SPECIFICATION OF SEQUENTIAL SYSTEMS

- SYNCHRONOUS SEQUENTIAL SYSTEMS
- MEALY AND MOORE MACHINES
- TIME BEHAVIOR
- STATE MINIMIZATION

DEFINITION

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

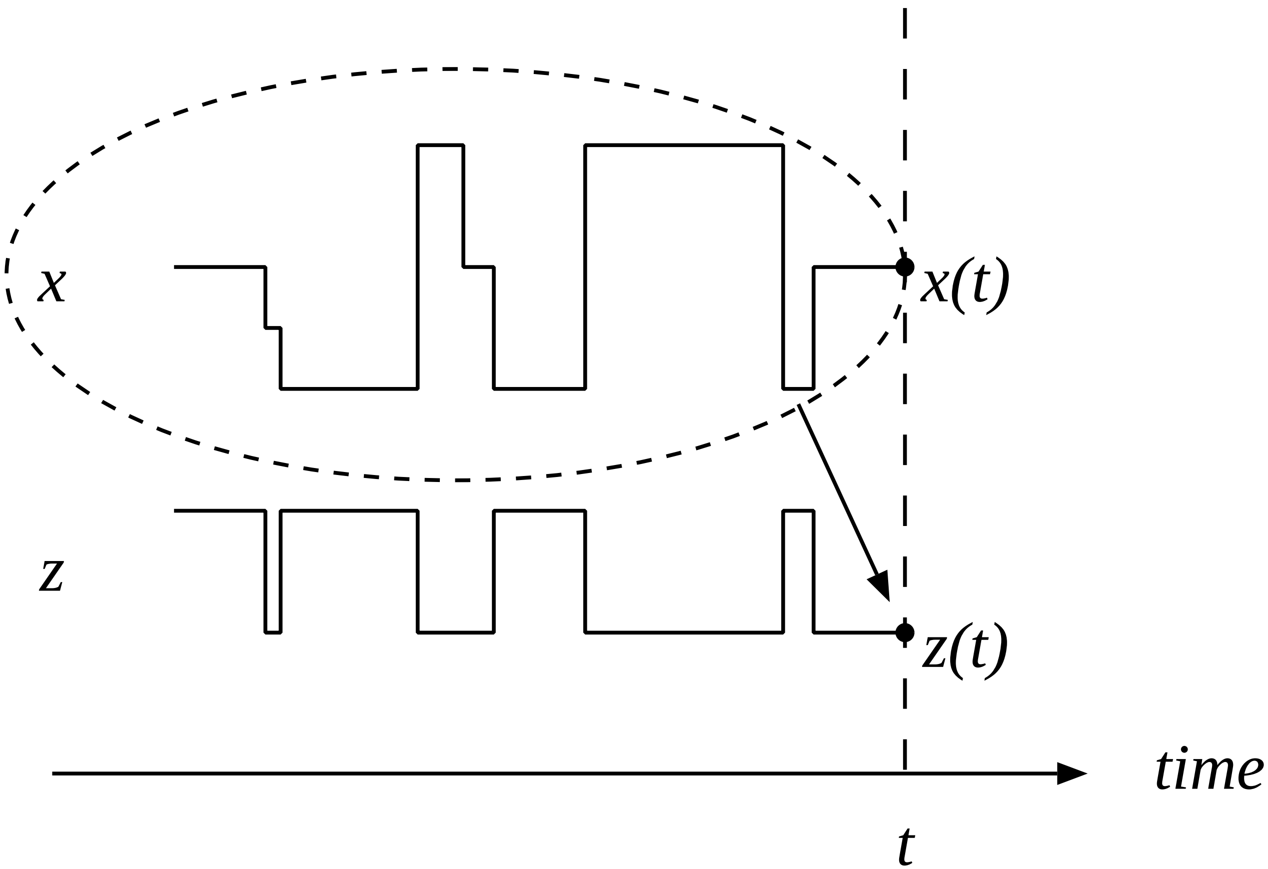

Figure 7.1: INPUT AND OUTPUT TIME FUNCTIONS.

SYNCHRONOUS AND ASYNCHRONOUS SYSTEMS

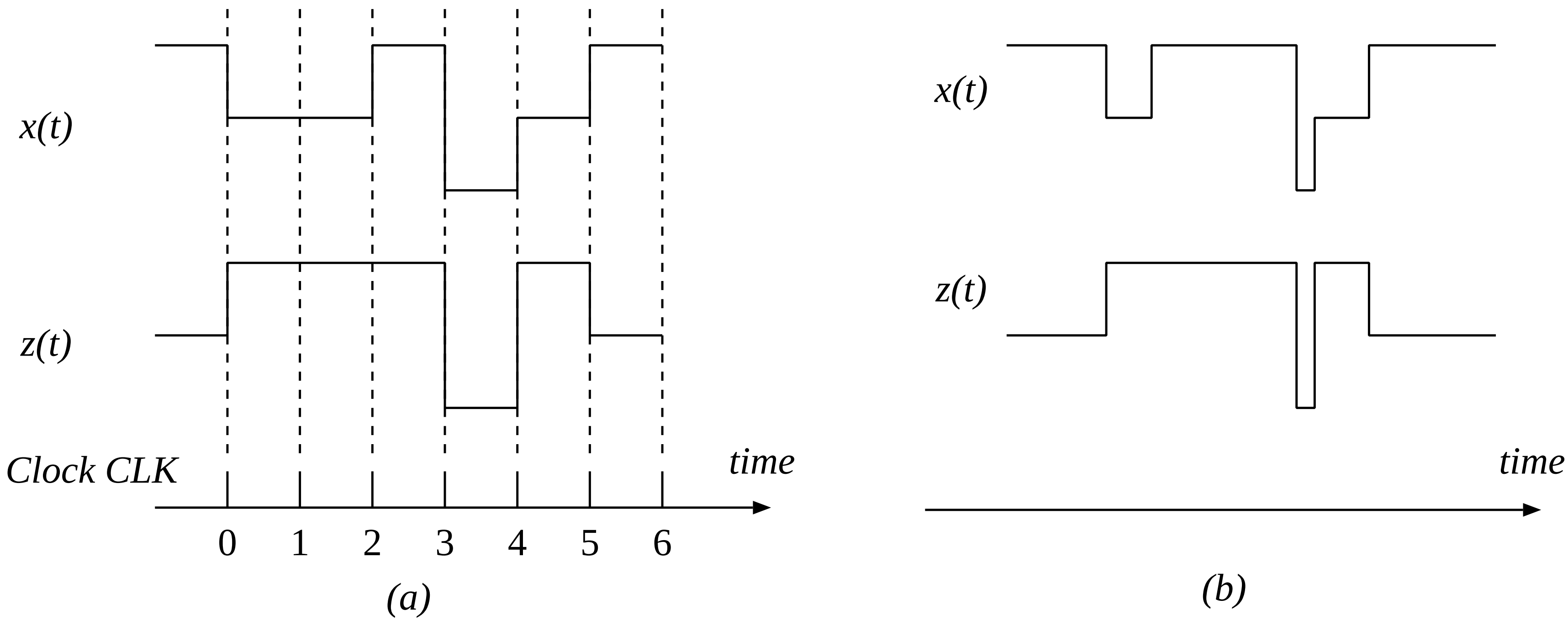

Figure 7.2: a) SYNCHRONOUS BEHAVIOR. b) ASYNCHRONOUS BEHAVIOR.

