CS221: Digital Design

FSM: Optimization and State Encoding

A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- FSM State Optimization
  - Row Matching method
  - Partitioning method
  - Implication chart (Next Class, Lect 33/34)
- FSM State Encoding (Lect 34/35)

FSM State Minimization

- Minimizing number of state reduce
  - Requirement of bigger size state register
  - Possibly reduce the Combinational Circuit Complexity (CCC) for the FSM

References materials

- **Section 14.6 and 14.7 of Kumar Book**
  - A. Anand Kumar, *Fundamentals of Digital Circuits* 3rd Edition, PHI. 2014 **((This book have a lot of examples to understand the concepts))**
- **Section 8.1 of Katz Book**
  - Randy H. Katz, G Borriello, *Contemporary Logic Design*, 2nd Edition, PHI, India, 2009
  - This is one of the prescribed/listed text book for the course along with Mano Book

Some Definitions

- **State Equivalence**: S1 and S2 are equivalent if for every input sequence applied to machine goes to same NS and Output
  - If S1(t+1)=S2(t+1) and Z1=Z2 then S1=S2

- **Distinguishable States**: Two states S1 and S2 are Distinguishable ***if and only if*** there exist at least one finite input sequence which produce different outputs from S1 and S2

FSM Optimization Methods

- FSM State Optimization
  - Given FSM
  - Goal is to reduce number of states of FSM
- Method 1: Row Matching Method
  - Completely specified machine (n^2 edges)
  - Partially specified machine
- Method 2: Partitioning Method
- Method 3: Implication Chart Method

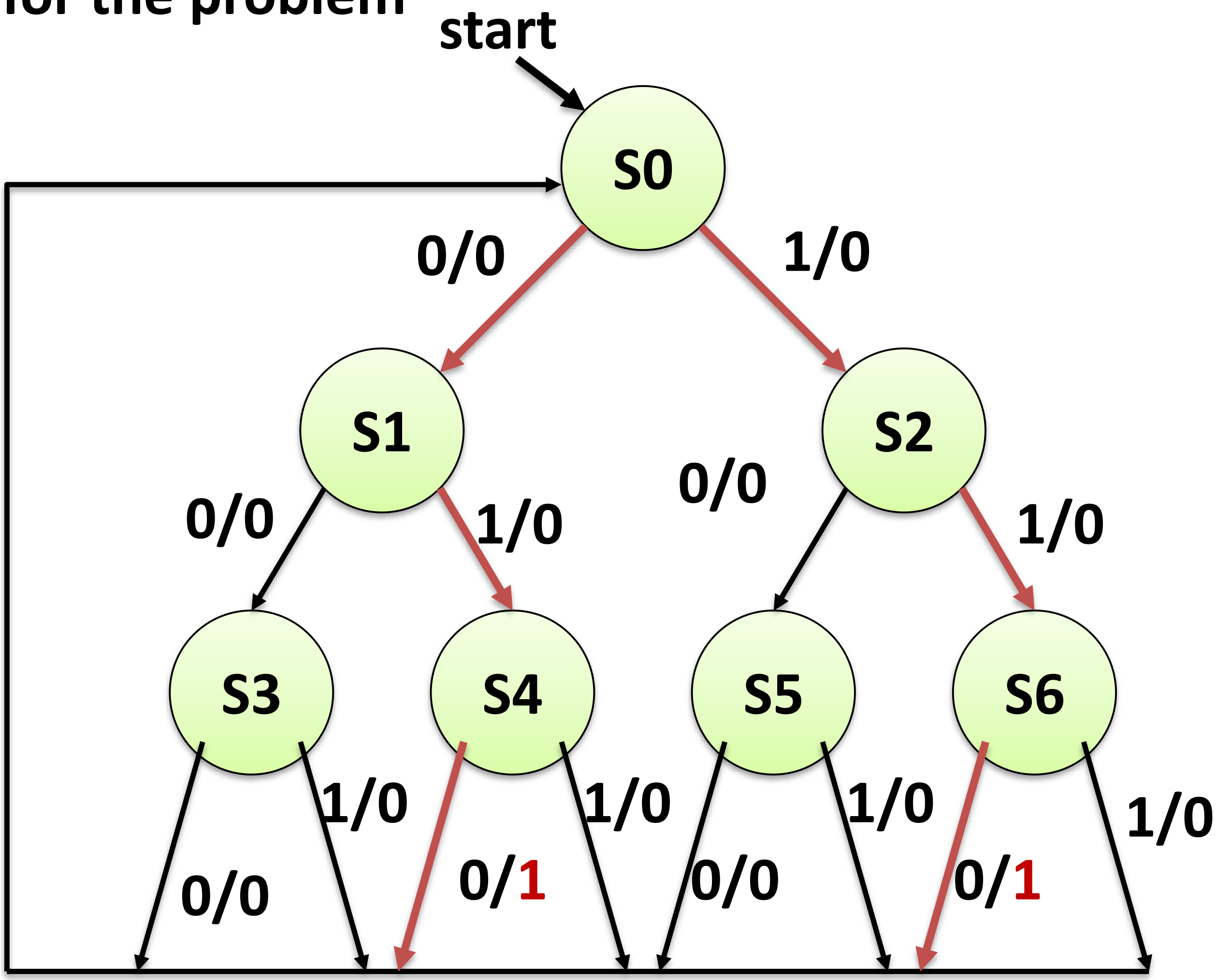
Row Matching Methods

State Minimization : Example

- **Sequence Detector for 010 or 110 :**
  - After each 3 bit input sequence
  - if it consist of 110 or 010, out put 1

State Minimization : Example

- Sequence Detector for 010 or 110:
  - After each 3 bit input sequence , if it consist of 110 or 010, out put 1
- Your designed FSM for the problem



Is it optimized?

State Minimization Example

- Sequence Detector for 010 or 110
- After is asserted after each 3 bit input sequence if it consist of 110 or 010

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S4 S5 S6)

State Minimization Example

- S4 and S6 are different as compared to other States based on output

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S5) (S4 S6)

State Minimization Example

- S4 and S6 have same NS and O/P, they are same

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S5) (S4 S6)

State Minimization Example

- S4 and S6 have same NS and O/P, they are same

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01	S4’	S0	S0	1	0
10	S5	S0	S0	0	0
11	S4’	S0	S0	1	0

(S0 S1 S2 S3 S5) (S4’)

State Minimization Example

- Reduced FSM after merging S4, S6 to S4’

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01/11	S4’	S0	S0	1	0
10	S5	S0	S0	0	0

(S0 S1 S2 S3 S5) (S4’)

