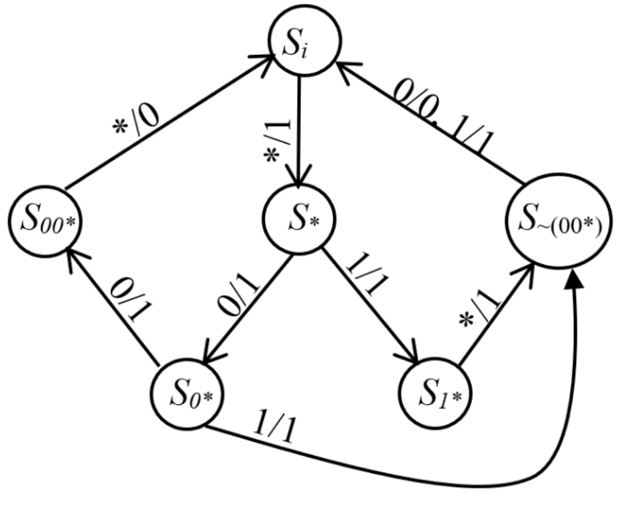
1. Optimal assignment of binary codes
    a. Was implementing binary code checker last time
    b. Drew naïve implementation of FSM, then minimized it



State transition diagram

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x =1
i	*	*	1	1
*****	0*	1*	1	1
0*	00*	~(00*)	1	1
1*	~(00*)	~(00*)	1	1
00*	*i*	*i*	0	0
~(00*)	*i*	*i*	0	1

State table

    c. Can assign binary codes for states randomly

    d. Rule of thumb for state binary code assignments

    e. Assign using the rules above

		AB			
Binary Code		**00**	**01**	**11**	**10**
C	**0**				
	1				

    f. Note that there may potentially be more than one valid code assignment that minimizes distance

2. Debugging an FSM
    a. Generally, much more efficient to put in effort to get it right to begin with

    b. One good way of seeing if your FSM you drew was right

    c. Examples
        i. With the BCD checker

        ii. With the vending machine below

3. More complicated FSMs
    a. Design a vending machine

    b. x_1, x_2

    c. Will use a Mealy model

    d. First, create state transition diagram

e. Next, minimize the number of states using the Partition Minimization Procedure

i. P_1

ii. P_2

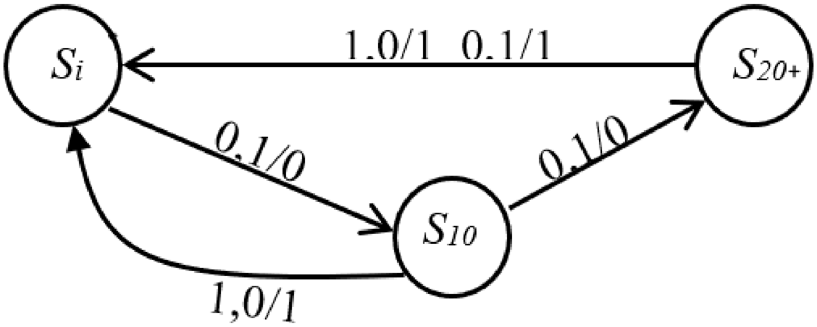
iii. Draw new state transition diagram

f. Assign code words next

$Binary\ Code$

		A	
		0	**1**
B	**0**		
	1		

g. Next, create state transition table



Binary Code A

		0	**1**
B	**0**	*i*	20+
	1	10	

Present State	**Binary Code**	**Present State**		**Inputs**		**Next State**		**Output** z
		A	**B**	x_1	x_2	**A'**	**B'**	
i	00	0	0	0	0			
i	00	0	0	0	1			
i	00	0	0	1	0			
		0	0	1	1			
10	01	0	1	0	0			
10	01	0	1	0	1			
10	01	0	1	1	0			
		0	1	1	1			
20+	10	1	0	0	0			
20+	10	1	0	0	1			
20+	10	1	0	1	0			
		1	0	1	1			
		1	1	0	0			
		1	1	0	1			
		1	1	1	0			
		1	1	1	1			

h. Finally, create K-maps from table above

Present State	Binary Code	Present State		Inputs		Next State		Output
		A	B	x_1	x_2	A'	B'	z
i	00	0	0	0	0	0	0	0
i	00	0	0	0	1	0	1	0
i	00	0	0	1	0	1	0	0
		0	0	1	1	d	d	d
10	01	0	1	0	0	0	1	0
10	01	0	1	0	1	1	0	0
10	01	0	1	1	0	0	0	1
		0	1	1	1	d	d	d
20+	10	1	0	0	0	1	0	0
20+	10	1	0	0	1	0	0	1
20+	10	1	0	1	0	0	0	1
		1	0	1	1	d	d	d
		1	1	0	0	d	d	d
		1	1	0	1	d	d	d
		1	1	1	0	d	d	d
		1	1	1	1	d	d	d

A' AB

		00	01	11	10
x_1x_2	00				
	01				
	11				
	10				

B' AB

		00	01	11	10
x_1x_2	00				
	01				
	11				
	10				

z AB

		00	01	11	10
x_1x_2	00				
	01				
	11				
	10				