SPECIFICATION OF SEQUENTIAL SYSTEMS

- Synchronous sequential systems
- Mealy and Moore machines
- Time behavior
- State minimization

Definition

$$z(t) = F(x(0,t))$$

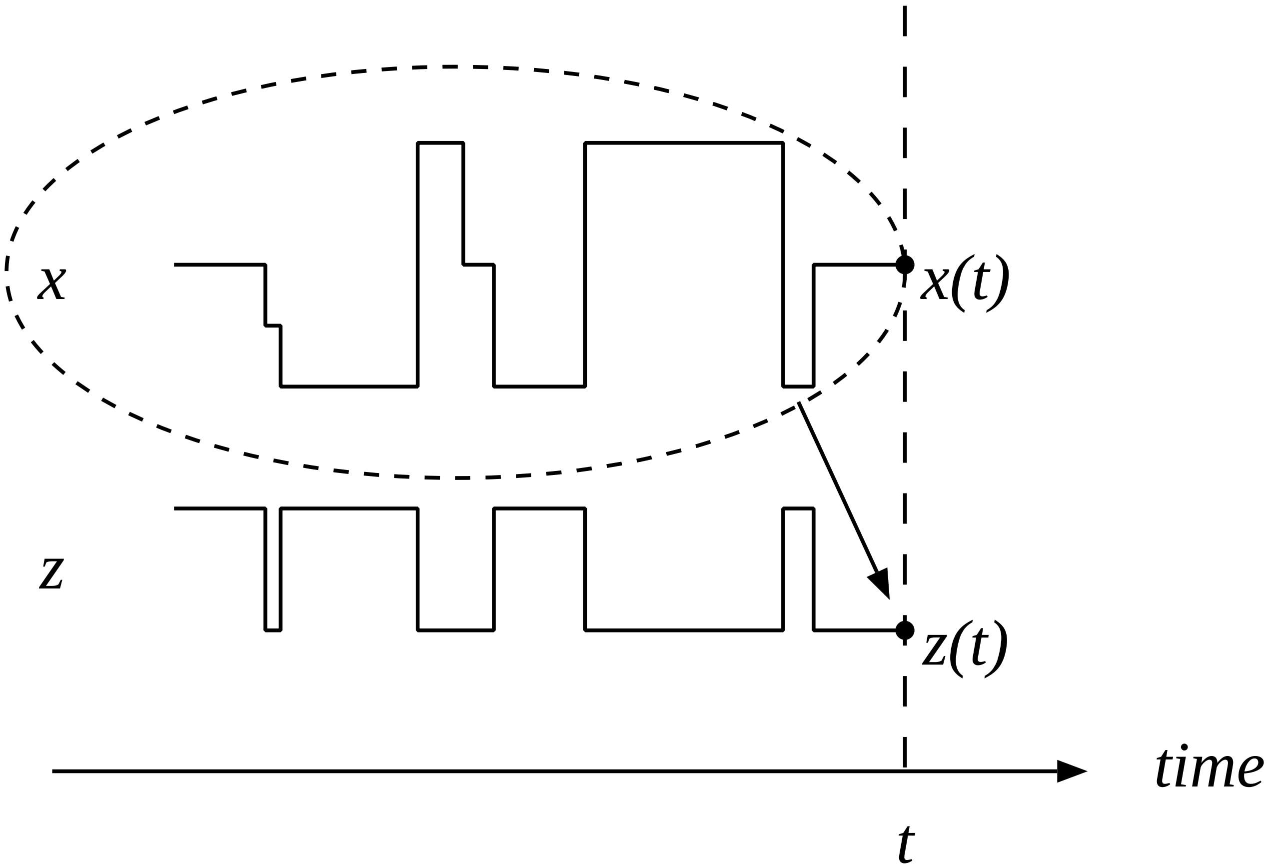


Figure 7.1: Input and output time functions.

Synchronous and asynchronous systems

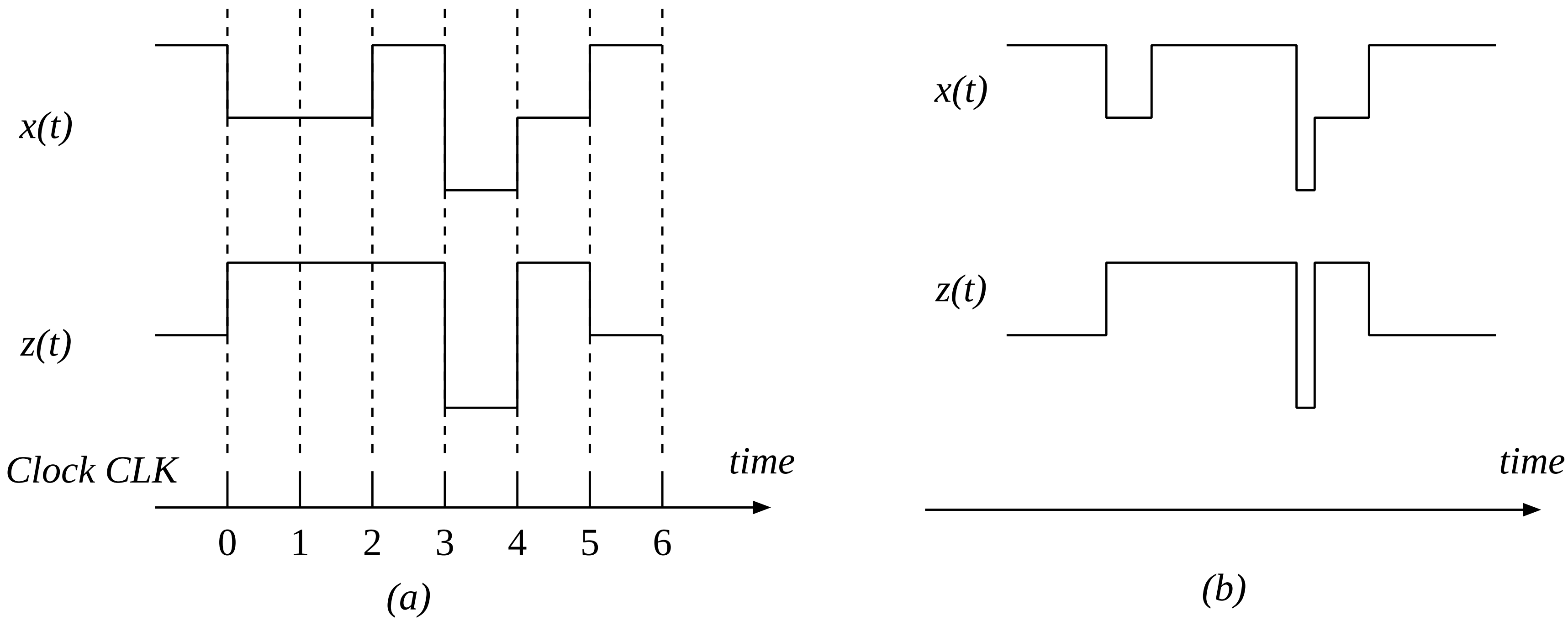


Figure 7.2: a) Synchronous behavior. b) Asynchronous behavior.