- CLOCK
- I/O SEQUENCE $x(t_1, t_2)$

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

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

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

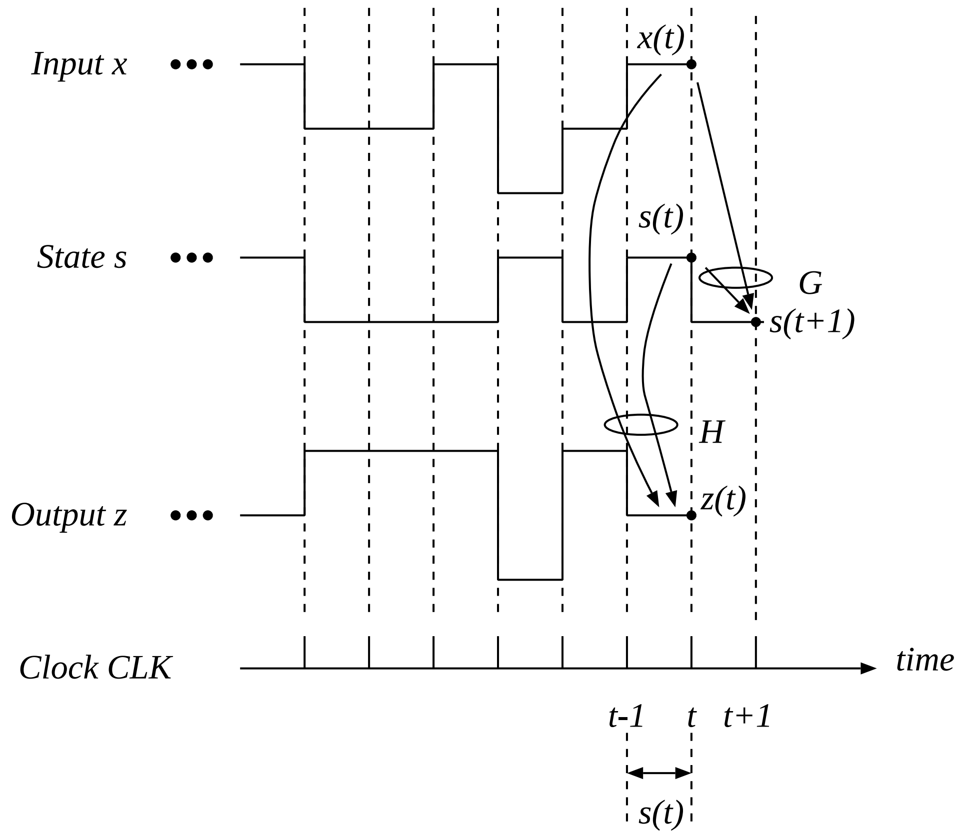

Figure 7.3: OUTPUT AND STATE TRANSITION FUNCTIONS

$$
\begin{array}{lrcl}
\text{State-transition function} & s(t+1) & = & G(s(t), x(t)) \\
\text{Output function} & z(t) & = & H(s(t), x(t))
\end{array}
$$

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

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

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

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

- EQUIVALENT IN CAPABILITIES

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

- STATE DIAGRAM

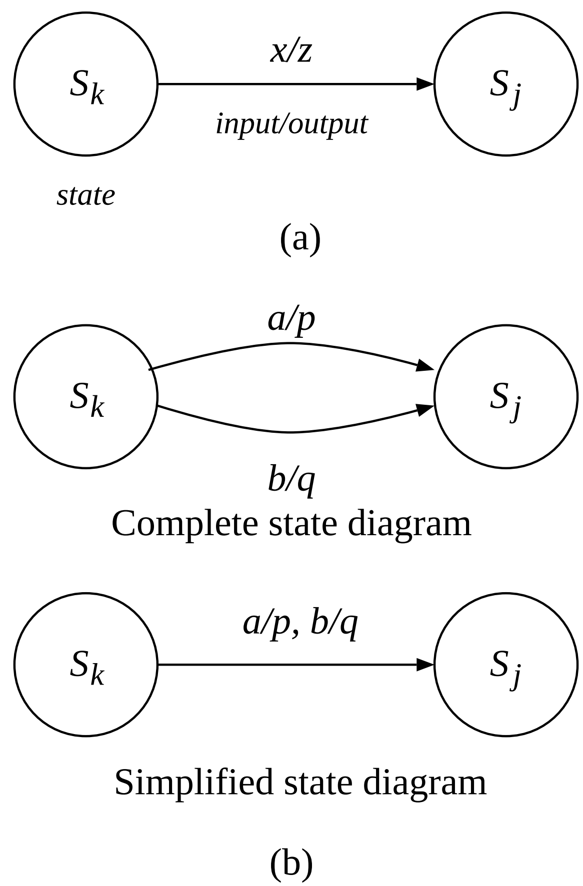

Figure 7.4: (a) STATE DIAGRAM REPRESENTATION. (b) SIMPLIFIED STATE DIAGRAM NOTATION.

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
	$s(t+1), z(t)$	

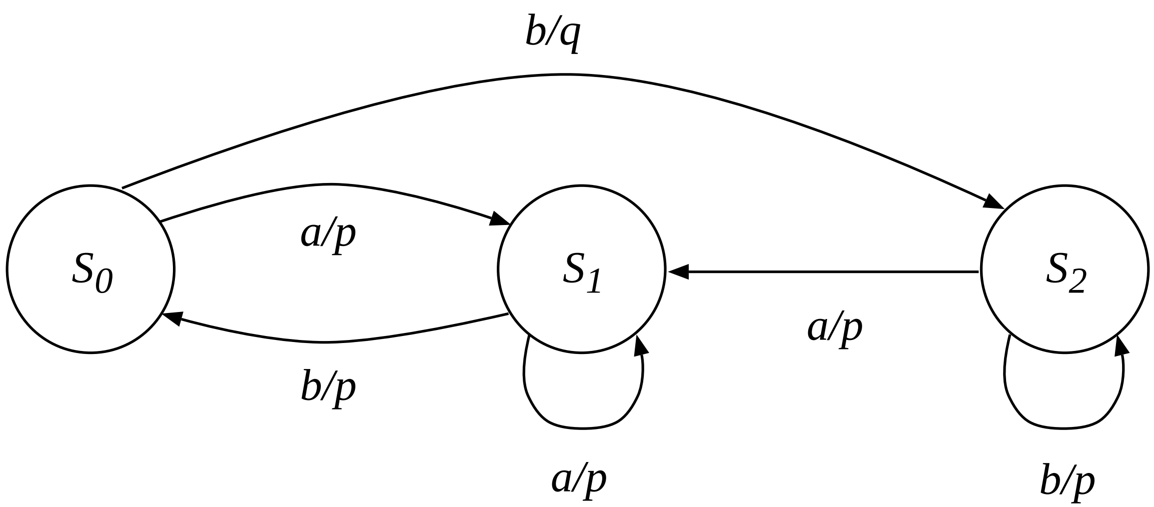

Figure 7.5: STATE DIAGRAM FOR EXAMPLE 7.6.