State Minimization Example

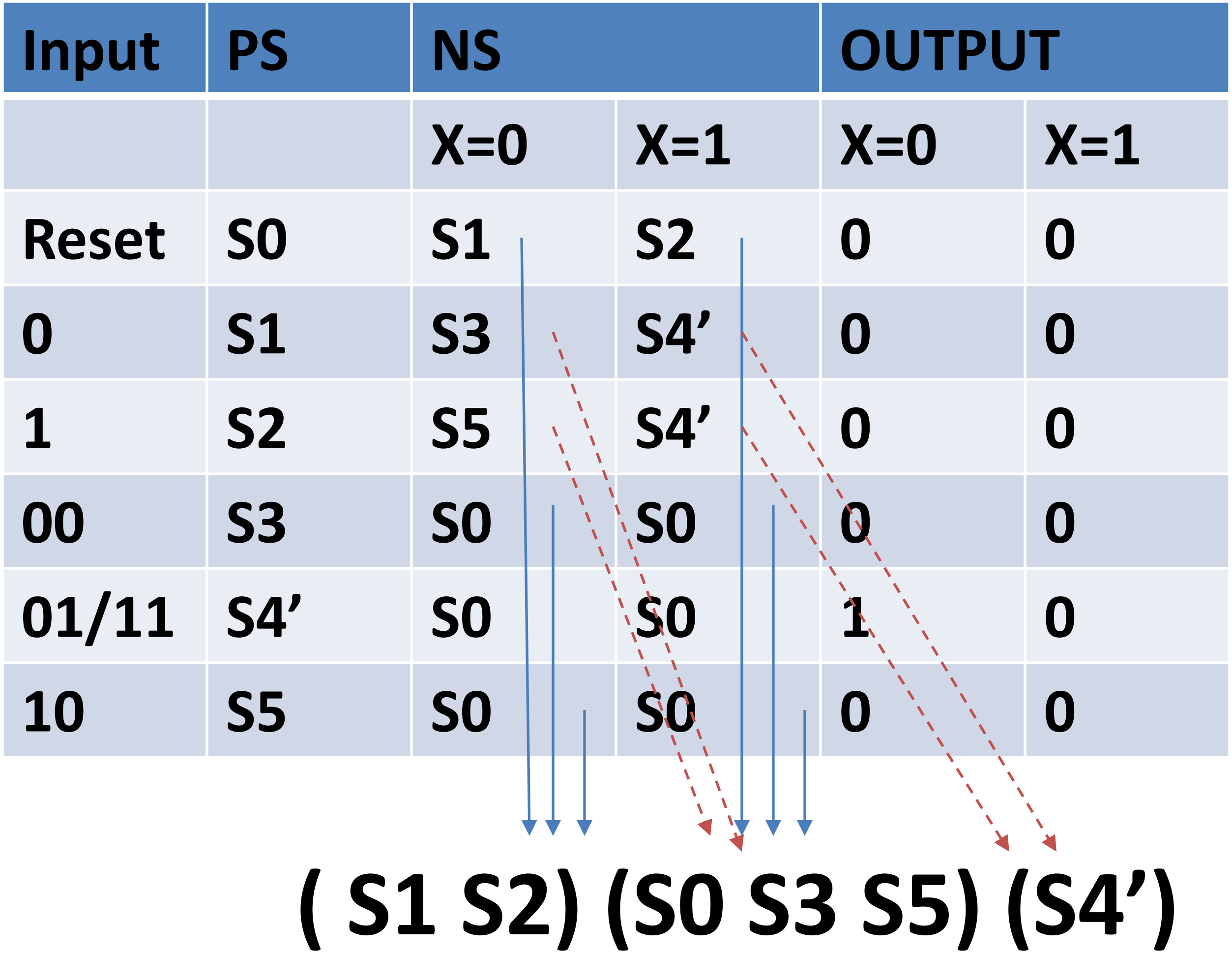
- NS of S1, S2 are different as compared to S0, S3 and S5

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01/11	S4’	S0	S0	1	0
10	S5	S0	S0	0	0

(S0 S1 S2 S3 S5) (S4’)

State Minimization Example

- NS of S1, S2 are different as compared to S0, S3 and S5



Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01/11	S4’	S0	S0	1	0
10	S5	S0	S0	0	0

(S1 S2) (S0 S3 S5) (S4’)

State Minimization Example

- NS of S1, S2 are different as compared to S0, S3 and S5

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01/11	S4’	S0	S0	1	0
10	S5	S0	S0	0	0

(S1 S2) (S0 S3 S5) (S4’)

State Minimization Example

- S3 and S5 have same NS and O/p, so they are same

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01/11	S4’	S0	S0	1	0
10	S5	S0	S0	0	0

(S1 S2) (S0 S3 S5) (S4’)

State Minimization Example

- Merging S3 and S5 to S3’

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3’	S4’	0	0
1	S2	S3’	S4’	0	0
00/10	S3’	S0	S0	0	0
01/11	S4’	S0	S0	1	0

(S1 S2) (S0 S3’) (S4’)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3’	S4’	0	0
1	S2	S3’	S4’	0	0
00/10	S3’	S0	S0	0	0
01/11	S4’	S0	S0	1	0

(S1 S2) (S0 S3’) (S4’)

State Minimization Example

- S1 and S2 have same NS and O/P, they are same

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3’	S4’	0	0
1	S2	S3’	S4’	0	0
00/10	S3’	S0	S0	0	0
01/11	S4’	S0	S0	1	0

(S1 S2) (S0 S3’) (S4’)

State Minimization Example

- Merging S1 and S2 to S1’

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1’	S1’	0	0
0/1	S1’	S3’	S4’	0	0
00/10	S3’	S0	S0	0	0
01/11	S4’	S0	S0	1	0

(S1’) (S0 S3’) (S4’)

State Minimization Example

- S0, S3’ are different state as their NS are different

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1’	S1’	0	0
0/1	S1’	S3’	S4’	0	0
00/10	S3’	S0	S0	0	0
01/11	S4’	S0	S0	1	0

(S1’) (S0 S3’) (S4’)

