Sequential Systems - ch. 7

course	
area	ucla
created	@January 26, 2023 2:33 PM
updated	@January 31, 2023 2:10 PM
✅ tasks	
📕 courses	
URL	
tags	cs

Definitions

▼

Big Ideas

▼ Serial Decimal Adder

▼ example

Resources

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/83551610-59dc-4bcf-a102-b4843422b2ef/ch7.pdf

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

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

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

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

State Description (Finite State Machines)

- e.g. state description of odd/even function

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	$a, 1$	$b, 0$	$b, 1$	$a, 1$	$b, 0$	$a, 0$	$b, 1$	$a, 1$

state description (PS = previous state)

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

▼ state diagram

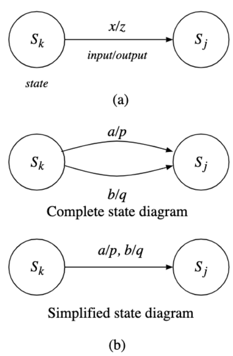

Figure 7.4: (a) State diagram representation. (b) Simplified state diagram notation.

▼ state transition table $\Longleftrightarrow$ state diagram

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

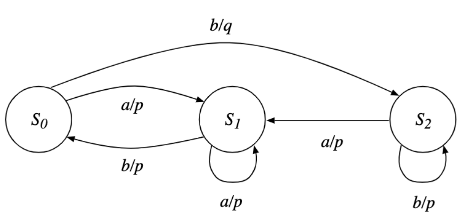

Figure 7.5: State diagram for Example 7.6.

▼ e.g. recursive subpatterns

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

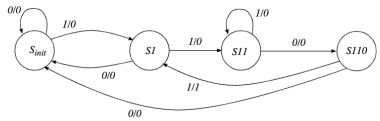

Figure 7.8: State diagram for Example 7.11

▼ Mealy and Moore Machines

▼ function descriptions (future state descriptions)

Mealy machine

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

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

Moore machine

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

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

▼ 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

▼ moore state diagram

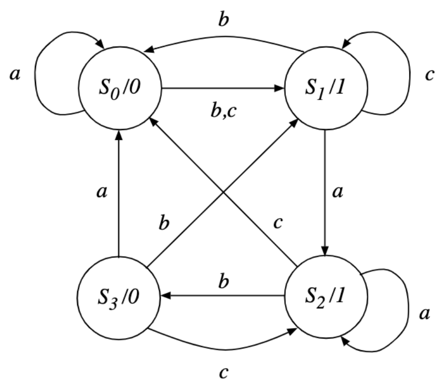

Figure 7.6: State diagram for Example 7.5

▼ Controllers

▼ Finite state machines that produce a control signal

▼ control signals determine actions of other systems → autonomous

▼ state diagram

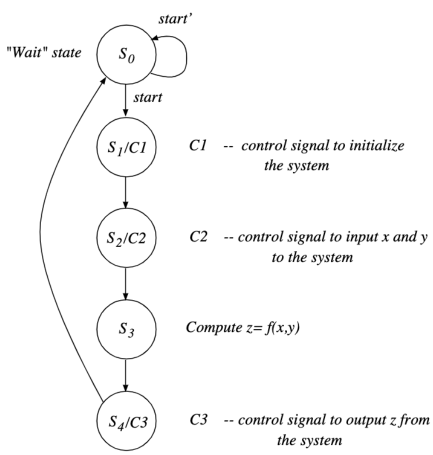

Figure 7.10: Autonomous controller: State diagram.

▼ e.g. candy machine

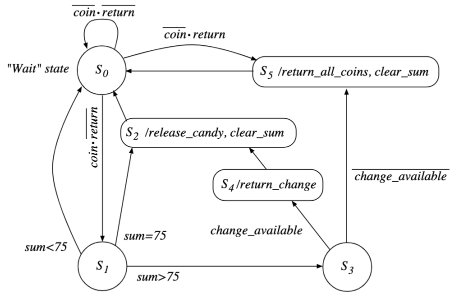

Figure 7.12: Controller for simple vending machine: State diagram.

▼ Equivalent Sequential Systems (reduced FSM)

▼ e.g. function → state transition table → reduced state diagram

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

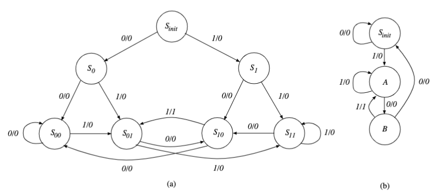

Figure 7.13: a) State diagram with redundant states; b) Reduced state diagram

▼ states that are not distinguishable by steps in between

- 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

▼ e.g. state transition table → new state

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	$B, 0$	$C, 0$	$F, 1$
	$NS,\ z$		

- 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

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

📌 **SUMMARY**