STATE DIAGRAM FOR A MOORE MACHINE

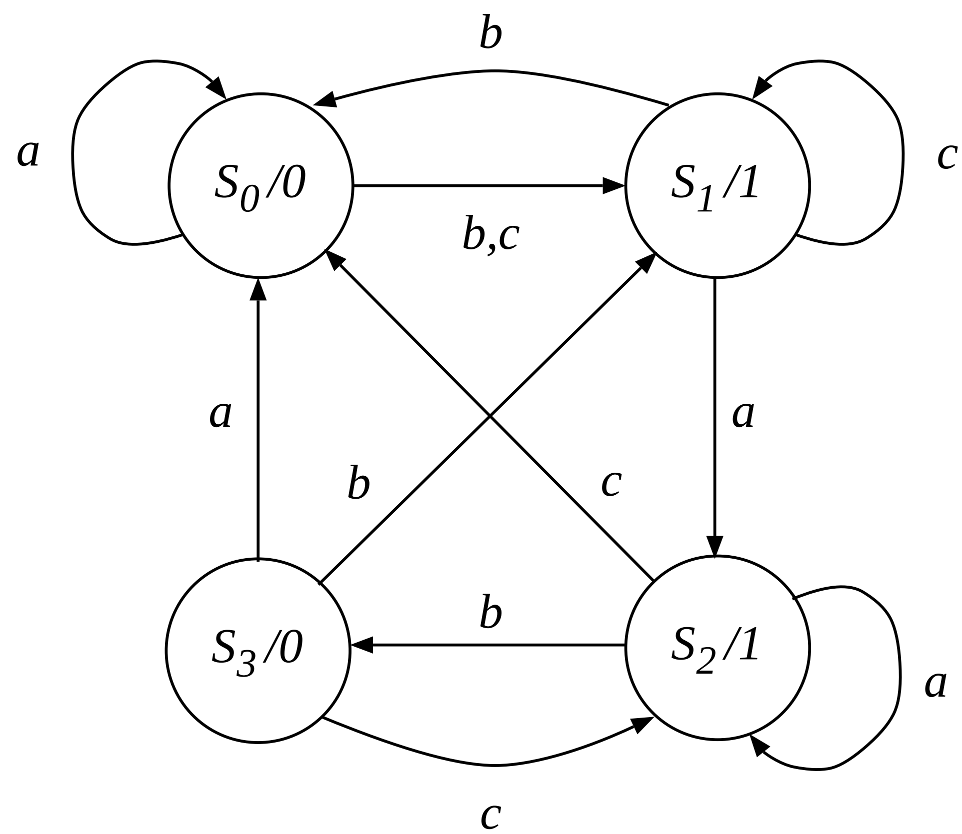

Figure 7.6: STATE DIAGRAM FOR EXAMPLE 7.5

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

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

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

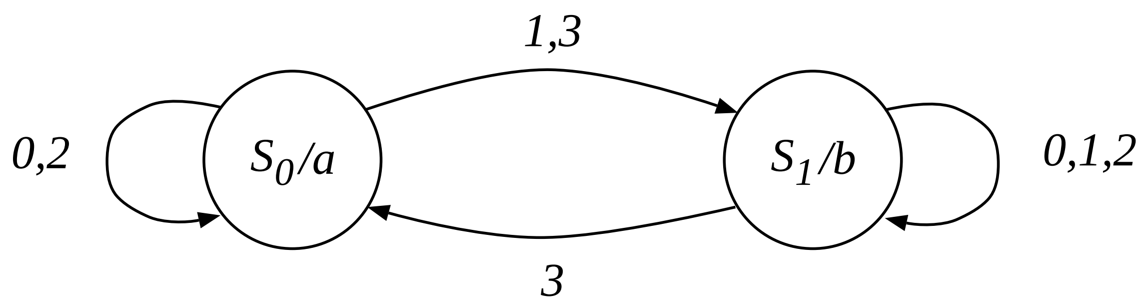

Figure 7.7: STATE DIAGRAM FOR EXAMPLE 7.7