- Clock
- I/O sequence

$$x(2,5) = aabc$$

$$z(2,5) = 1021$$

Example 7.1: Serial decimal adder

x	1638753
y	3652425
z	5291178

- least-significant digit first (at t=0)

t	0	1	2	3	4	5	6
x(t)	3	5	7	8	3	6	1
y(t)	5	2	4	2	5	6	3
z(t)	8	7	1	1	9	2	5

State description

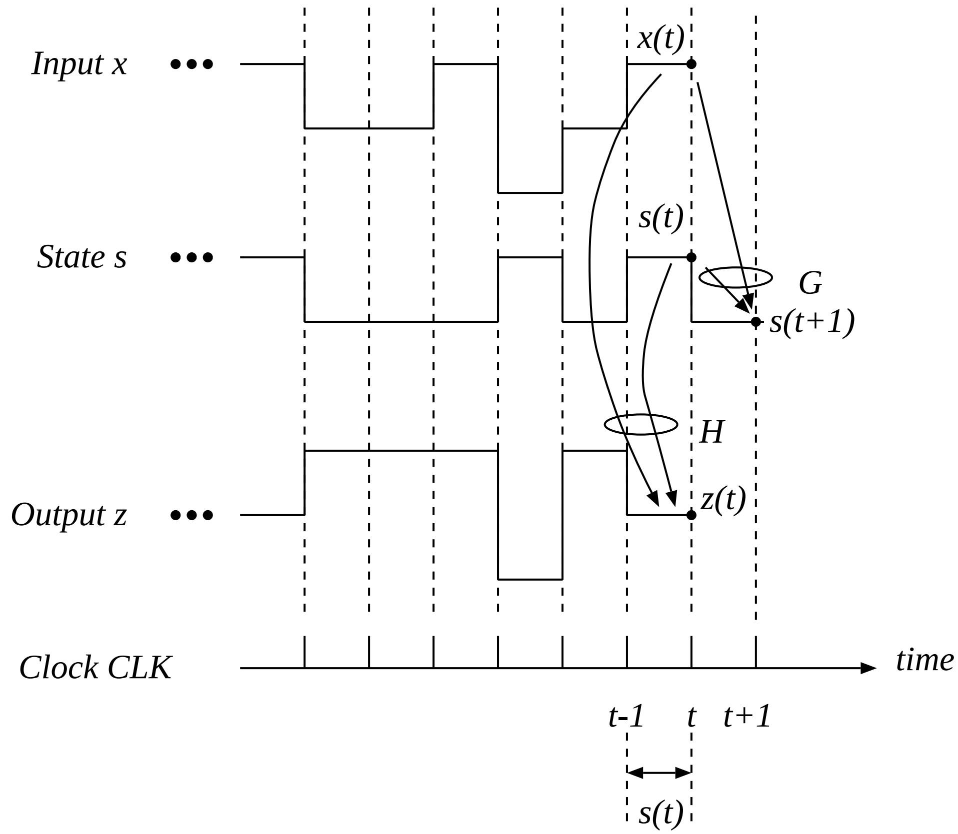


Figure 7.3: Output and state transition functions

$$\text{State-transition function} \quad s(t+1) = G(s(t), x(t))$$
$$\text{Output function} \quad z(t) = H(s(t), x(t))$$

Example 7.3: State description of serial adder

Input: $x(t), y(t) \in \{0, 1, ..., 9\}$
Output: $z(t) \in \{0, 1, ..., 9\}$
State: $s(t) \in \{0, 1\}$ (the carry)
Initial state: $s(0) = 0$

Functions: The transition and output functions are

$$s(t+1) = \begin{cases} 1 \ \textbf{if} \ \ x(t) + y(t) + s(t) \geq 10 \\ 0 \ \textbf{otherwise} \end{cases}$$

$$z(t) = (x(t) + y(t) + s(t)) \ mod \ 10$$

Example:

t	0	1	2	3	4	5	6
x(t)	3	5	7	8	3	6	1
y(t)	5	2	4	2	5	6	3
s(t)	0	0	0	1	1	0	1
z(t)	8	7	1	1	9	2	5

Example 7.4: Odd/Even

Time-behavior specification:

Input: $x(t) \in \{a, b\}$

Output: $z(t) \in \{0, 1\}$

Function: $z(t) = \begin{cases} 1 & \textbf{if} \;\; x(0,t) \text{ contains an even number of } b'\text{s} \\ 0 & \textbf{otherwise} \end{cases}$

I/O sequence:

t	0	1	2	3	4	5	6	7
x, z	$a, 1$	$b, 0$	$b, 1$	$a, 1$	$b, 0$	$a, 0$	$b, 1$	$a, 1$

Example 7.4: State description of odd/even

Input: $x(t) \in \{a, b\}$
Output: $z(t) \in \{0, 1\}$
State: $s(t) \in \{\text{EVEN, ODD}\}$
Initial state: s(0)= EVEN

Functions: Transition and output functions

PS	$x(t) = a$	$x(t) = b$
EVEN	EVEN, 1	ODD, 0
ODD	ODD, 0	EVEN, 1
	$NS,\ z(t)$	

Mealy and Moore machines

Mealy machine

$$z(t) = H(s(t), x(t))$$

$$s(t+1) = G(s(t), x(t))$$

Moore machine

$$z(t) = H(s(t))$$

$$s(t+1) = G(s(t), x(t))$$

Example 7.5: Moore sequential system

Input:	$x(t) \in \{a, b, c\}$
Output:	$z(t) \in \{0, 1\}$
State:	$s(t) \in \{S_0, S_1, S_2, S_3\}$
Initial state:	$s(0) = S_0$

Functions: Transition and output functions:

PS	Input			
	a	b	c	
S_0	S_0	S_1	S_1	0
S_1	S_2	S_0	S_1	1
S_2	S_2	S_3	S_0	1
S_3	S_0	S_1	S_2	0
	NS			Output

Representation of state-transition and output functions

- State diagram

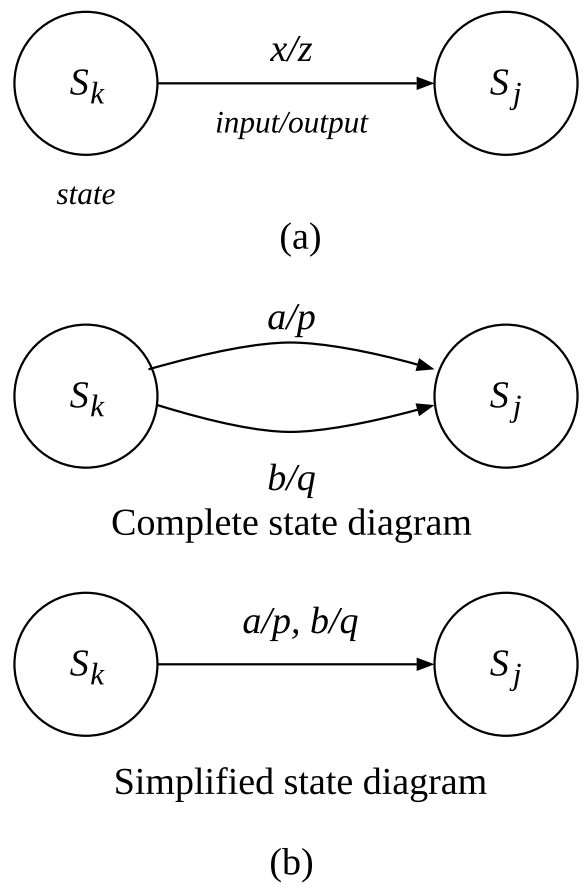


Figure 7.4: (a) State diagram representation. (b) Simplified state diagram notation.

Example 7.6

Functions: The transition and output functions are

$s(t)$	$x(t)$	
	a	b
S_0	S_1, p	S_2, q
S_1	S_1, p	S_0, p
S_2	S_1, p	S_2, p
	$s(t+1), z(t)$	

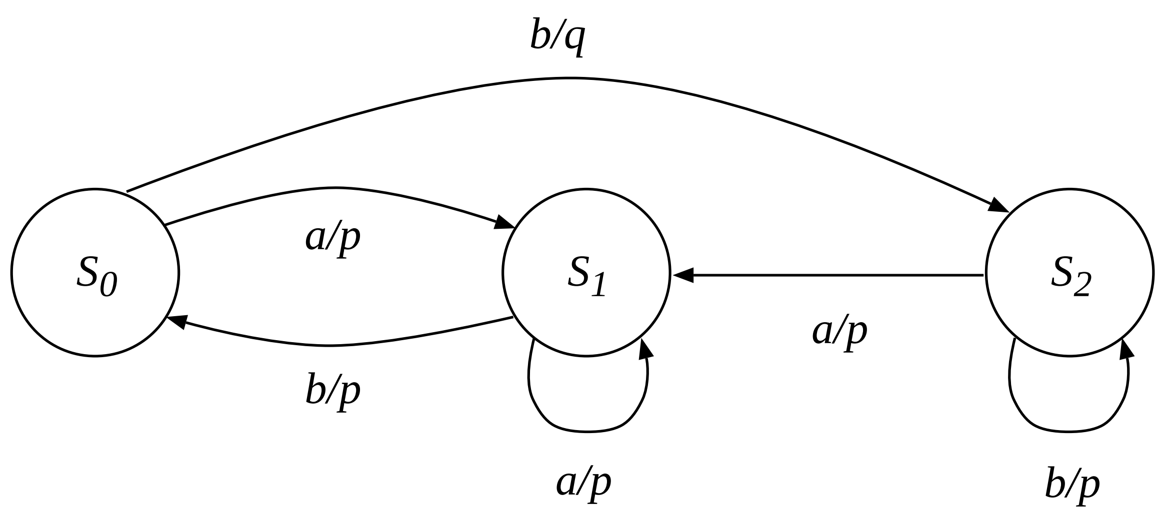


Figure 7.5: State diagram for Example 7.6.