State Minimization Example

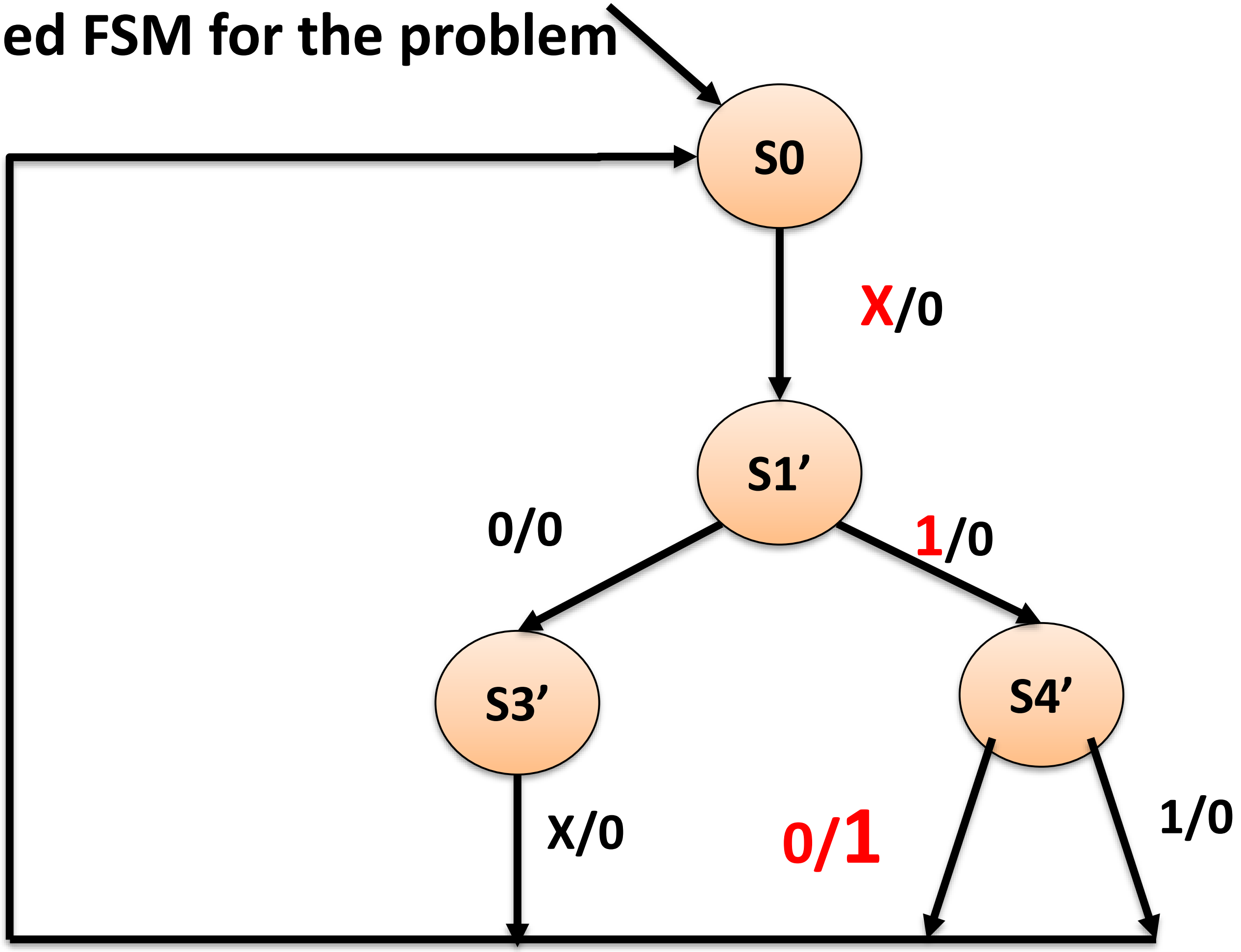
- No further matching =➔ Reduced one

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1’	S1’	0	0
0/1	S1’	S3’	S4’	0	0
00/10	S3’	S0	S0	0	0
01/11	S4’	S0	S0	1	0

(S1’) (S0) (S3’) (S4’)

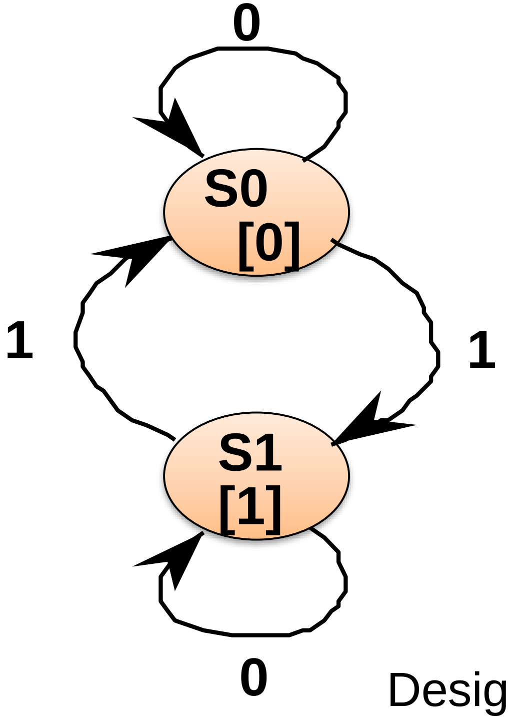
Minimized FSM

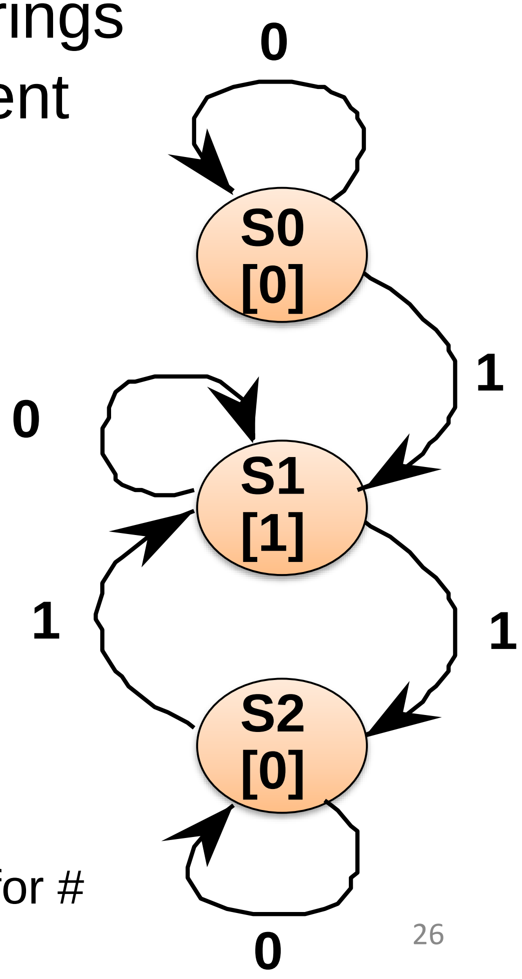
- Sequence Detector for 010 or 110:
  - After each 3 bit input sequence , if it consist of 110 or 010, out put 1
- Your designed FSM for the problem



Row Matching Method : Fallacies

- Odd Parity Checker:
  - Two alternative state diagrams
  - Identical output behavior on all input strings
  - FSMs are *equivalent*, but require different implementations





Design state diagram without concern for # of states,   Reduce later

Critique of Row Matching

- Straightforward to understand and easy to implement

- Problem: does not allows yield the most reduced state table!

	Next State		
Present State	X=0	X=1	Output
S_0	S_0	S_1	0
S_1	S_1	S_2	1
S_2	S_2	S_1	0

(S0S1S2)
Based on
output
(S0S2) (S1)

No way to combine states S0 and S2 based on Next State Criterion!