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

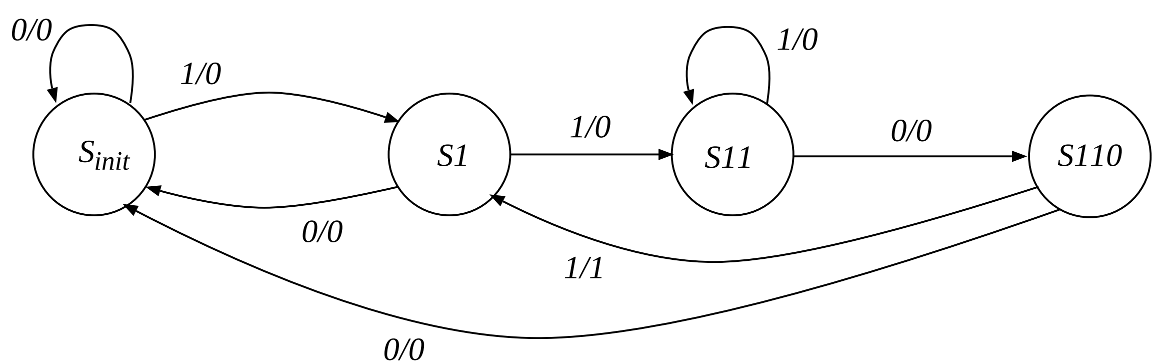

Figure 7.8: STATE DIAGRAM FOR Example 7.11

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

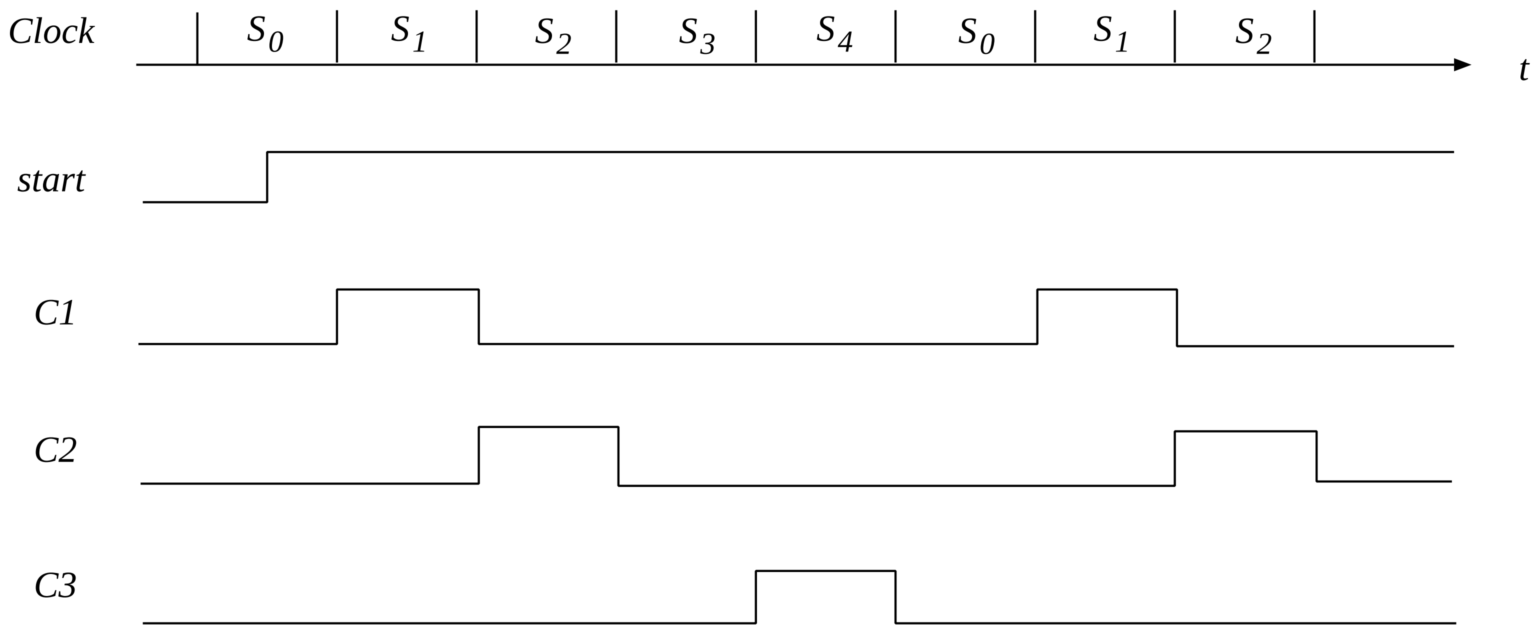

Figure 7.9: AUTONOMOUS CONTROLLER: TIMING DIAGRAM.

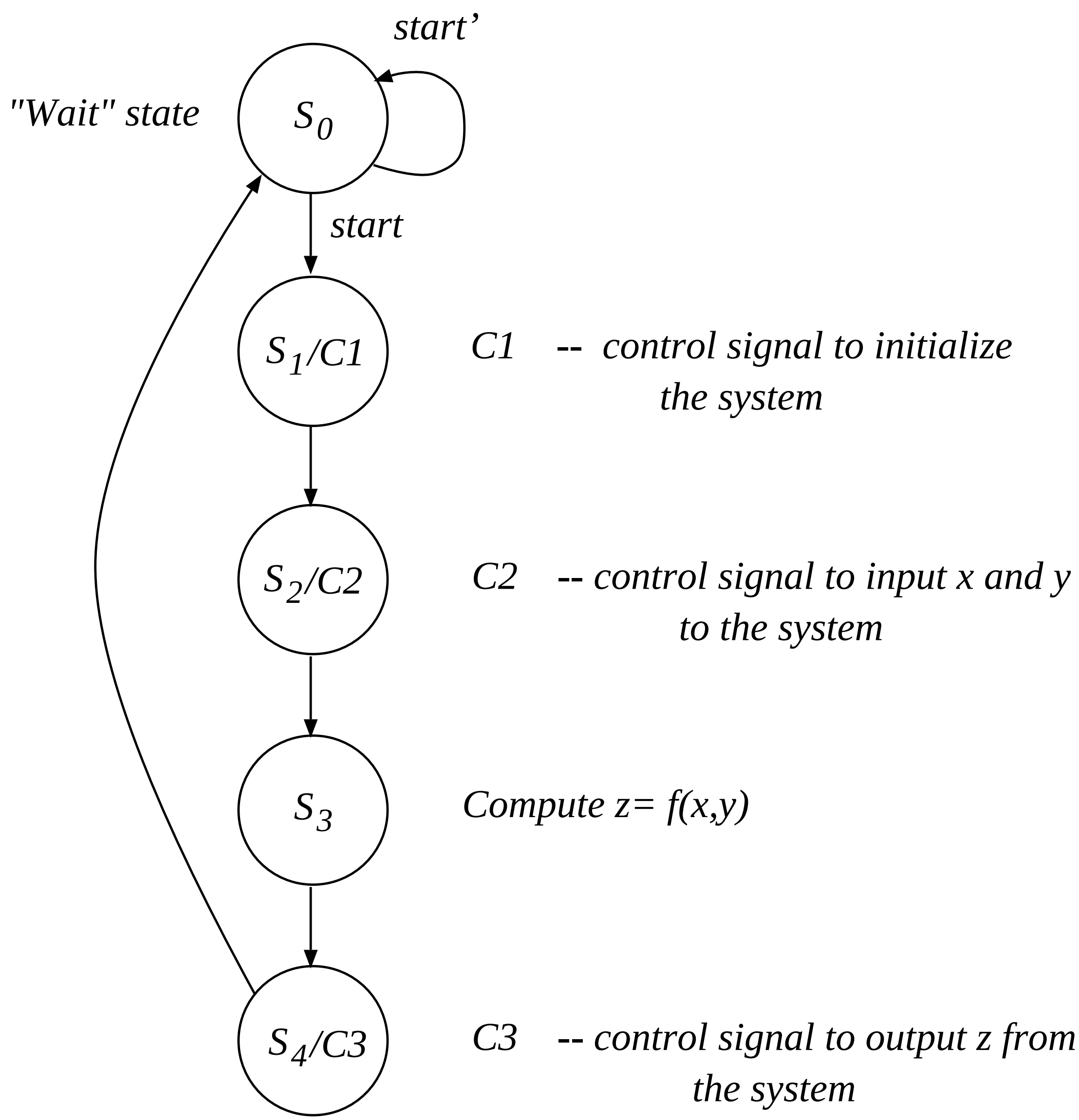

Figure 7.10: AUTONOMOUS CONTROLLER: STATE DIAGRAM.