State diagram for a Moore machine

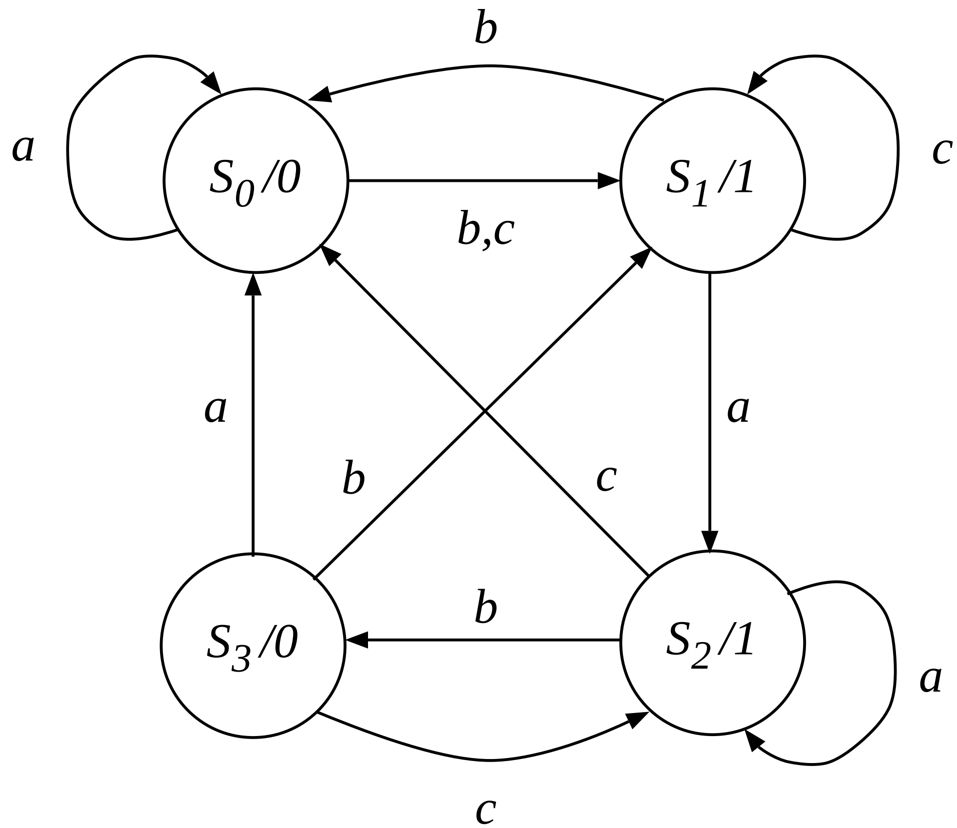


Figure 7.6: State diagram for Example 7.5

Example 7.7: Use of conditional expressions

Input:	$x(t) \in \{0, 1, 2, 3\}$
Output:	$z(t) \in \{a, b\}$
State:	$s(t) \in \{S_0, S_1\}$
Initial state:	$s(0) = S_0$

Functions: The transition and output functions are

$$s(t+1) = \begin{cases} S_0 \text{ } \mathbf{if} \text{ } (s(t) = S_0 \\ \qquad \mathbf{and} \text{ } [x(t) = 0 \text{ } \mathbf{or} \text{ } x(t) = 2]) \\ \qquad \mathbf{or} \text{ } (s(t) = S_1 \text{ } \mathbf{and} \text{ } x(t) = 3) \\ S_1 \text{ } \mathbf{otherwise} \end{cases}$$

$$z(t) = \begin{cases} a \text{ } \mathbf{if} \text{ } s(t) = S_0 \\ b \text{ } \mathbf{if} \text{ } s(t) = S_1 \end{cases}$$

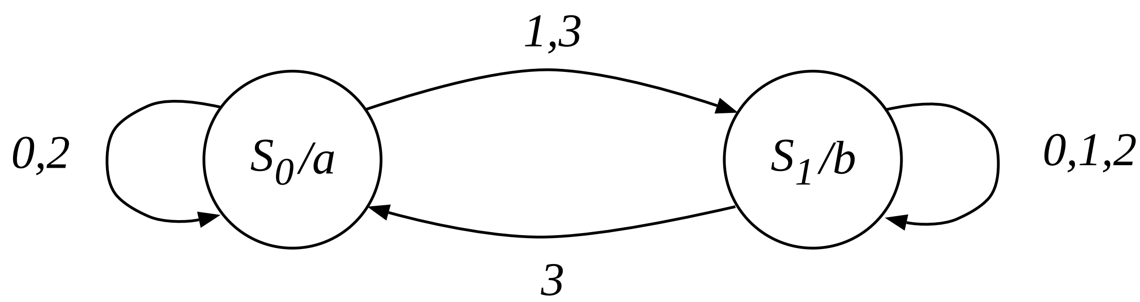


Figure 7.7: State diagram for Example 7.7

State names

Example 7.8: Integers as state names

A modulo-64 counter

Input:	$x(t) \in \{0, 1\}$
Output:	$z(t) \in \{0, 1, 2, \ldots, 63\}$
State:	$s(t) \in \{0, 1, 2, \ldots, 63\}$
Initial state:	$s(0) = 0$

Functions: The transition and output functions are

$$\begin{aligned} s(t+1) &= [s(t) + x(t)] \bmod 64 \\ z(t) &= s(t) \end{aligned}$$

Example 7.9: Vectors as state names

Input: $e(t) \in \{1, 2, \ldots, 55\}$
Output: $z(t) \in \{0, 1, 2, \ldots, 55\}$
State: $\underline{s}(t) = (s_{55}, \ldots, s_1), \quad s_i \in \{0, 1, 2, \ldots, 99\}$
Initial state: $\underline{s}(0) = (0, 0, \ldots, 0)$

Functions: The transition and output functions are

$$s_i(t+1) = \begin{cases} [s_i(t) + 1] \bmod 100 & \textbf{if} \quad e(t) = i \\ & i = 1, 2, \ldots, 55 \\ s_i(t) & \textbf{otherwise} \end{cases}$$

$$z(t) = \begin{cases} i & \textbf{if} \quad e(t) = i \textbf{ and } s_i(t) = 99 \\ 0 & \textbf{otherwise} \end{cases}$$

Time behavior and finite-state machines

- State description $\Rightarrow$ I/O sequence (Example 7.10)

Initial state: $s(0) = S_2$
Functions: Transition and output functions are

PS	$x(t)$			
	a	b	c	
S_0	S_0	S_1	S_1	p
S_1	S_2	S_0	S_1	q
S_2	S_2	S_3	S_0	q
S_3	S_0	S_1	S_2	p
	NS			$z(t)$

t	0	1	2	3	4
x	a	b	c	a	
s	S_2	S_2	S_3	S_2	S_2
z	q	q	p	q	

Time behavior $\Rightarrow$ state description

- Not all time-behaviors are realizable:

$$z(t) = \begin{cases} 1 \ \textbf{if} \ \ x(0,t) \text{ has same number of } 0'\text{s and } 1'\text{s} \\ 0 \ \textbf{otherwise} \end{cases}$$