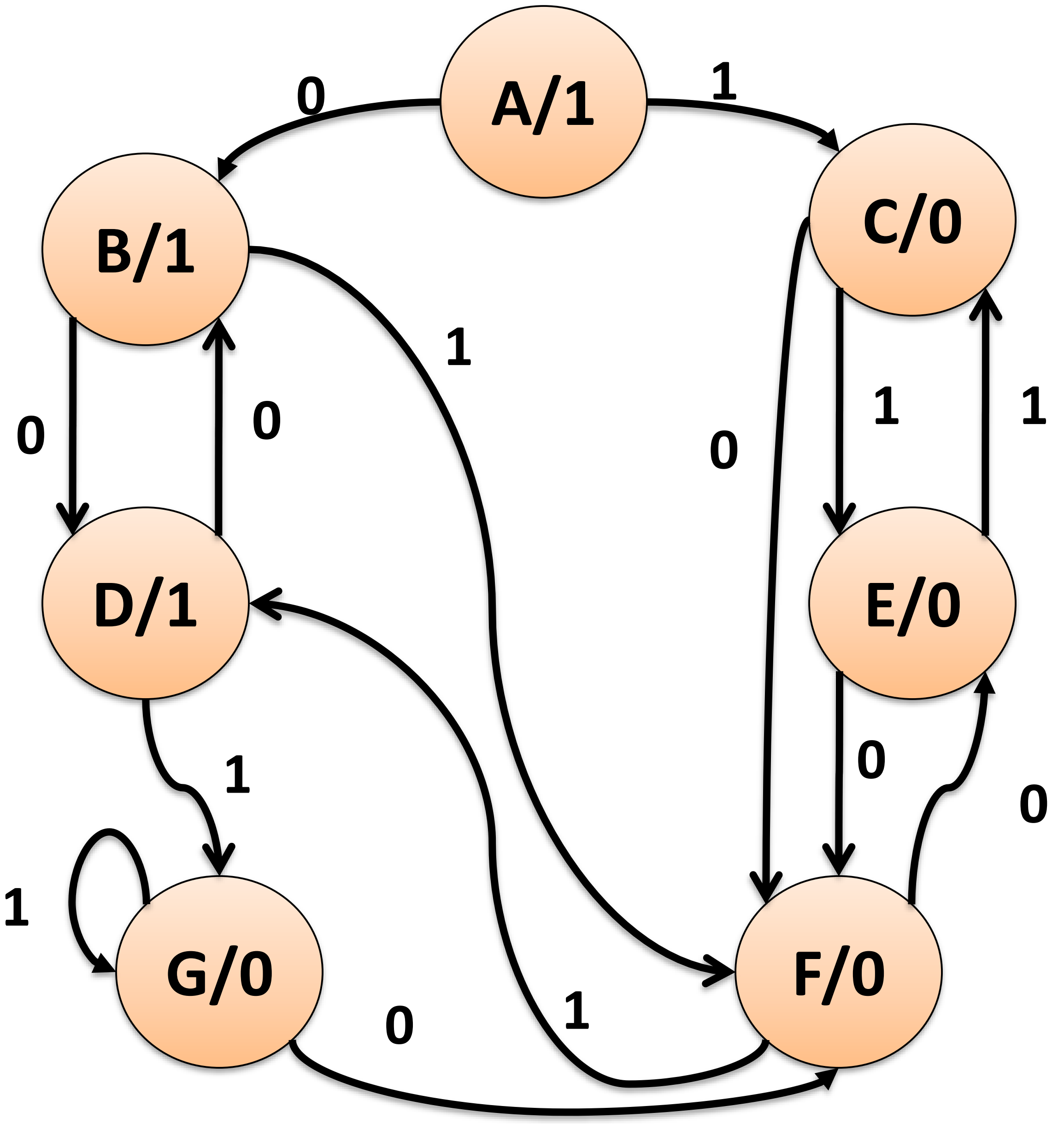
Partitioning Methods

State Minimization: Partitioning

- Form an initial partition (P_1)
  - that includes **all the** states.
- Form a 2nd partition (P_2)
  - By separating the states into two blocks based upon their output values.
- Form a third partition (P_3)
  - by separating the states into blocks corresponding to the next state values.
- Continue partitioning until
  - Two successive partitions are the same (i.e. $P_{N-1} = P_N$).
- All states in any one block are equivalent
  - Equivalent states can be combined into a single state.

State Minimization Partitioning: Example

State Diagram



State Minimization: Partitioning

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

State Minimization: Partitioning

Initial Partition:

P_1 = (ABCDEFG)

The initial partition contains all states in the state diagram / table.

State Minimization: Partitioning

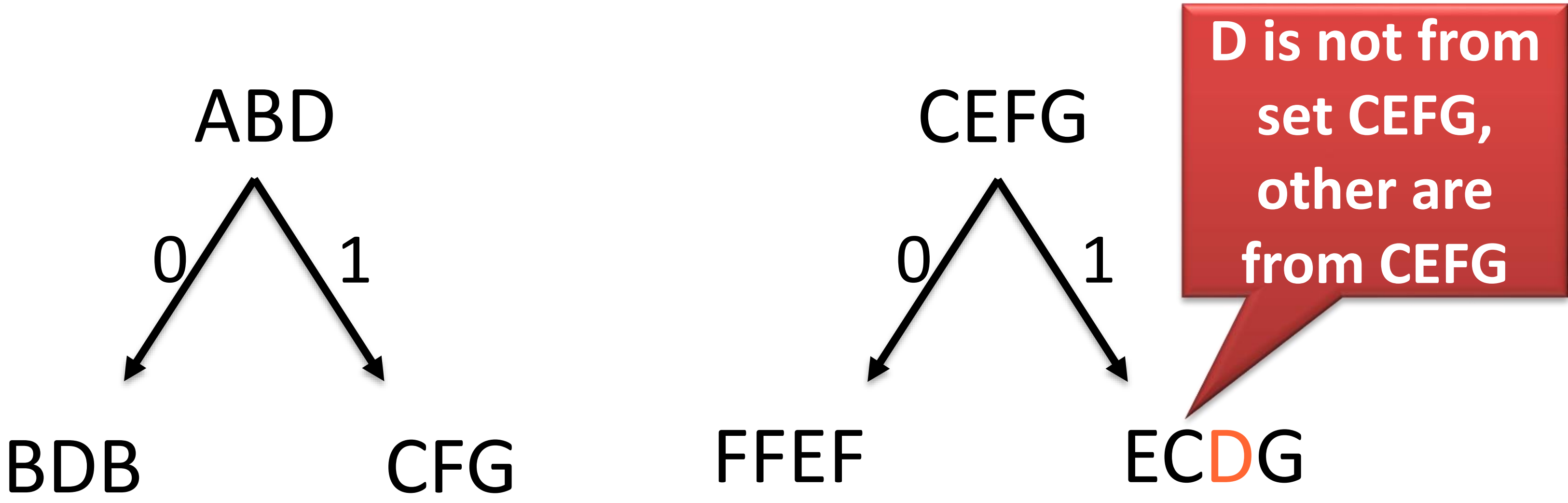
- Separate states based on output value.
  - $P_2 = (ABD)(CEFG)$

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

State Minimization: Partitioning

P_2 = (ABD) (CEFG)

- Separate states based on next state values.