GENERAL CONTROLLER

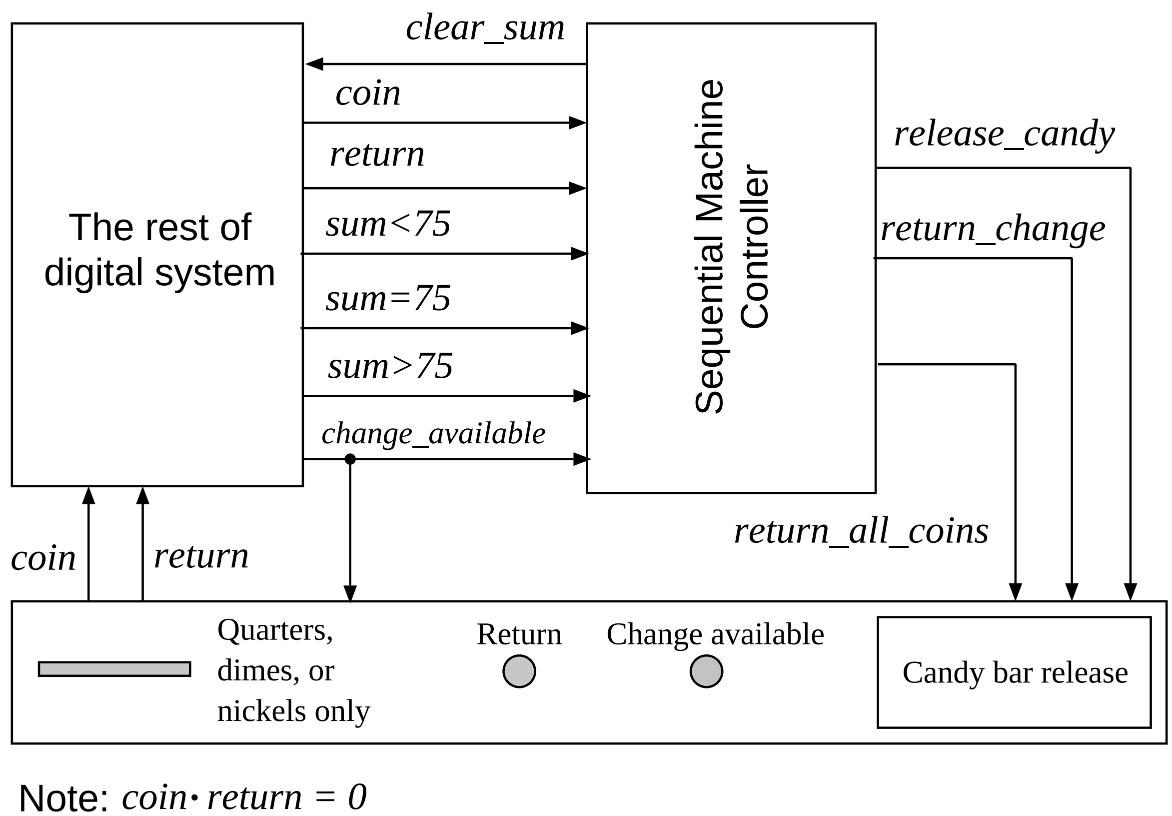

Figure 7.11: CONTROLLER FOR SIMPLE VENDING MACHINE: BLOCK DIAGRAM.

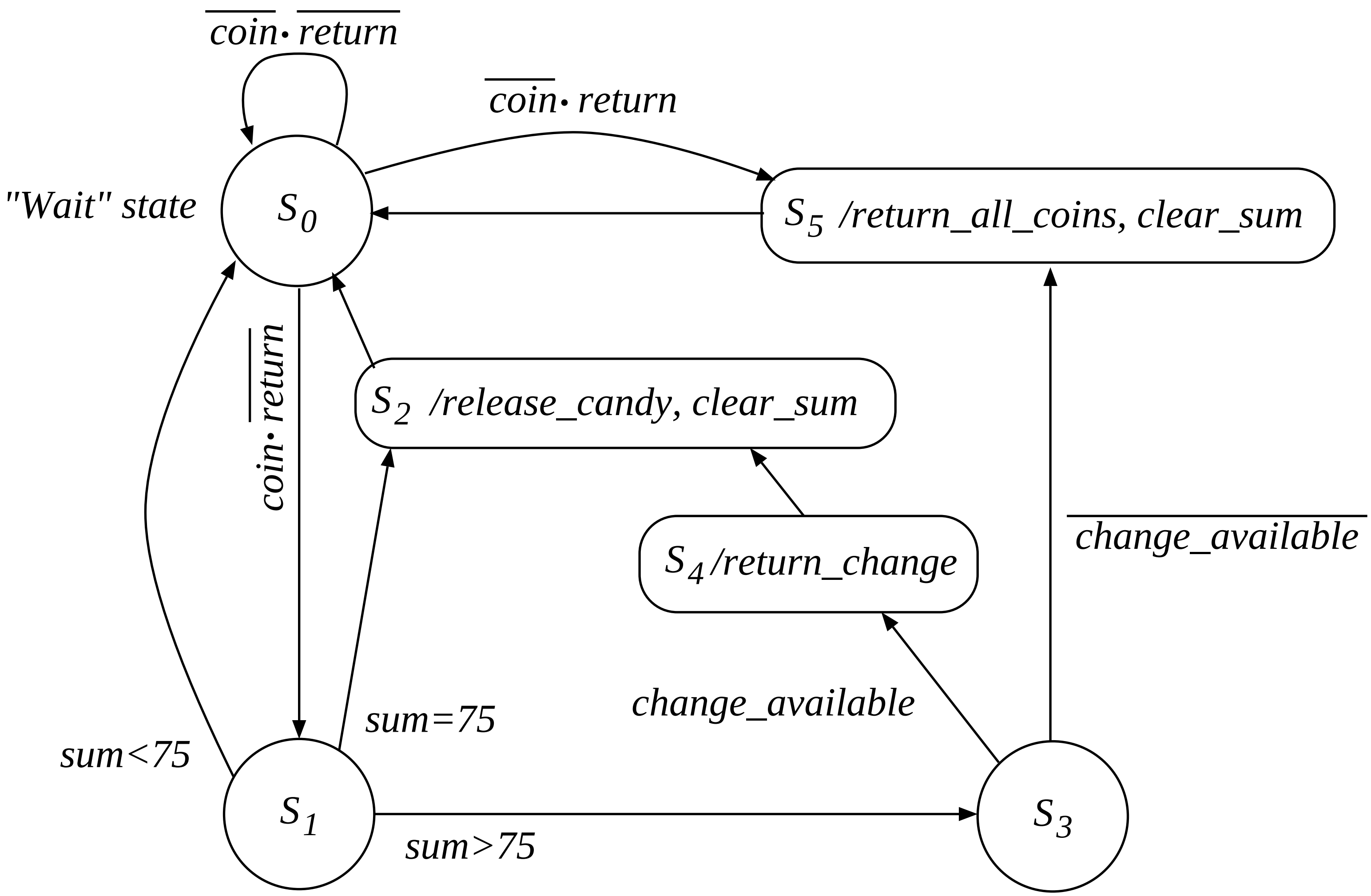

Figure 7.12: CONTROLLER FOR SIMPLE VENDING MACHINE: STATE DIAGRAM.

