[CS M51A F14] Solution to Assignment 4

Due: 12/05/14

TA: Hyunjin Kim (roykim78@cs.ucla.edu)

Homework Problems (100 points total)

Problem 1 (5 points)

A system is described by the following table. Minimize the number of states. Show the corresponding minimal table.

	Input	
PS	$x = 0$	$x = 1$
A	F,0	D,1
B	F,0	C,1
C	B,0	F,0
D	G,0	A,0
E	I,0	H,1
F	A,0	C,1
G	A,0	D,1
H	A,1	C,2
I	E,0	H,1

Solution

$P_1 = \{A, B, E, F, G, I\}, \{C, D\}, \{H\}$

	group 1						group 2		group 3
	A	B	E	F	G	I	C	D	H
0	1	1	1	1	1	1	1	1	1
1	2	2	3	2	2	3	1	1	2

Now partition $P_2 = \{A, B, F, G\}\{E, I\}\{C, D\}\{H\}$:

	group 1				group 2		group 3		group 4
	A	B	F	G	E	I	C	D	H
0	1	1	1	1	2	2	1	1	1
1	3	3	3	3	4	4	1	1	3

Partition P_3 is the same as P_2 and we stop here. The minimal table is therefore:

	Input	
PS	$x = 0$	$x = 1$
a	a,0	c,1
b	b,0	d,1
c	a,0	a,0
d	a,1	c,2

Problem 2 (20 points)

Design a binary string detector which takes as input a string of binary values, and outputs a 1 when it detects a string of 1011 or 1100.

1. The input is $x(t)$, and the output is $z(t)$. Write a state description of the string detector by specifying the input and output sets, and write the output function.

   Solution

$$
\begin{array}{rl}
\text{Input:} & x(t) \in \{0,1\} \\
\text{Output:} & z(t) \in \{0,1\} \\
\text{Function:} & z(t) = \begin{cases} 0 & \text{if } x(t-3,t) = 1011 \text{ or } x(t-3,t) = 1100 \\ 1 & \text{otherwise} \end{cases}
\end{array}
$$

2. Fill in the state transition table for a Mealy machine of the string detector. Do not optimize the number of states yet.

   Solution

PS	$x=0$	$x=1$
S_{init}	$S_0, 0$	$S_1, 0$
S_0	$S_{00}, 0$	$S_{01}, 0$
S_1	$S_{10}, 0$	$S_{11}, 0$
S_{00}	$S_{000}, 0$	$S_{001}, 0$
S_{01}	$S_{010}, 0$	$S_{011}, 0$
S_{10}	$S_{100}, 0$	$S_{101}, 0$
S_{11}	$S_{110}, 0$	$S_{111}, 0$
S_{000}	$S_{000}, 0$	$S_{001}, 0$
S_{001}	$S_{010}, 0$	$S_{011}, 0$
S_{010}	$S_{100}, 0$	$S_{101}, 0$
S_{011}	$S_{110}, 0$	$S_{111}, 0$
S_{100}	$S_{000}, 0$	$S_{001}, 0$
S_{101}	$S_{010}, 0$	$S_{011}, 1$
S_{110}	$S_{100}, 1$	$S_{101}, 0$
S_{111}	$S_{110}, 0$	$S_{111}, 0$
	NS, z	

3. Minimize the number of states of the transition table, and show the final minimized table.

   Solution Looking at the output of the previous table, we first get P_1 as shown:

   $P_1 = \{S_{init}, S_0, S_1, S_{00}, S_{01}, S_{10}, S_{11}, S_{000}, S_{001}, S_{010}, S_{011}, S_{100}, S_{111}\}$ (outputs are both 0), $\{S_{101}\}$ (output is 0 for $x=0$ and 1 for $x=1$), and $\{S_{110}\}$ (output is 1 for $x=0$ and 0 for $x=1$)

	group 1													group 2	group 3
	S_{init}	S_0	S_1	S_{00}	S_{01}	S_{10}	S_{11}	S_{000}	S_{001}	S_{010}	S_{011}	S_{100}	S_{111}	S_{101}	S_{110}
0	1	1	1	1	1	1	3	1	1	1	3	1	3	1	1
1	1	1	1	1	1	2	1	1	1	2	1	1	1	1	2