$s(t)$ = difference between number of 1's and 0's

$$s(t+1) = \begin{cases} s(t)+1 \ \textbf{if} \ \ x(t) = 1 \\ s(t)-1 \ \textbf{otherwise} \end{cases}$$

$$z(t) = \begin{cases} 1 \ \textbf{if} \ \ s(t) = 0 \\ 0 \ \textbf{otherwise} \end{cases}$$

$\Rightarrow$ difference unbounded: not a finite-state system

Procedure for obtaining FSM from time behavior

1. Determine a set of states representing required events
2. Determine the transition function
3. Determine the output function

- Example 7.11

Input:	$x(t) \in \{0, 1\}$
Output:	$z(t) \in \{0, 1\}$
Function:	$z(t) = \begin{cases} 1 & \textbf{if} \quad x(t-3, t) = 1101 \\ 0 & \textbf{otherwise} \end{cases}$

- pattern detector $\Rightarrow$ detect subpatterns

State	indicates that
S_{init}	Initial state; also no subpattern
S_1	First symbol (1) of pattern has been detected
S_{11}	Subpattern 11 has been detected
S_{110}	Subpattern 110 has been detected

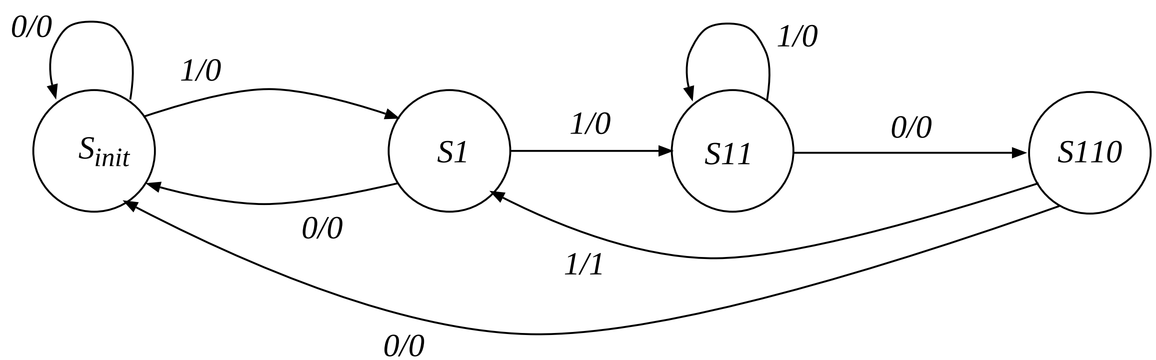


Figure 7.8: State diagram for Example 7.11

Finite-memory sequential systems

$$z(t) = F(x(t-m+1,t))$$

Example 7.12:

$$z(t) = \begin{cases} p & \textbf{if} \quad x(t-3,t) = aaba \\ q & \textbf{otherwise} \end{cases}$$

$\Rightarrow$ finite memory of length four

- All finite-memory machines are FS systems
- Not all FS systems are finite memory

$$z(t) = \begin{cases} 1 & \textbf{if} \quad \text{number of } 1'\text{s in } x(0,t) \text{ is even} \\ 0 & \textbf{otherwise} \end{cases}$$

Controllers

- The state description is primary
- FSM producing control signals
- Control signals determine actions performed in other parts of system
- *Autonomous*: Fixed sequence of states, independent of inputs

Autonomous controller

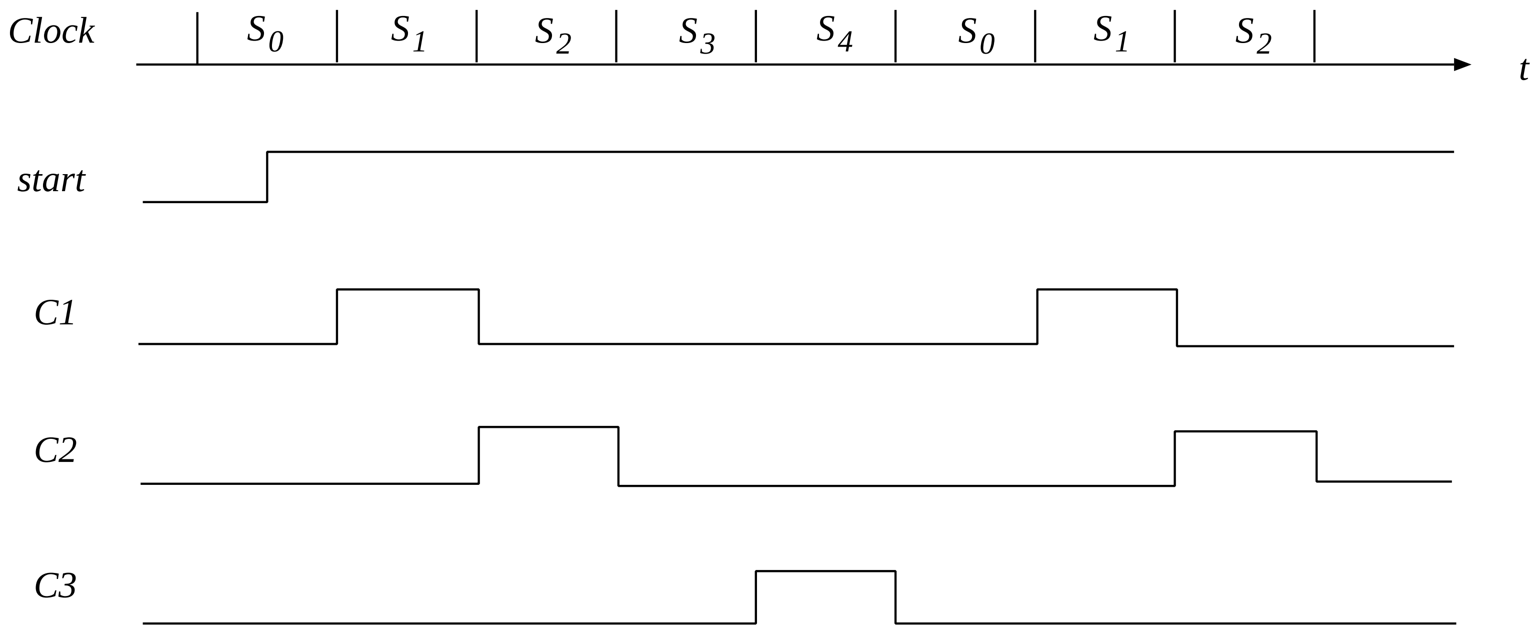


Figure 7.9: Autonomous controller: Timing diagram.