EQUIVALENT SEQUENTIAL SYSTEMS: SAME TIME BEHAVIOR

Input: $x(t) \in \{0,1\}$

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

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

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

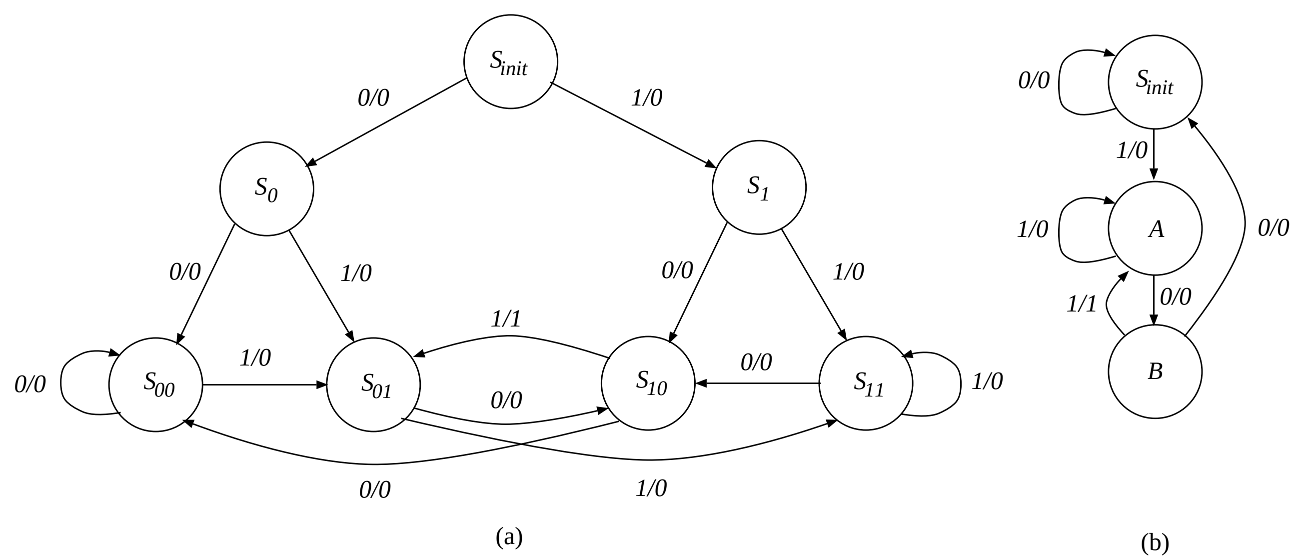

Figure 7.13: a) STATE DIAGRAM WITH REDUNDANT STATES; b) REDUCED STATE DIAGRAM

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: $s(0) = A$

Functions: TRANSITION AND OUTPUT

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 OUTPUT 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)-EQUIVALENT STATES S_v and S_w

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

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

$(i+1)$-EQUIVALENT STATES

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

   PROOF:

   IF PART:

   - SINCE THE STATES ARE i-EQUIVALENT, THEY ARE ALSO 1-EQUIVALENT
   - THEREFORE, IF THE NEXT STATES ARE i-EQUIVALENT, THE STATES ARE (i+1)-EQUIVALENT

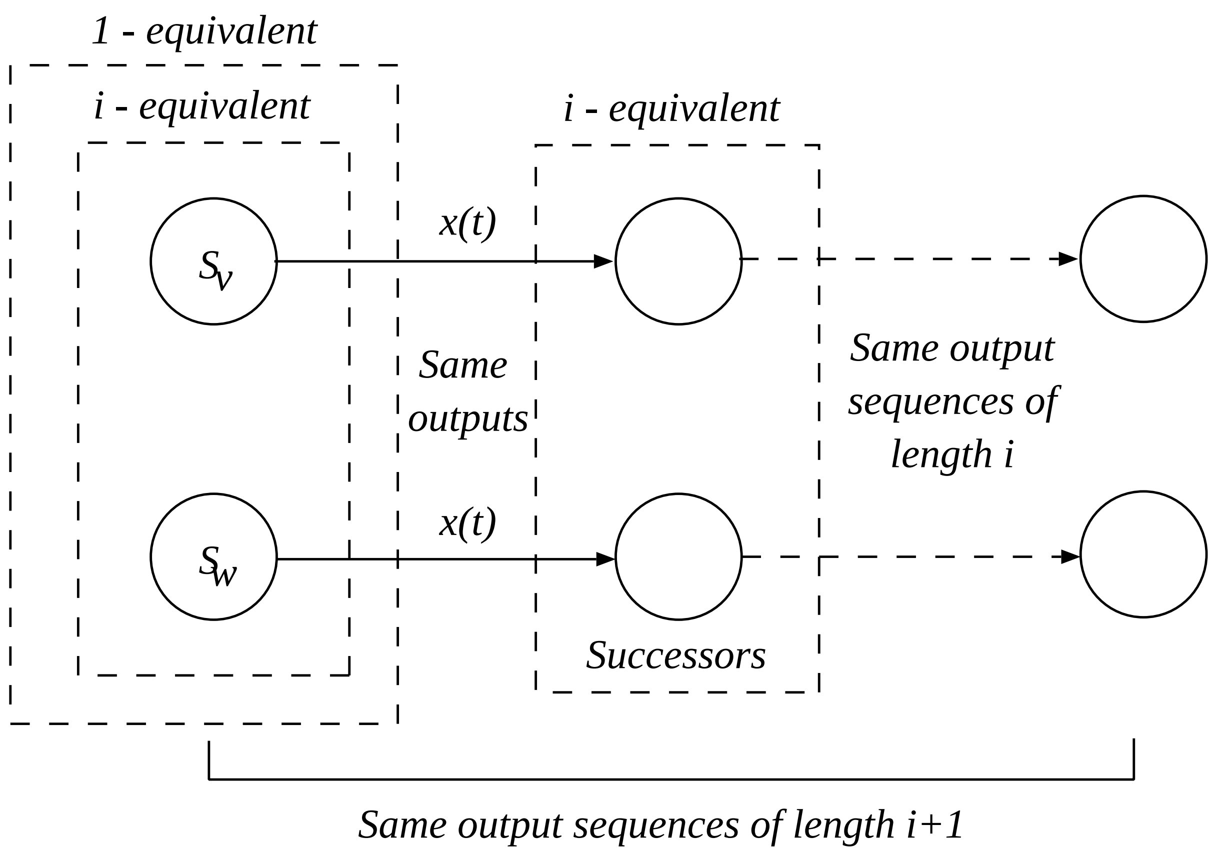

Figure 7.14: ILLUSTRATION OF $(i+1)$-EQUIVALENCE RELATION.

$(i+1)$-EQUIVALENT STATES (cont.)

ONLY IF PART: BY CONTRADICTION

- IF FOR SOME INPUT a THE NEXT STATES ARE NOT i-EQUIVALENT THEN THERE EXISTS A SEQUENCE T OF LENGTH i 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)-EQUIVALENT

