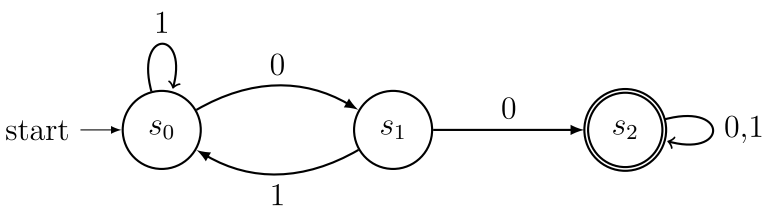
(e) The set of bit strings that contains at least two 0's.

Solution.

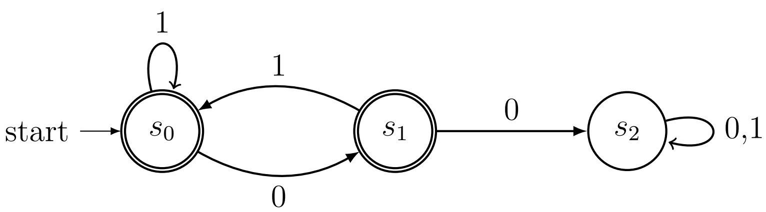
(a)



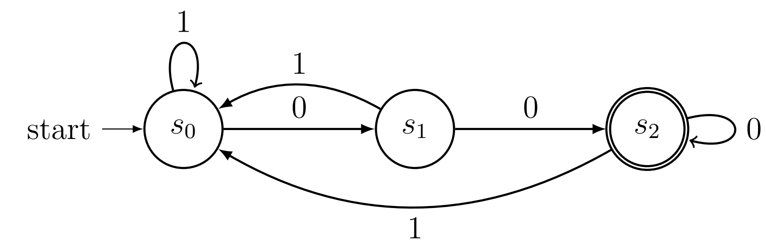
(b)



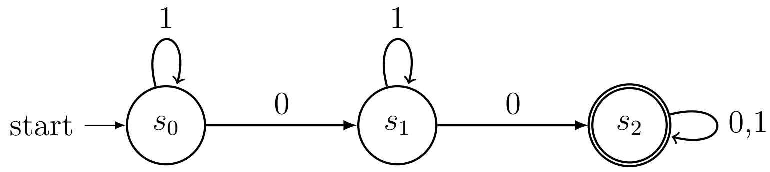
(c)



(d)



(e)



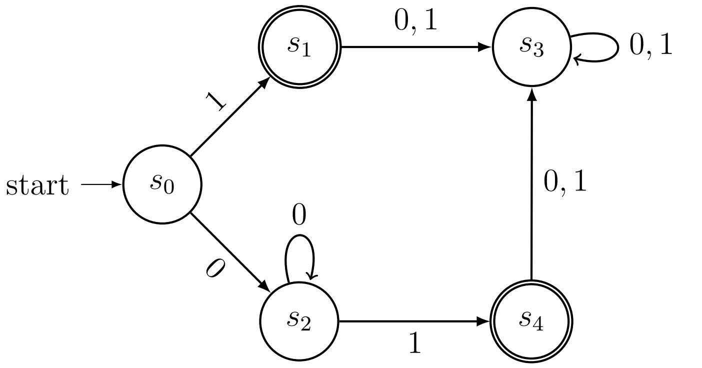


Figure 1.5: M_0

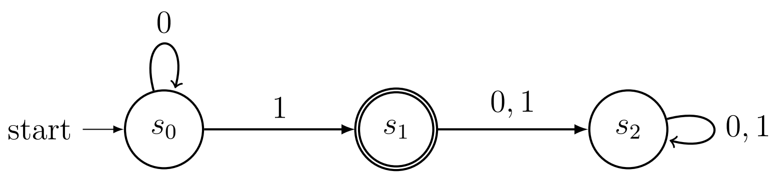


Figure 1.6: M_1

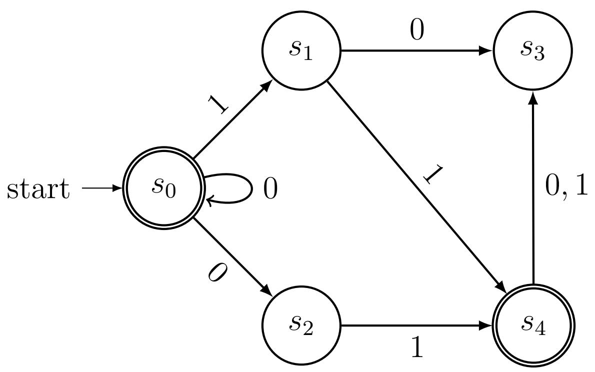
Problem 1.2.2. Show that M_0 and M_1 are equivalent.

Solution. For a string x to be recognized by M_0, x must take us from s_0 to the final state s_1 or the final state s_4.

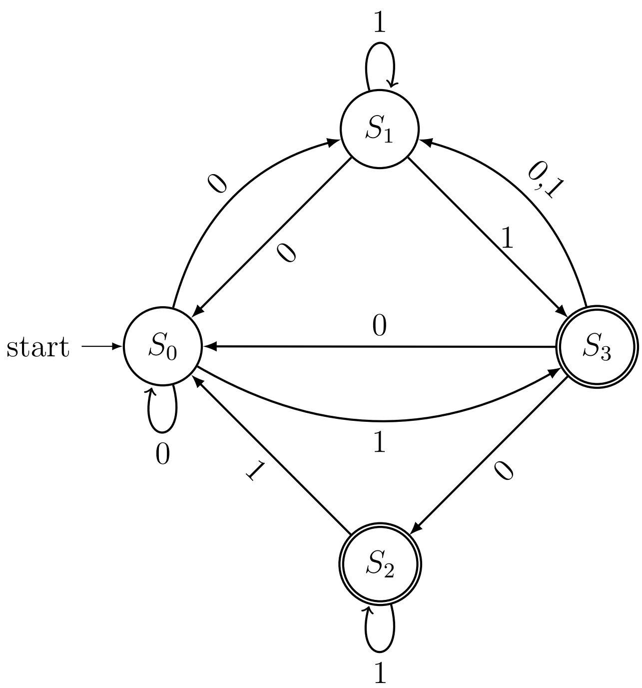
The only string that takes us from s_0 to s_1 is the string 1.

The strings that take us from s_0 to s_4 are those strings that begin with a 0, which takes us from s_0 to s_2, followed by zero or more additional zero which keep the machine in state s_2 followed by a 1, which takes us from s_2 to the final state s_4.

<u>H.W.</u> Nondeterministic finite state automata



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



	f	
	Input	
State	0	1
s_0	$s_0,\ s_1$	s_3
s_1	s_0	$s_1,\ s_3$
s_2		$s_0,\ s_2$
s_3	$s_0,\ s_1,\ s_2$	s_1