   Now partition P_2 is:

	group 1								group 2		group 3			group 4	group 5
	S_{init}	S_0	S_1	S_{00}	S_{01}	S_{000}	S_{001}	S_{100}	S_{10}	S_{010}	S_{11}	S_{011}	S_{111}	S_{101}	S_{110}
0	1	1	2	1	2	1	2	1	1	1	5	5	5	2	1
1	1	1	3	1	3	1	3	1	4	4	3	3	3	3	4

And partition P_3 is:

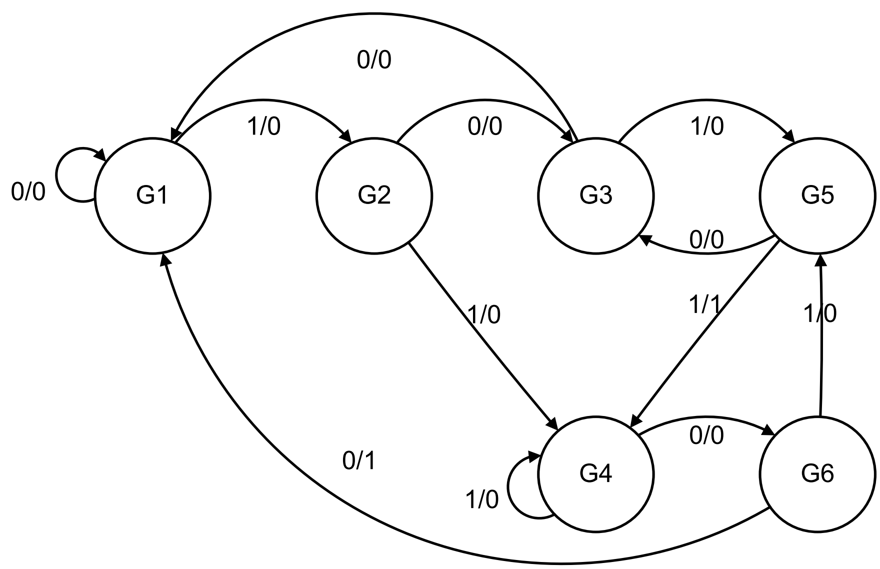
	group 1					group 2			group 3		group 4			group 5	group 6
	S_{init}	S_0	S_{00}	S_{000}	S_{100}	S_1	S_{01}	S_{001}	S_{10}	S_{010}	S_{11}	S_{011}	S_{111}	S_{101}	S_{110}
0	1	1	1	1	1	3	3	3	1	1	6	6	6	3	1
1	2	2	2	2	2	4	4	4	5	5	4	4	4	4	5

We can see that P_4 will be the same as P_3 and we stop here. By naming the states $G1$ to $G6$, we can write the following table.

PS	$x = 0$	$x = 1$
$G1$	$G1, 0$	$G2, 0$
$G2$	$G3, 0$	$G4, 0$
$G3$	$G1, 0$	$G5, 0$
$G4$	$G6, 0$	$G4, 0$
$G5$	$G3, 0$	$G4, 1$
$G6$	$G1, 1$	$G5, 0$
	NS, z	

4. Draw the minimized table as a state diagram.

   Solution The state diagram is as shown.

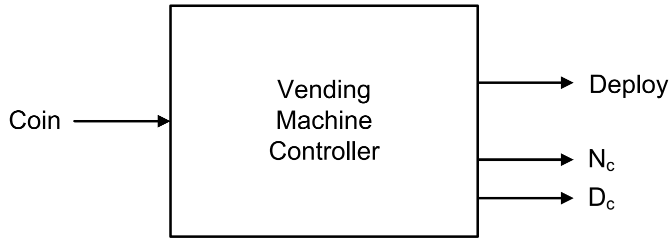


Problem 3 (20 points)

Our goal is to design a vending machine which sells stamps. The price of a stamp is 35 cents (for the sake of the problem). The machine accepts only nickels (5 cents), dimes (10 cents), and quarters (25 cents).