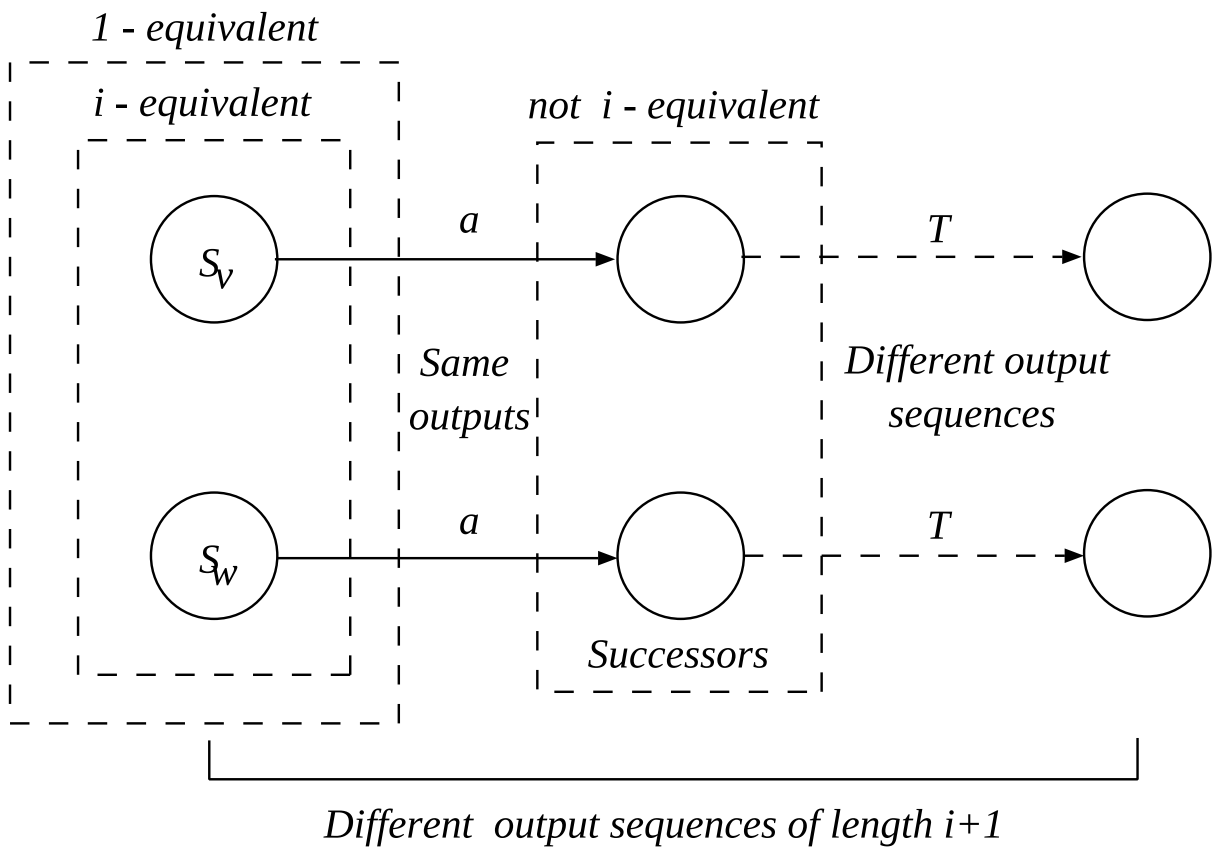

Figure 7.15: ILLUSTRATION OF $(i+1)$-EQUIVALENCE RELATION.

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 (look at the outputs)

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		

- 1-EQUIVALENT IF SAME "row pattern"

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

Example 7.15 (cont.)

- NUMBER THE CLASSES IN P_1
- TWO STATES ARE IN THE SAME CLASS OF P_2
  IF THEIR SUCCESSOR COLUMNS HAVE THE SAME NUMBERS

	1			2		
P_1	$(A,$	$C,$	$E)$	$(B,$	$D,$	$F)$
a	1	1	1	2	2	2
b	2	2	2	2	2	1
c	2	2	2	1	1	2

BY IDENTIFYING IDENTICAL COLUMNS OF SUCCESSORS, WE GET

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

Example 7.15 (cont.)

- SAME PROCESS TO OBTAIN THE NEXT PARTITION:

	1			2		3
P_2	$(A,$	$C,$	$E)$	$(B,$	$D),$	(F)
a	1	1	1	3	3	
b	2	2	3	2	2	
c	2	2	3	1	1	

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

- SIMILARLY, WE DETERMINE $P_4 = (A, C)\ (E)\ (B, D)\ (F)$

BECAUSE $P_4 = P_3$ THIS IS ALSO THE EQUIVALENCE PARTITION P

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

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

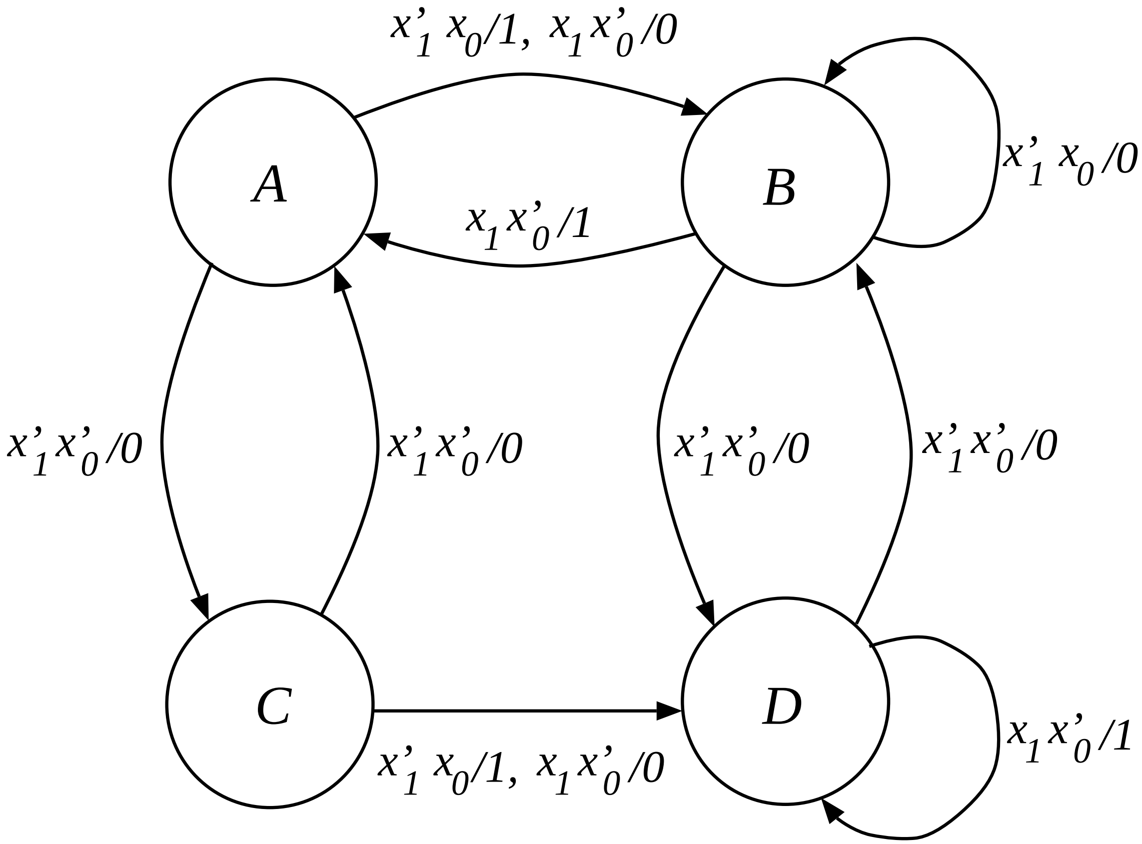

Figure 7.16: SWITCHING EXPRESSIONS AS ARC LABELS

SPECIFICATION OF DIFFERENT TYPES OF SEQUENTIAL SYSTEMS

MODULO-p COUNTER: 0, 1, 2, ..., p-1, 0, 1, ...

$$
\begin{aligned}
z(t) &= [\sum_{i=0}^{t} x(i)] \; mod \; p \\
s(t+1) &= [s(t) + x(t)] \; mod \; p \\
z(t) &= s(t) \quad \text{(if same coding)}
\end{aligned}
$$