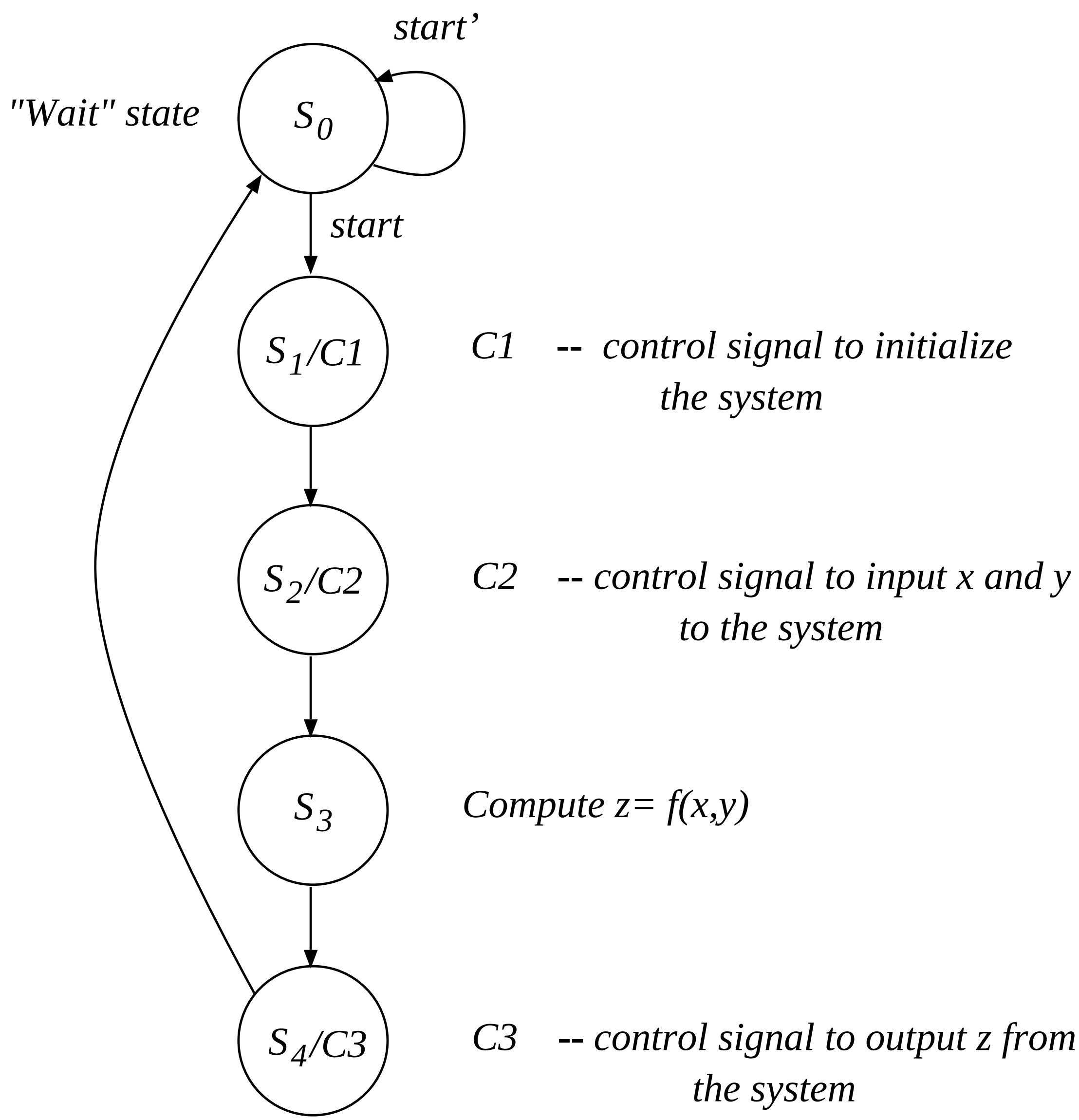
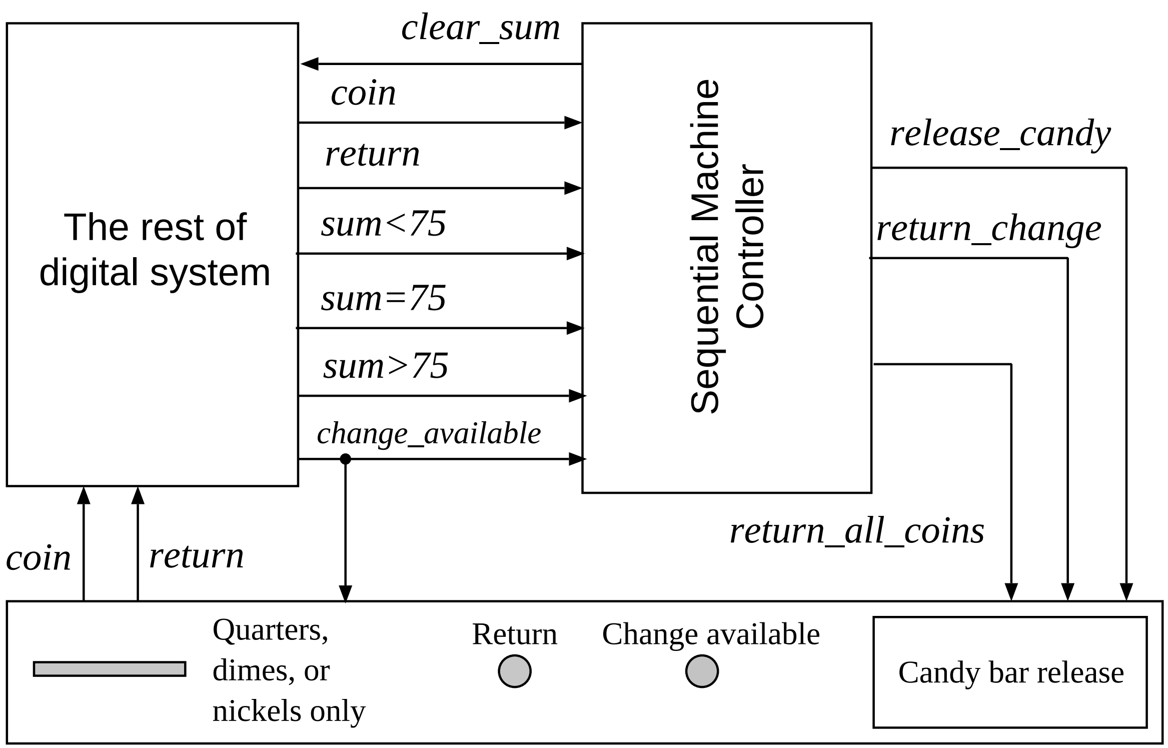


Figure 7.10: Autonomous controller: State diagram.

General controller



Note: *coin· return = 0*

Figure 7.11: Controller for simple vending machine: Block diagram.

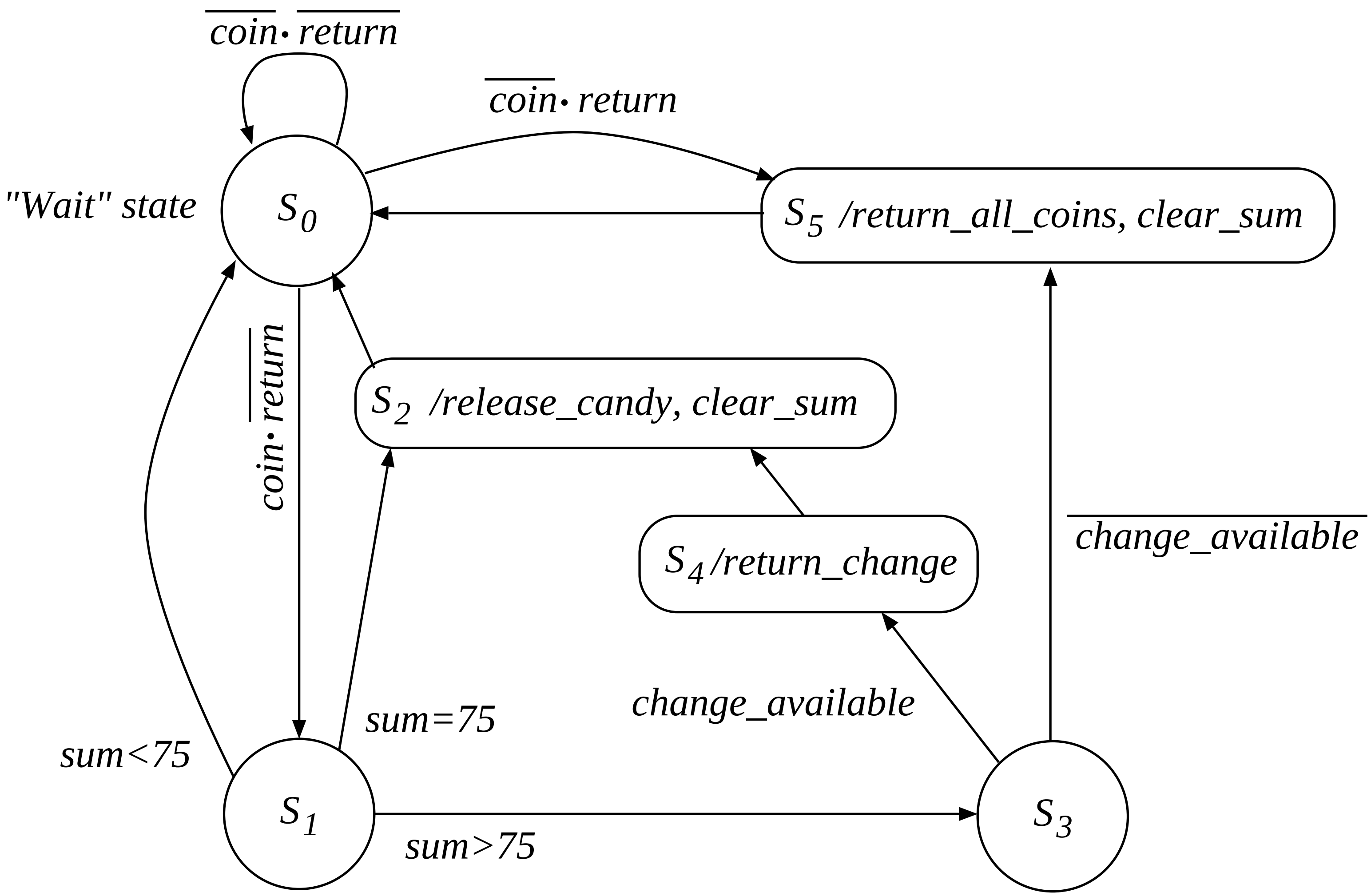


Figure 7.12: Controller for simple vending machine: State diagram.