When the total value of coins is equal to or larger than the price of the stamp, the machine deploys the stamp, and returns any change as necessary. The change is given in nickels or dimes, and is returned in a way such that the total number of coins returned is the smallest possible. For example, if the amount of change is 30 cents, the machine returns 3 dimes, and not a mixture of dimes and nickels which will result in a higher coin count.

The machine's control module looks like the diagram below. It has an input *Coin* which denotes the type of coin deposited, and has three outputs: *Deploy*, which is 1 when the machine needs to deploy a stamp and 0 otherwise, N_c, number of nickels to return as change, and D_c, number of dimes to return as change.



1. What is the minimum number of states necessary for the control module? What would each state represent? (*Hint: To find the minimum number of states, first write any state machine which has the functionality that you want, and try to reduce the number of states afterwards.*)

   Solution One way of representing states would be to have a state for each possible total value of coins that have been inserted into the vending machine up to that time point. As the coin with the minimum value is a nickel, we can go up in multiples of 5. Any input which makes the current total of coins go over the price of the stamp makes the machine output the stamp and appropriate change, and sends the machine to the initial state.

   For this we will need the following states: $S_{init}, S_5, S_{10}, S_{15}, S_{20}, S_{25}$ and S_{30}.

   To make sure that this is the minimum number of necessary states, we apply the state minimization method. First, we need to write the state transition table as shown. The value of the input *Coin* is equal to N for nickel, D for dime, and Q for quarter. The three output digits are the values of $Deploy, N_c$ and D_c, in the same order.

	$Coin = N$	$Coin = D$	$Coin = Q$
S_{init}	$S_5, 000$	$S_{10}, 000$	$S_{25}, 000$
S_5	$S_{10}, 000$	$S_{15}, 000$	$S_{30}, 000$
S_{10}	$S_{15}, 000$	$S_{20}, 000$	$S_{init}, 100$
S_{15}	$S_{20}, 000$	$S_{25}, 000$	$S_{init}, 110$
S_{20}	$S_{25}, 000$	$S_{30}, 000$	$S_{init}, 101$
S_{25}	$S_{30}, 000$	$S_{init}, 100$	$S_{init}, 111$
S_{30}	$S_{init}, 100$	$S_{init}, 110$	$S_{init}, 102$

   With this table, we shall try to minimize the states.

   $P_1 = \{S_{init}, S_5\}, \{S_{10}\}, \{S_{15}\}, \{S_{20}\}, \{S_{25}\}, \{S_{30}\}$

	group 1		group 2	group 3	group 4	group 5	group 6
	S_{init}	S_5	S_{10}	S_{15}	S_{20}	S_{25}	S_{30}
N	1	2	3	4	5	6	1
D	2	3	4	5	6	1	1
Q	5	6	1	1	1	1	1

   Now we see that $P_2 = \{S_{init}\}, \{S_5\}, \{S_{10}\}, \{S_{15}\}, \{S_{20}\}, \{S_{25}\}, \{S_{30}\}$. Since all states are partitioned, it is not possible to reduce any states and we know that our state set is the minimum set.

   The minimum number of states for the vending machine is 7. Each state represents the total value of coins that have been inserted into the vending machine up to that time point.

2. Show the state transition table. The output should be written as a three-digit number, where each digit corresponds to the value of *Deploy*, N_c and D_c, in that order.

   Solution The minimum state transition table is given above, in the solution to part 1.

Problem 4 (15 points)

We are given a sequential system as shown below.

Inputs:	$x \in \{a, b, c\}$
Outputs:	$z \in \{0, 1\}$
Function:	$z = 1$ if $x(t-3, t) = abca$ and the number of a's in $x(0, t)$ is even

Obtain a "loose" state description of the system by creating a functionally valid state machine without worrying about it having the minimum number of states, and afterwards minimize the number of states.