D is not from set CEFG, other are from CEFG

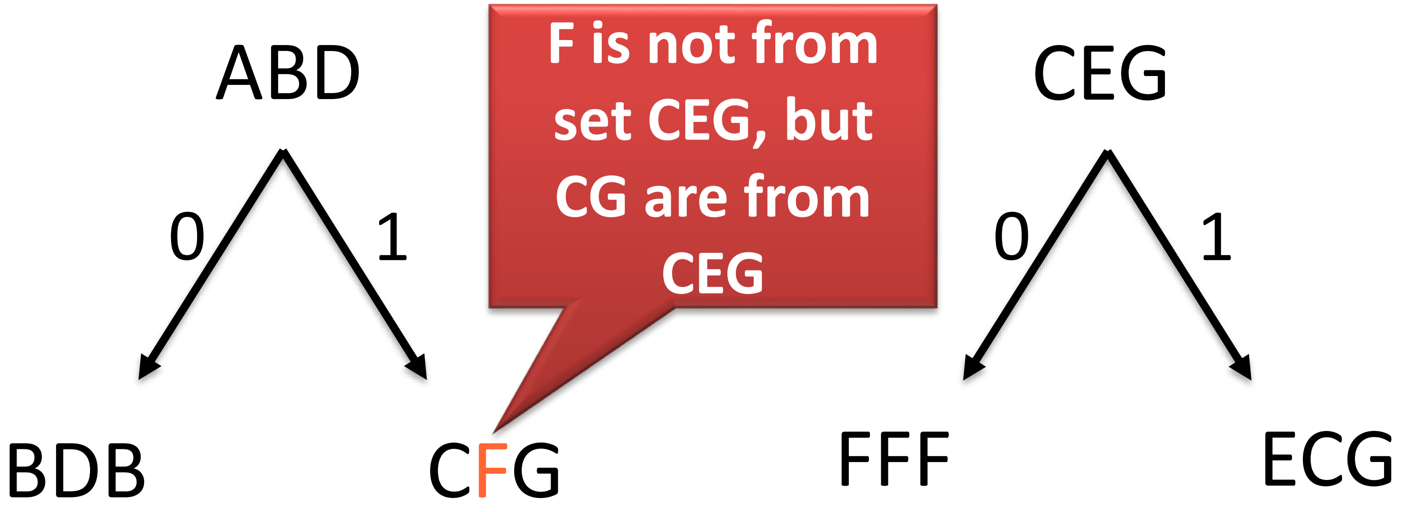
P_3 = (ABD) (CEG) (F)

unique state

State Minimization: Partitioning

$$P_3 = (ABD)\ \ (CEG)\ \ (F)$$

- Separate states based on next state values.



ABD

0

1

BDB

CFG

F is not from set CEG, but CG are from CEG

CEG

0

1

FFF

ECG

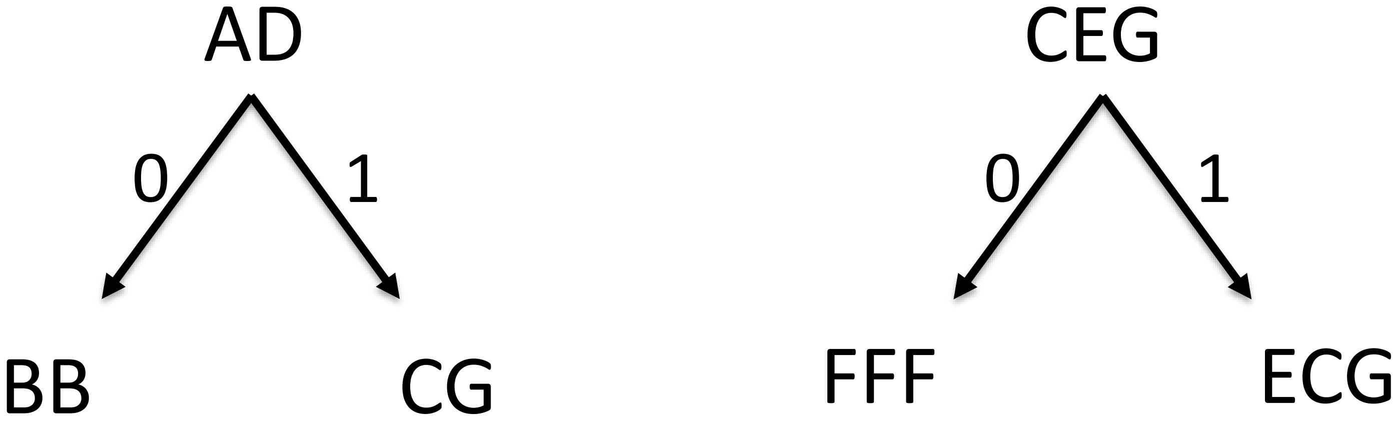
$$P_4 = (AD)\ (CEG)\ \ (F)\ (B)$$

unique states

State Minimization: Partitioning

P_4 = (AD) (CEG) (F) (B)

- Separate states based on next state values.



P_5 = (AD)(CEG)(F)(B)

↑

Same as previous partition (P_4)

State Minimization: Partitioning

- Since $P_4 = P_5$, state minimization is complete.
- The equivalent states are:
  - A = D
  - C = E = G
  - B
  - F

P_4 = (AD) (CEG) (F) (B)

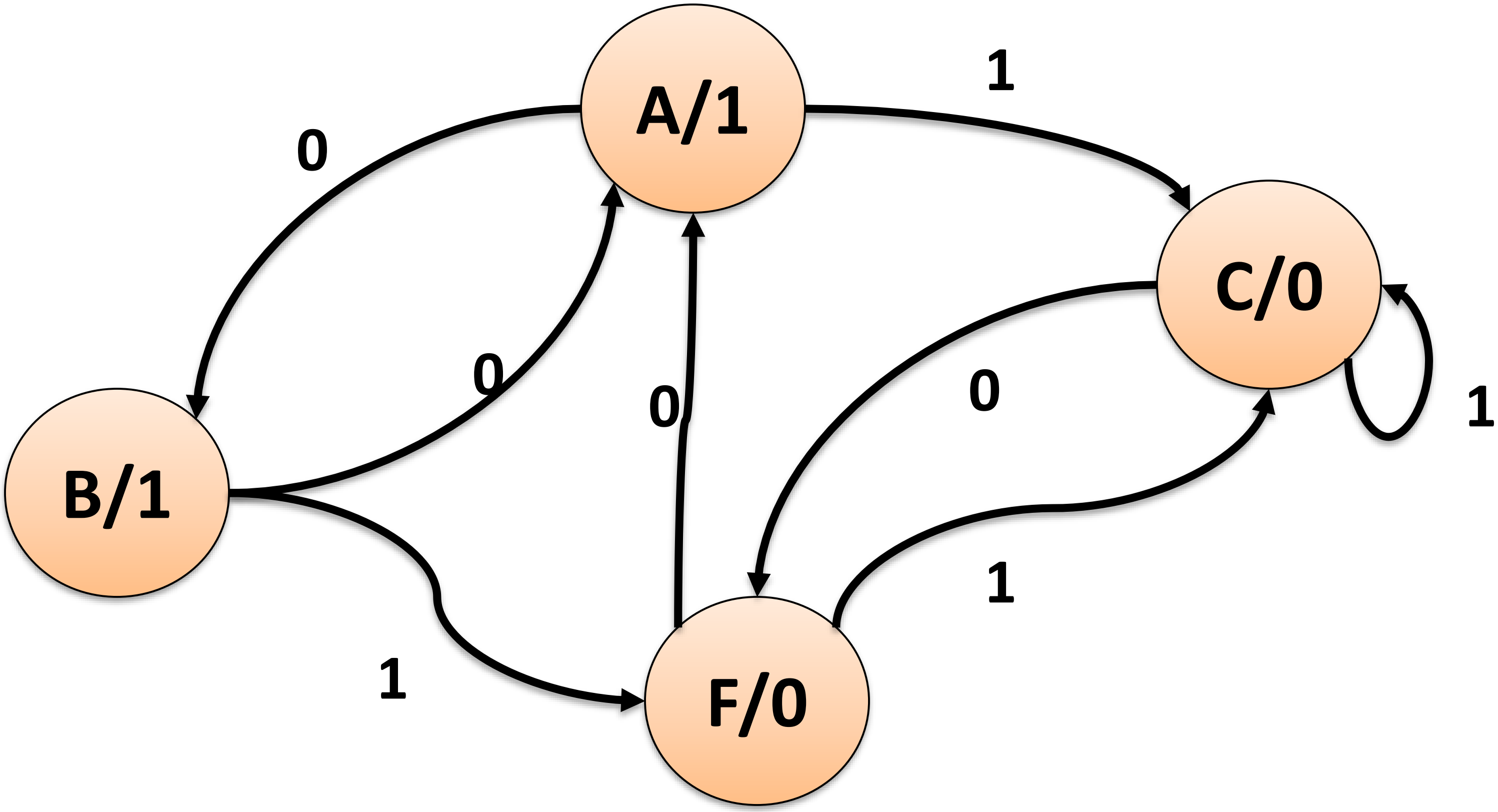
- Thus, the FSM can be realized with just 4 states.