Equivalent sequential systems: same time behavior

Input: $x(t) \in \{0,1\}$
Output: $z(t) \in \{0,1\}$
Function: $z(t) = \begin{cases} 1 \textbf{ if } \; x(t-2,t) = 101 \\ 0 \textbf{ otherwise} \end{cases}$

t	0	1	2	3	4	5	6	7	8
x	0	0	1	0	1	0	1	0	0
z	0	0	0	0	1	0	1	0	0

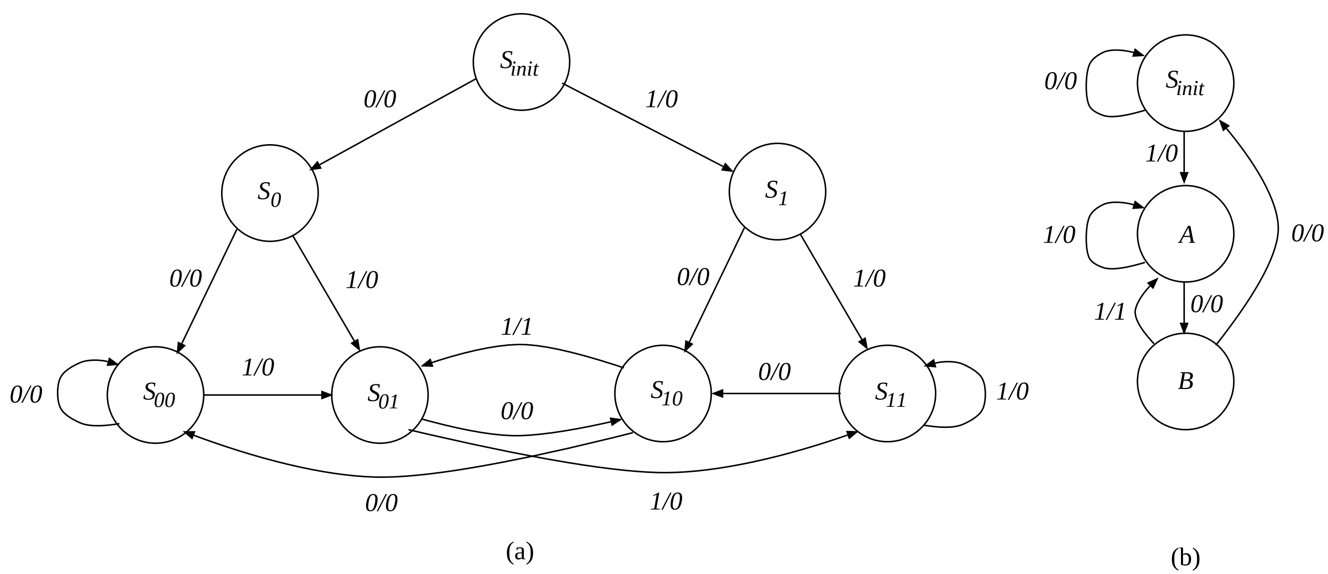


Figure 7.13: a) State diagram with redundant states; b) Reduced state diagram

Equivalent states

- k-distinguishable states: diff. output sequences

$$z(x(t, t+k-1), S_v) \neq z(x(t, t+k-1), S_w)$$

Example:

State	$x(3,7)$	$z(3,7)$
S_1	0210	0011
S_3	0210	0001

- k-equivalent states: not distinguishable for sequences of length k

  $-P_k$ – partition of states into k-equivalent classes

- Equivalent states

  – not distinguishable for any k

Example 7.14

Input: $x(t) \in \{a, b, c\}$
Output: $z(t) \in \{0, 1\}$
State: $s(t) \in \{A, B, C, D, E, F\}$
Initial state: $s(0) = A$

Function: The transition and output functions are

PS	$x = a$	$x = b$	$x = c$
A	$E, 0$	$D, 1$	$B, 0$
B	$F, 0$	$D, 0$	$A, 1$
C	$E, 0$	$B, 1$	$D, 0$
D	$F, 0$	$B, 0$	$C, 1$
E	$C, 0$	$F, 1$	$F, 0$
F	$B, 0$	$C, 0$	$F, 1$
		$NS,\ z$	

Example 7.14 (cont.)

- A and B are 1-distinguishable because

$$z(b, A) \neq z(b, B)$$

- A and C are 1-equivalent because

$$z(x(t), A) = z(x(t), C), \quad for\ all\ x(t) \in I$$

- A and C are also 2-equivalent because

$$\begin{aligned}
z(aa, A) &= z(aa, C) &= \mathsf{00} \\
z(ab, A) &= z(ab, C) &= \mathsf{01} \\
z(ac, A) &= z(ac, C) &= \mathsf{00} \\
z(ba, A) &= z(ba, C) &= \mathsf{10} \\
z(bb, A) &= z(bb, c) &= \mathsf{10} \\
z(bc, A) &= z(bc, C) &= \mathsf{11} \\
z(ca, A) &= z(ca, C) &= \mathsf{00} \\
z(cb, A) &= z(cb, C) &= \mathsf{00} \\
z(cc, A) &= z(cc, C) &= \mathsf{01}
\end{aligned}$$

Procedure to minimize number of states

Obtaining P_1: Directly from transition function

From P_i to P_{i+1} ...

1. P_{i+1} is a refinement of P_i (states (i+1)-equiv. must also be i-equiv.)

	possible	not possible
P_i	$(A,B,C)(D)$	
P_{i+1}	$(A,C)(B)(D)$	$(A,D)(B)(C)$

For (i+1)-equiv.

$$z(x(t,t+i),S_v) = z(x(t,t+i),S_w)$$

for arbitrary $x(t,t+i)$

Then

$$z(x(t,t+i-1),S_v) = z(x(t,t+i-1),S_w)$$

Example:

$$z(abcd,S_v) = z(abcd,S_w) = 1234$$

then

$$z(abc,S_v) = z(abc,S_w) = 123$$

2. Two states are (i+1)-equivalent if and only if
   a) they are i-equivalent, and
   b) for all $x \in I$, the corresponding next states are i-equivalent

If:

- Since the states are i-equiv. they are also 1-equiv.
- Then, if the next states are i-equiv, the states are (i+1)-equiv.

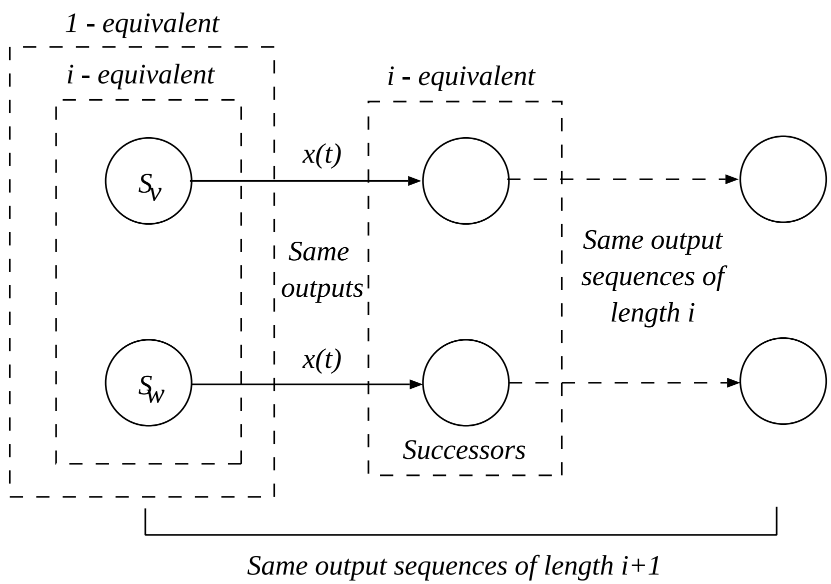


Figure 7.14: Illustration of $(i+1)$-equivalence relation.