Solution The loose description has a total of 8 states. We will set them up as shown below:

$$\begin{aligned}
s(t) &= A \text{ if } x(t-1) = a \text{ and number of } a\text{'s is even} \\
s(t) &= B \text{ if } x(t-1) = a \text{ and number of } a\text{'s is odd} \\
s(t) &= C \text{ if } x(t-2, t-1) = ab \text{ and number of } a\text{'s is even} \\
s(t) &= D \text{ if } x(t-2, t-1) = ab \text{ and number of } a\text{'s is odd} \\
s(t) &= E \text{ if } x(t-3, t-1) = abc \text{ and number of } a\text{'s is even} \\
s(t) &= F \text{ if } x(t-3, t-1) = abc \text{ and number of } a\text{'s is odd} \\
s(t) &= G \text{ if } x(t-3, t-1) = \mathit{other} \text{ and number of } a\text{'s is even} \\
s(t) &= H \text{ if } x(t-3, t-1) = \mathit{other} \text{ and number of } a\text{'s is odd}
\end{aligned}$$

Using these states, the corresponding state transition table is

	Input		
PS	$x = a$	$x = b$	$x = c$
A	$B, 0$	$C, 0$	$G, 0$
B	$A, 0$	$D, 0$	$H, 0$
C	$B, 0$	$G, 0$	$E, 0$
D	$A, 0$	$H, 0$	$F, 0$
E	$B, 0$	$G, 0$	$G, 0$
F	$A, 1$	$H, 0$	$H, 0$
G	$B, 0$	$G, 0$	$G, 0$
H	$A, 0$	$H, 0$	$H, 0$

Now we attempt to reduce the number of states from here. From the outputs we get

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

For P_2, we get

	group 1							group 2
	A	B	C	D	E	G	H	F
a	1	1	1	1	1	1	1	
b	1	1	1	1	1	1	1	
c	1	1	1	2	1	1	1	

For P_3, we get

	group 1						group 2	group 3
	A	B	C	E	G	H	D	F
a	1	1	1	1	1	1		
b	1	2	1	1	1	1		
c	1	1	1	1	1	1		

For P_4, we get

	group 1					group 2	group 3	group 4
	A	C	E	G	H	B	D	F
a	2	2	2	2	1			
b	1	1	1	1	1			
c	1	1	1	1	1			

For P_5, we get

	group 1				group 2	group 3	group 4	group 5
	A	C	E	G	H	B	D	F
a	3	3	3	3				
b	1	1	1	1				
c	1	1	1	1				

From this, we can write

$$P_5 = P_4 = (A, C, E, G)(H)(B)(D)(F)$$

The final table is

	Input		
PS	$x = a$	$x = b$	$x = c$
$G1$	$G3, 0$	$G1, 0$	$G1, 0$
$G2$	$G1, 0$	$G2, 0$	$G2, 0$
$G3$	$G1, 0$	$G4, 0$	$G2, 0$
$G4$	$G1, 0$	$G2, 0$	$G5, 0$
$G5$	$G1, 1$	$G1, 0$	$G1, 0$
	NS,Output		

LogiSim Design Problem (40 points)

1 A Pattern Detector

1.1 Provide implementation details of your circuit by filling in the following table with the minimum number (4) of states.

	Input	
PS	$x = 0$	$x = 1$
$S0$	$S0, 0$	$S1, 0$
$S1$	$S0, 0$	$S2, 0$
$S2$	$S3, 0$	$S2, 0$
$S3$	$S0, 0$	$S1, 1$
	NS,Output	

1.2 Write the expressions of the next state and the output from the transition table

$NS_1 = S_1' S_0 x + S_1 S_0'$
$NS_0 = S_1' S_0' x + S_1 S_0' x' + S_1 S_0 x$
$z = S_1 S_0 x$
(+5 points) $NS_0 = S_1' S_0' x + S_1 S_0' x' + z$

1.3 Logisim Design