FSM: State Minimization

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	0

Minimized State Table

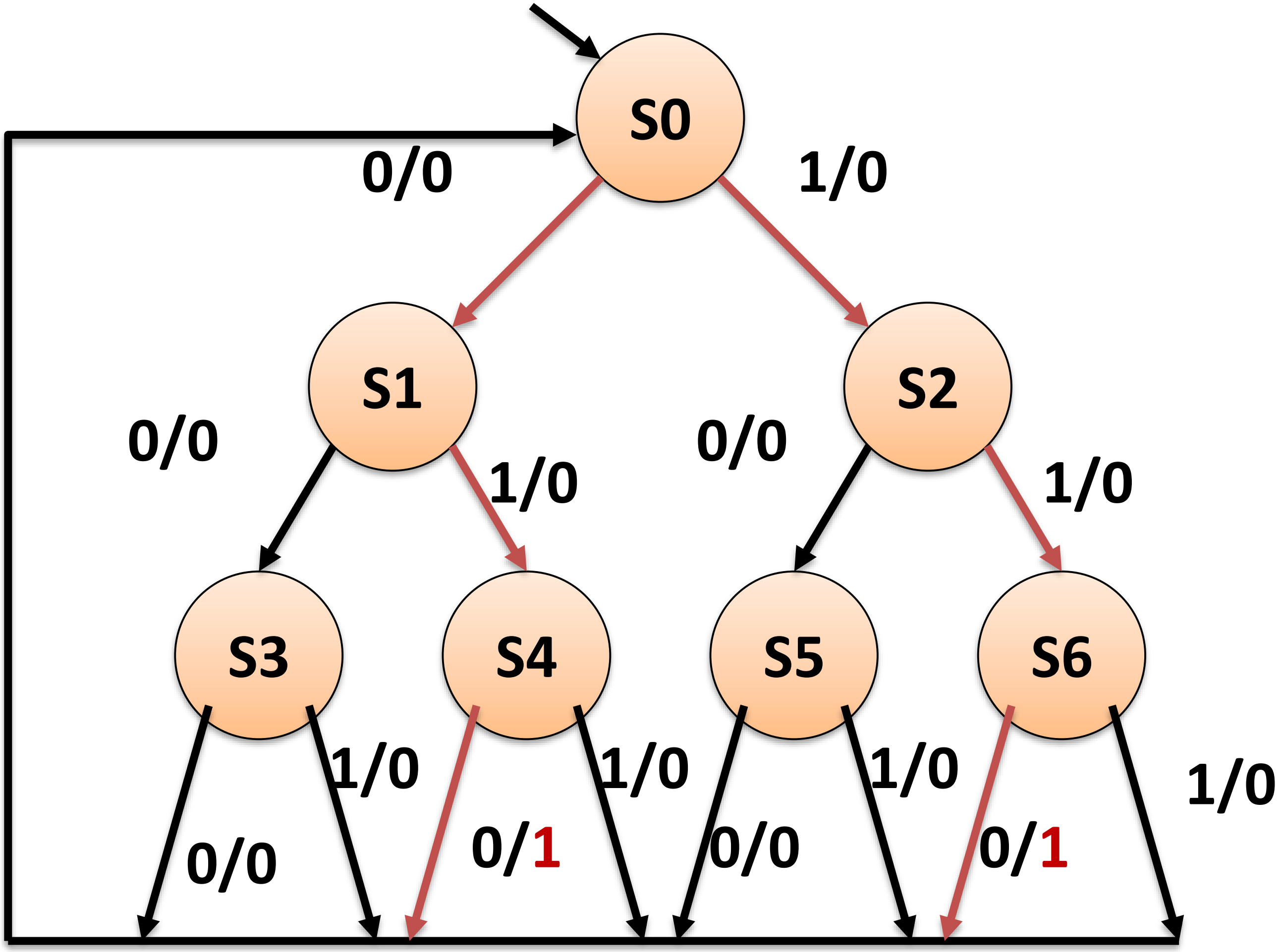
FSM: State Minimization



Minimized
State Diagram

State Minimization: Previous Example

- Sequence Detector for 010 or 110
- After is asserted after each 3 bit input sequence if it consist of 110 or 010



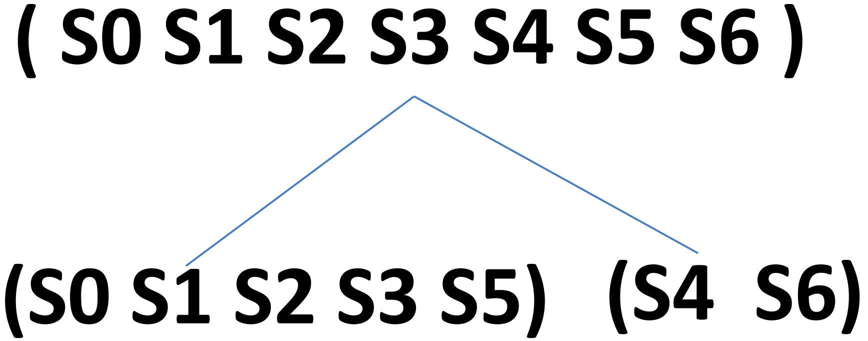
State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S4 S5 S6)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

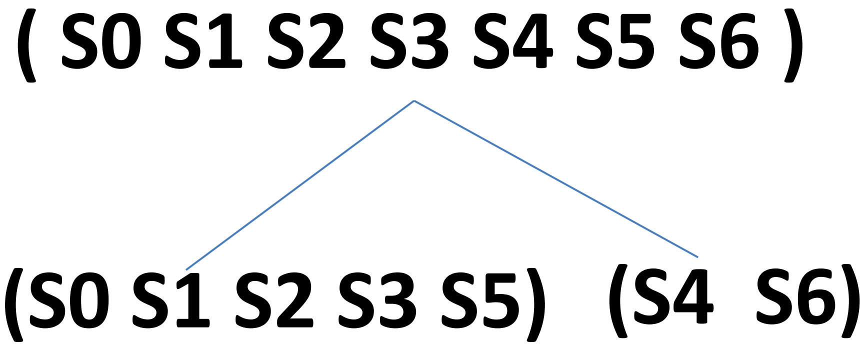


(S0 S1 S2 S3 S4 S5 S6)