Only if: By contradiction.

- If for some input, say a, the next states are not i-equivalent then there exists a sequence of length i, say T, such that these next states are distinguishable.

Therefore,

$$z(aT, S_v) \neq z(aT, S_w)$$

$\rightarrow S_v$ and S_w not (i+1)-equiv.

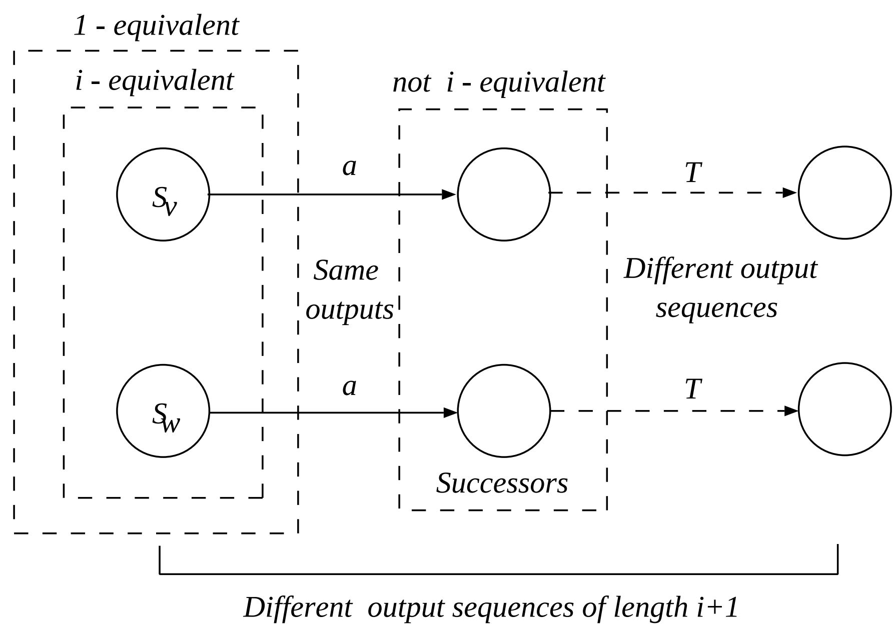


Figure 7.15: Illustration of $(i+1)$-equivalence relation.

When to stop? Stop when P_{i+1} is the same as P_i

- This is the equivalence partition
- The process always terminates

Procedure: Summary

1. Obtain P_1 (directly from the state table)
2. Obtain P_{i+1} from P_i
   by grouping states that are i-equivalent
   and whose corresponding successors are i-equivalent
3. Terminate when $P_{i+1} = P_i$
4. Write the reduced table

Example 7.15

PS	$x(t) = a$	$x(t) = b$	$x(t) = c$
A	0	1	0
B	0	0	1
C	0	1	0
D	0	0	1
E	0	1	0
F	0	0	1
		$NS,\ z$	

$$P_1 = (A, C, E) \ \ (B, D, F)$$

Example 7.15 (cont.)

PS	$x(t)$		
	a	b	c
A	E	D	B
C	E	B	D
E	C	F	F
	NS		

PS	$x(t)$		
	a	b	c
B	F	D	A
D	F	B	C
F	B	C	F
	NS		

$$P_2 = (A, C, E) \;\; (B, D) \;\; (F)$$

$$P_3 = (A, C) \;\; (E) \;\; (B, D) \;\; (F)$$

$$P_4 = P_3 = (A, C)\ (E)\ (B, D)\ (F)$$

Example 7.15 (cont.)

The minimal system:

PS	$x = a$	$x = b$	$x = c$
A	$E, 0$	$B, 1$	$B, 0$
B	$F, 0$	$B, 0$	$A, 1$
E	$A, 0$	$F, 1$	$F, 0$
F	$B, 0$	$A, 0$	$F, 1$
	$NS,\ z$		

Binary specification of sequential systems

- The state coding is called *state assignment*
- Coding functions:

Input	$C_I :$	$I \rightarrow \{0,1\}^n$
Output	$C_O :$	$O \rightarrow \{0,1\}^m$
State	$C_S :$	$S \rightarrow \{0,1\}^k$

Example 7.16

PS	$x = a$	$x = b$	$x = c$
A	$E, 0$	$B, 1$	$B, 0$
B	$F, 0$	$B, 0$	$A, 1$
E	$A, 0$	$F, 1$	$F, 0$
F	$B, 0$	$A, 0$	$F, 1$
	NS, z		

Binary coding

Input code

$x(t)$	$x_1(t)x_0(t)$
a	00
b	01
c	10

Output code

$z(t)$	
0	0
1	1

State assignment

$s(t)$	$s_1(t)s_0(t)$
A	00
B	01
E	10
F	11

The resulting binary specification:

$s_1(t)s_0(t)$	$x_1x_0 = 00$	$x_1x_0 = 01$	$x_1x_0 = 10$
00	$10, 0$	$01, 1$	$01, 0$
01	$11, 0$	$01, 0$	$00, 1$
10	$00, 0$	$11, 1$	$11, 0$
11	$01, 0$	$00, 0$	$11, 1$
	$s_1(t+1)s_0(t+1),\ z$		