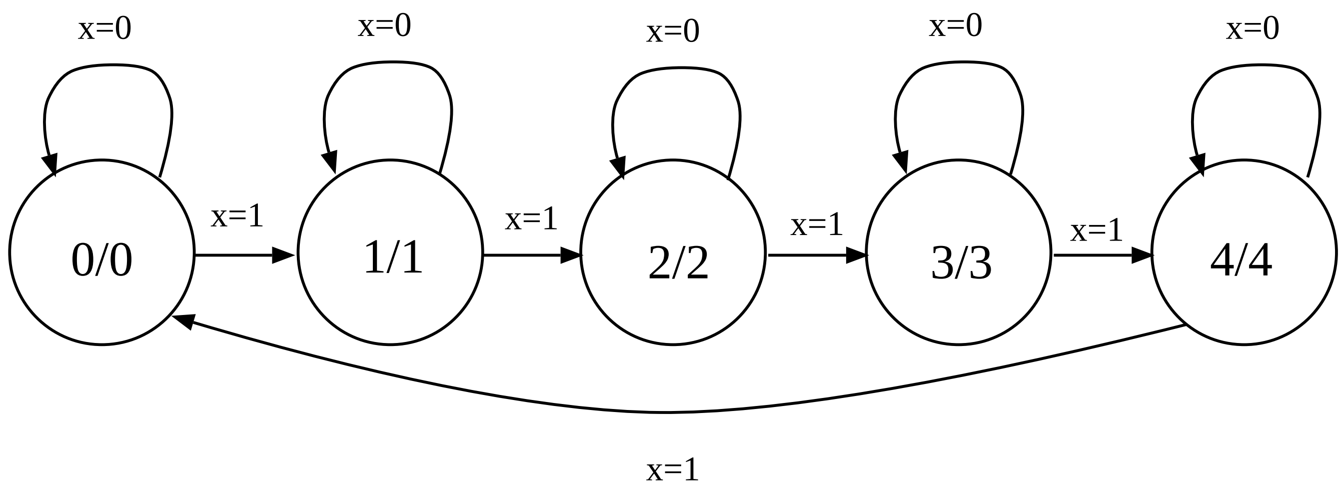

Figure 7.17: STATE DIAGRAM OF A MODULO-5 COUNTER

PATTERN RECOGNIZER

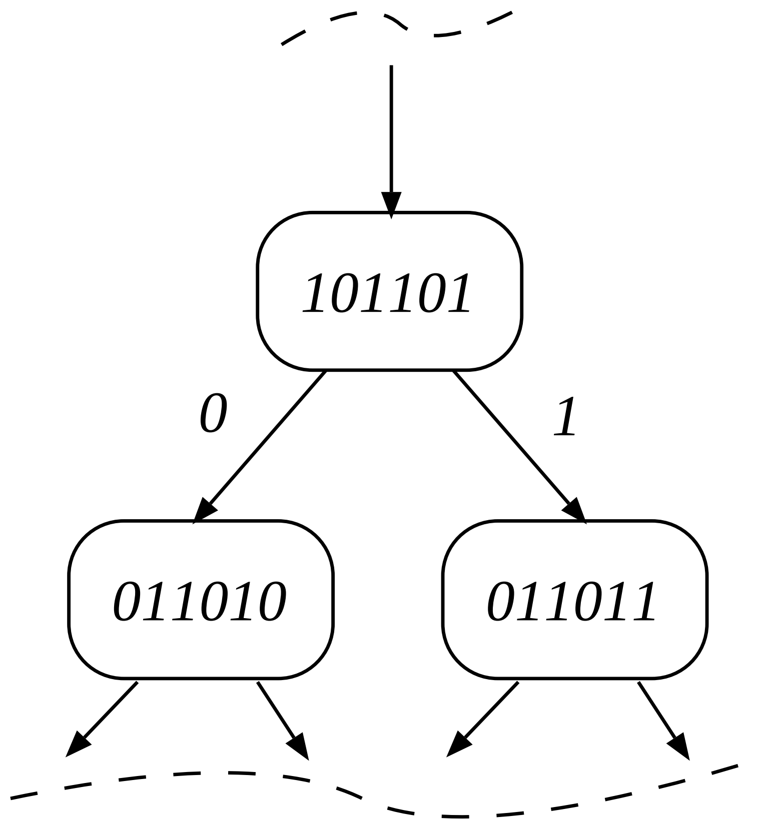

Figure 7.18: FRAGMENT OF STATE DIAGRAM OF PATTERN RECOGNIZER

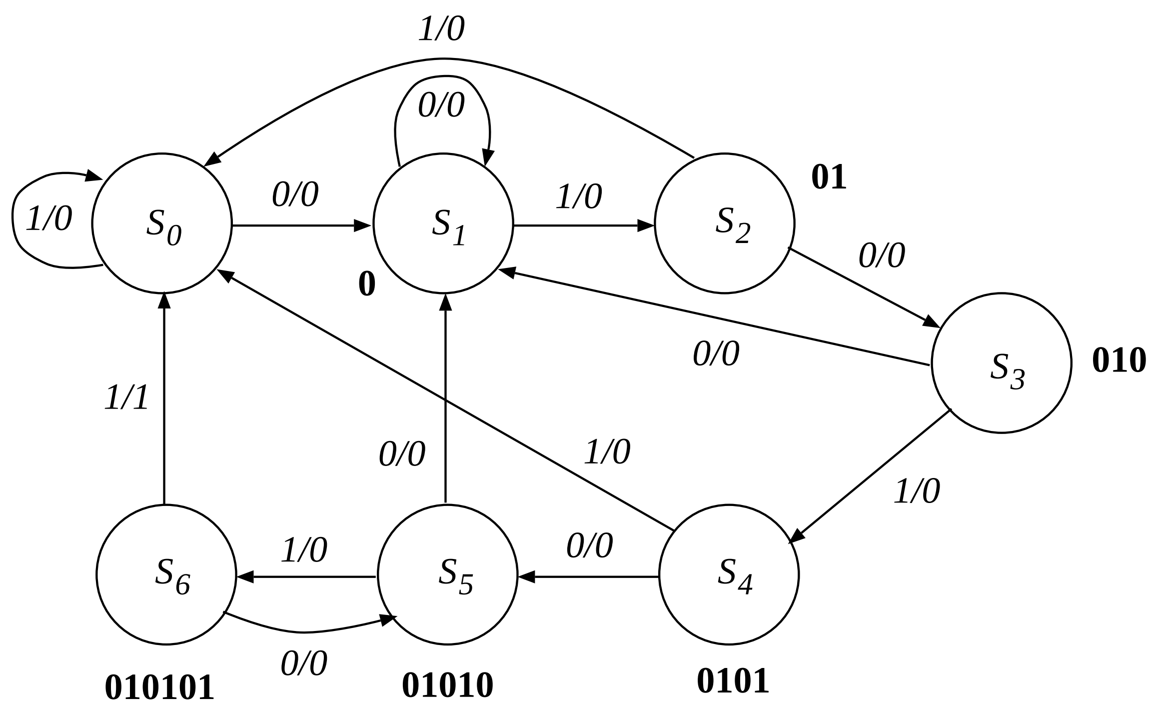

Figure 7.19: STATE DIAGRAM OF A PATTERN RECOGNIZER