(S0 S1 S2 S3 S5) (S4 S6)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0



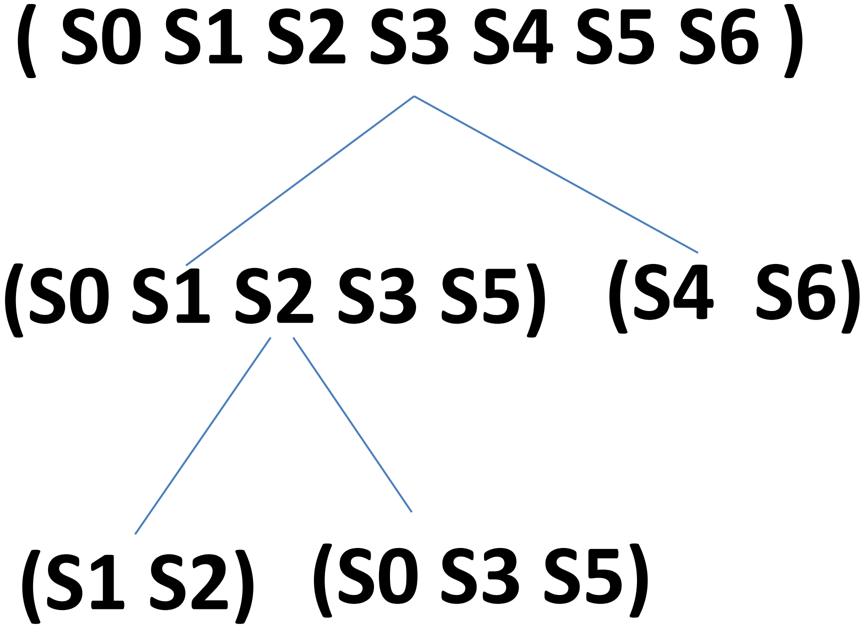
(S0 S1 S2 S3 S4 S5 S6)

(S0 S1 S2 S3 S5)

(S4 S6)

State Minimization Example

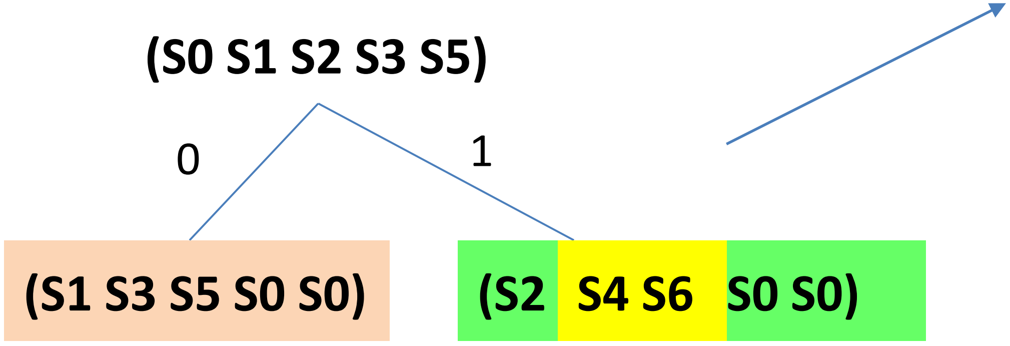
Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0



(S0 S1 S2 S3 S4 S5 S6)

(S0 S1 S2 S3 S5) (S4 S6)

(S1 S2) (S0 S3 S5)



(S0 S1 S2 S3 S5)

0

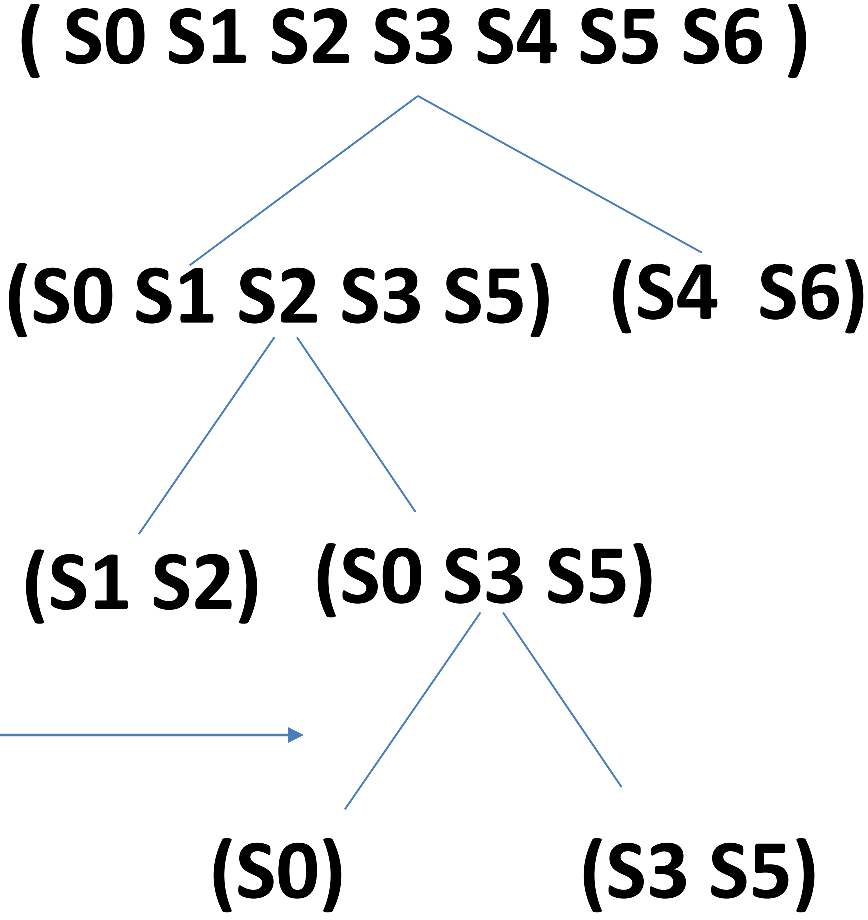
1

(S1 S3 S5 S0 S0)

(S2 S4 S6 S0 S0)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

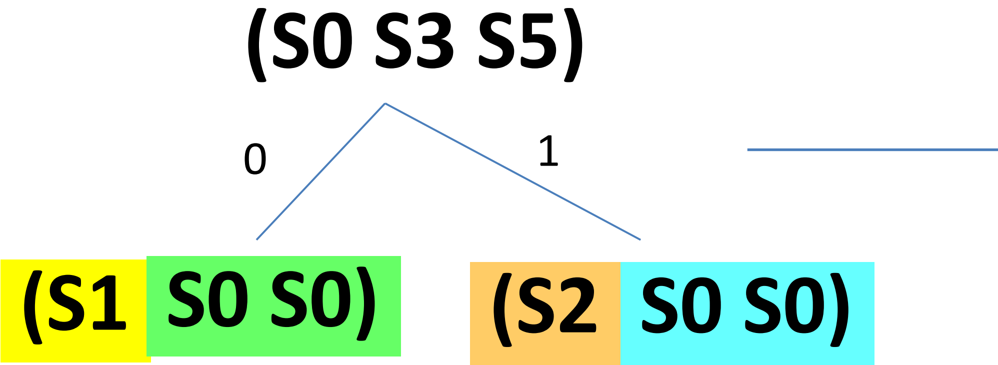


(S0 S1 S2 S3 S4 S5 S6)