Labeling arcs with switching expressions

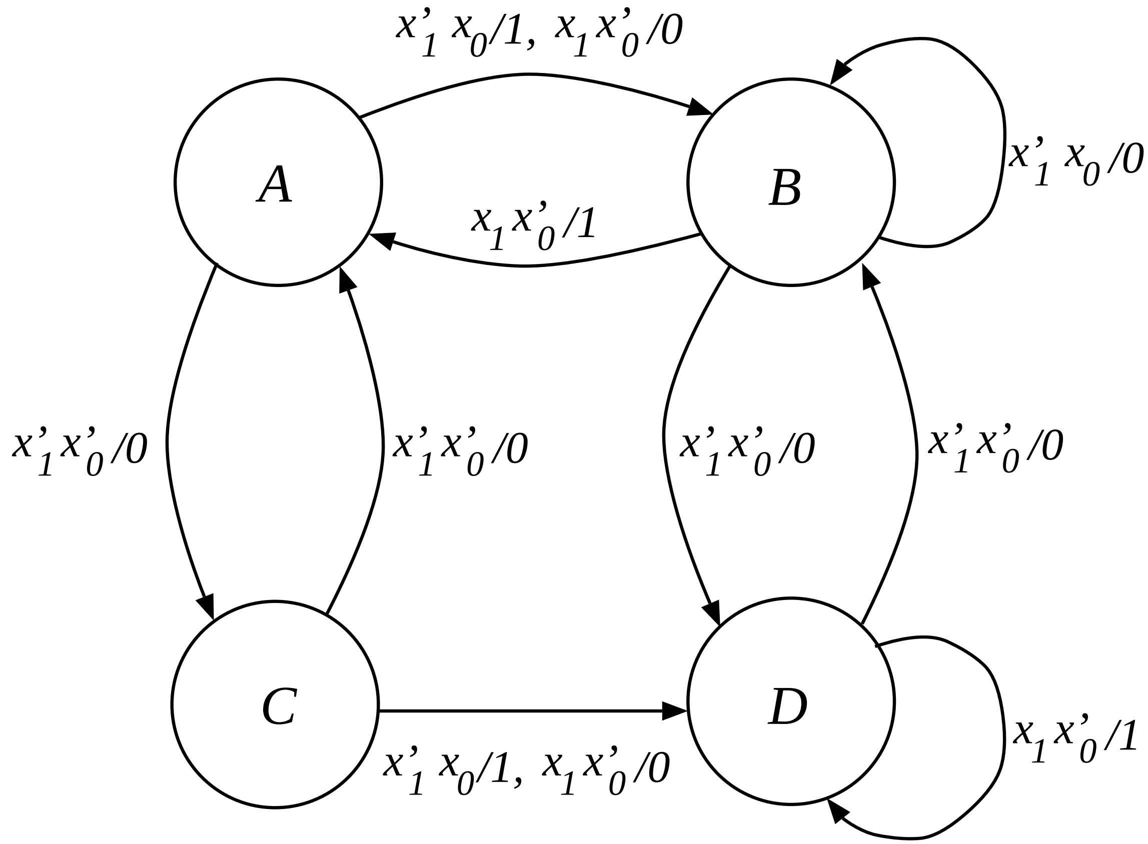


Figure 7.16: Switching expressions as arc labels

Specification of different types of sequential systems

Modulo-p counter

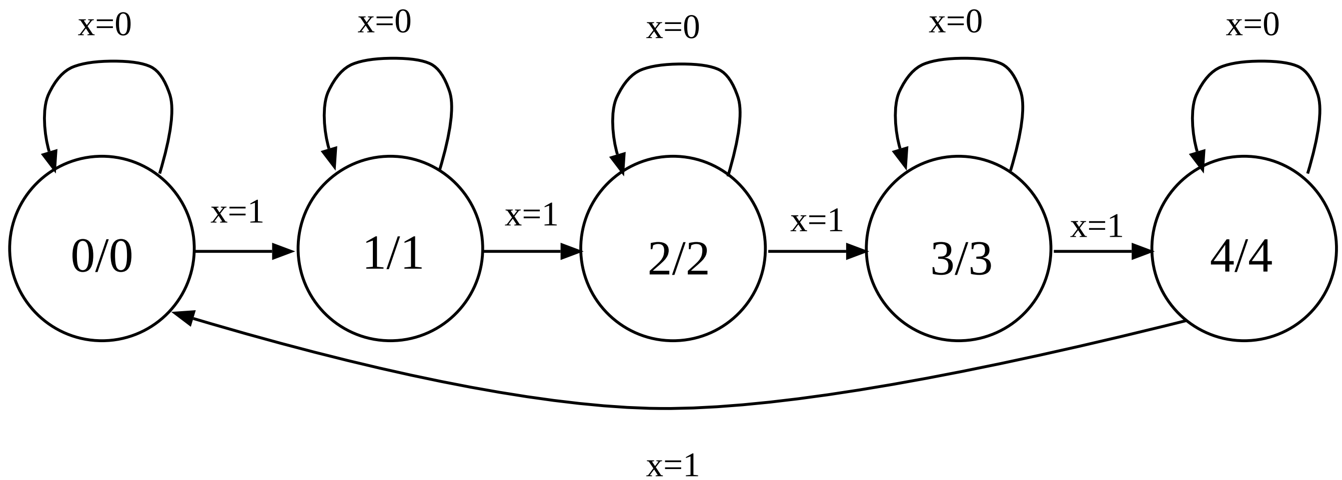


Figure 7.17: State diagram of a modulo-5 counter

Pattern recognizer

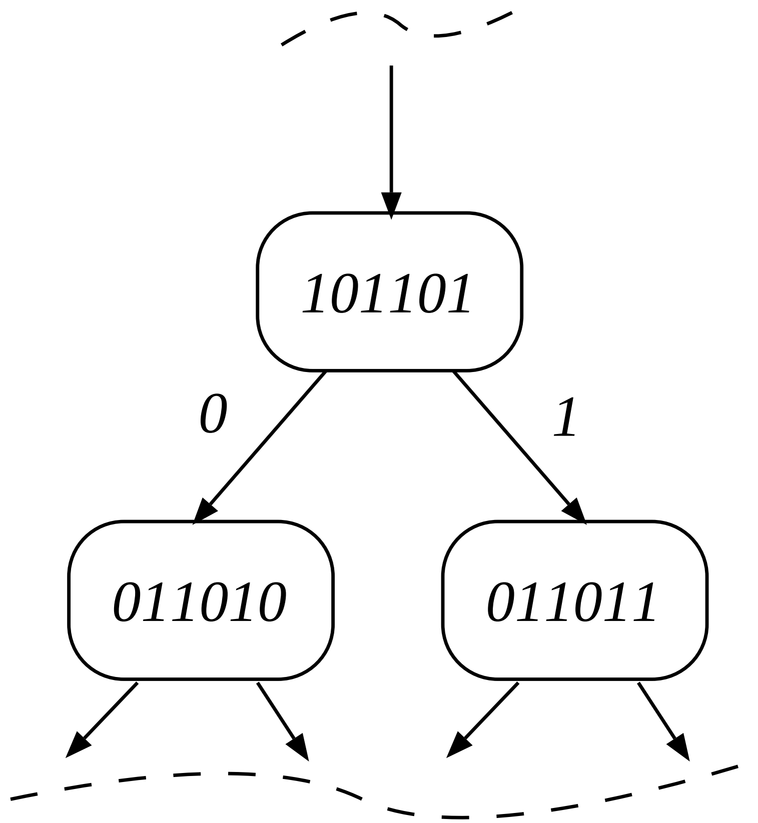


Figure 7.18: Fragment of state diagram of pattern recognizer

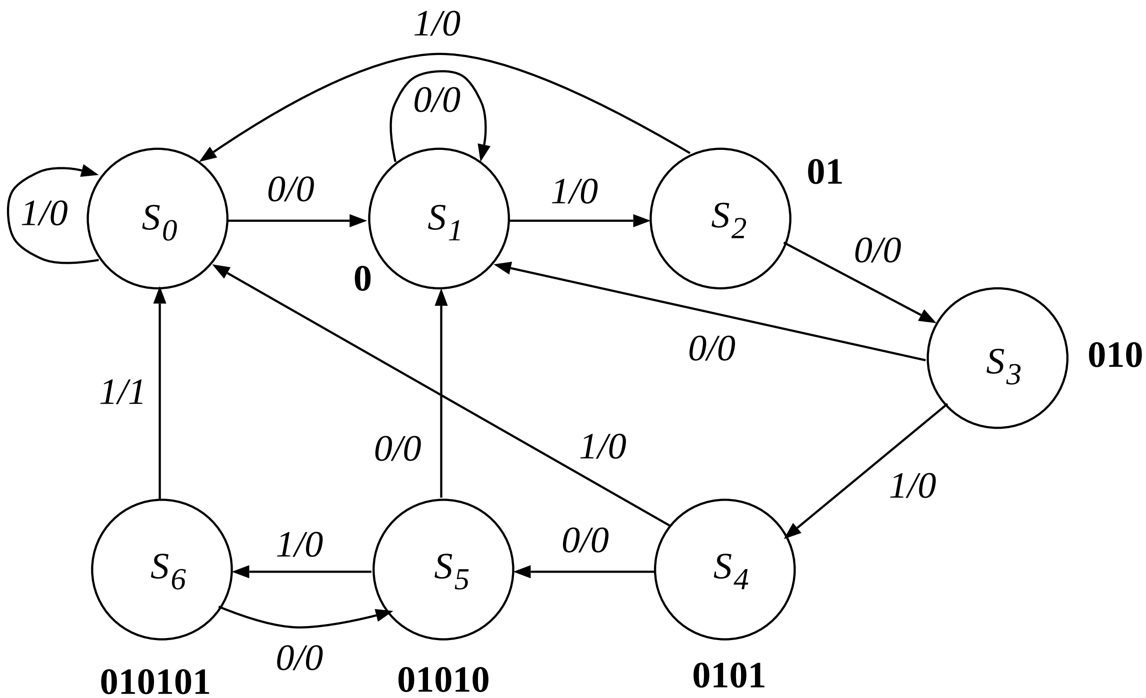


Figure 7.19: State diagram of a pattern recognizer