(S0 S1 S2 S3 S5) (S4 S6)

(S1 S2) (S0 S3 S5)

(S0) (S3 S5)

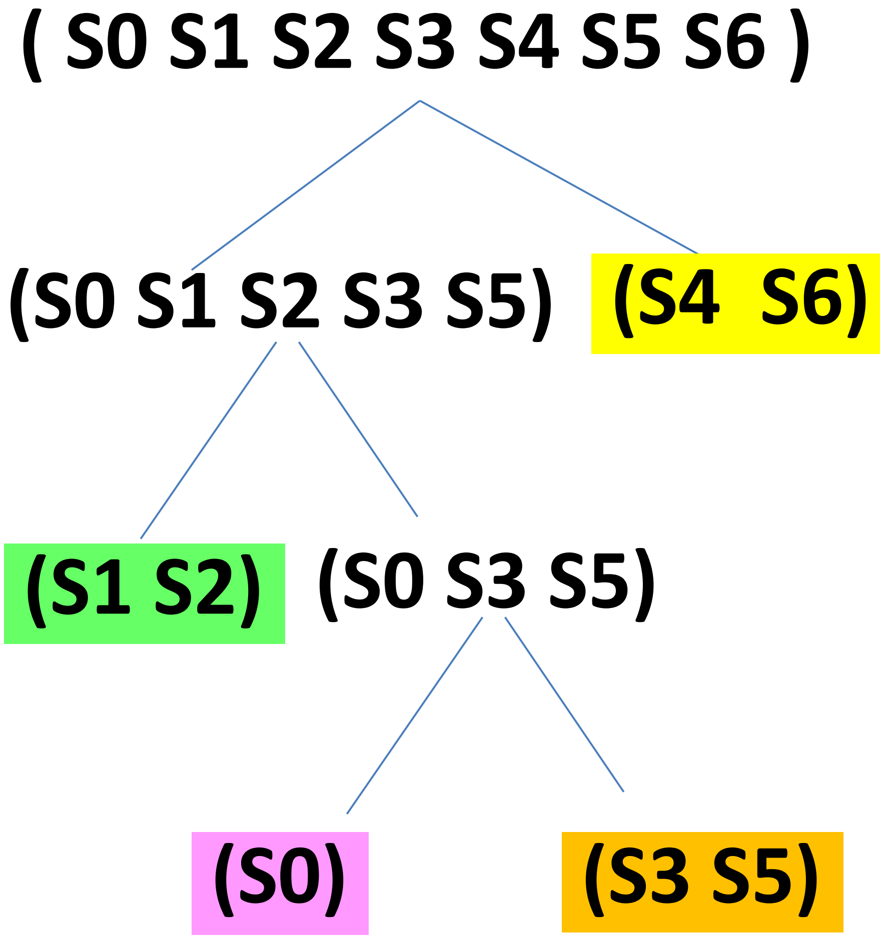


(S0 S3 S5)

(S1 S0 S0) (S2 S0 S0)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0



(S0 S1 S2 S3 S4 S5 S6)

(S0 S1 S2 S3 S5) (S4 S6)

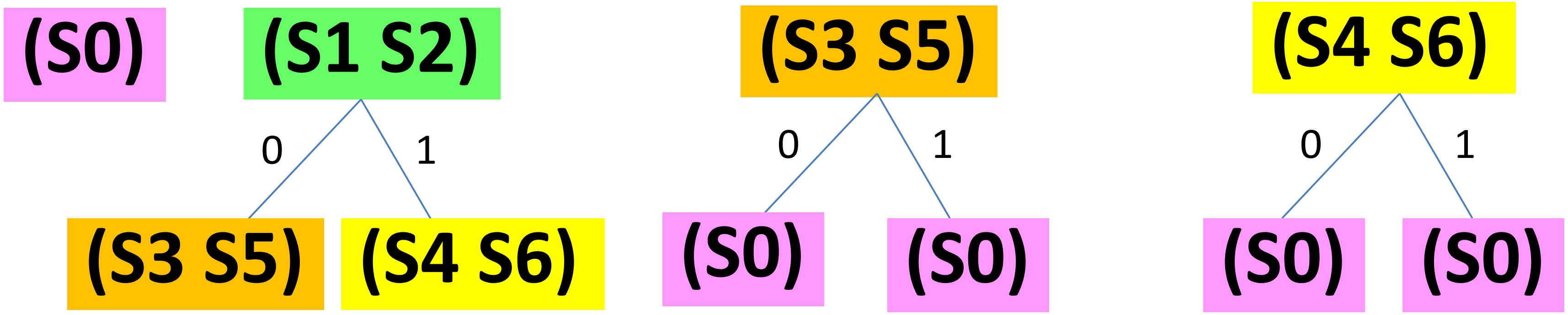
(S1 S2) (S0 S3 S5)

(S0) (S3 S5)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S4 S5 S6)



(S0)

(S1 S2)

0 1

(S3 S5) (S4 S6)

(S3 S5)

0 1

(S0) (S0)

(S4 S6)

0 1

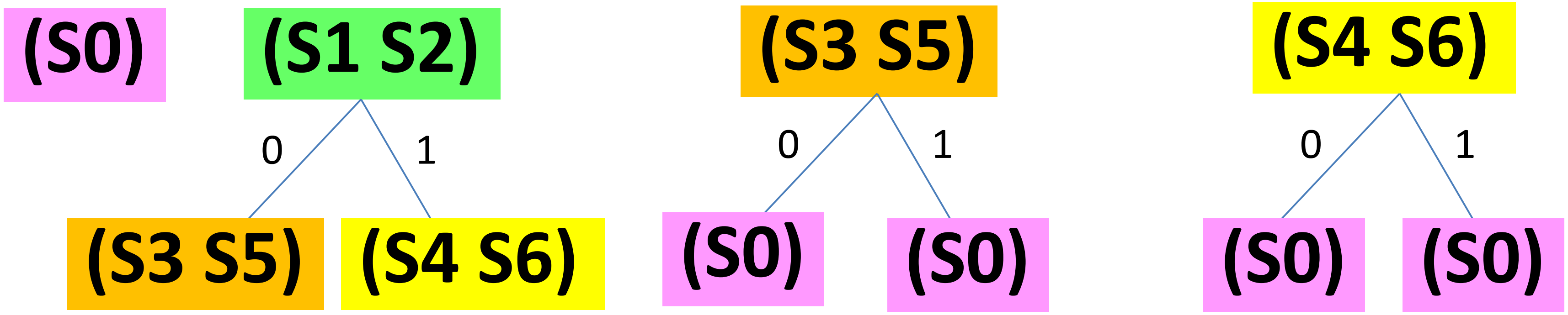
(S0) (S0)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S4 S5 S6)

No further partitions possible



(S0)

(S1 S2)

0 (S3 S5)

1 (S4 S6)

(S3 S5)

0 (S0)

1 (S0)

(S4 S6)

0 (S0)

1 (S0)

State Minimization Example

Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1	S2	0	0
0	S1	S3	S4	0	0
1	S2	S5	S6	0	0
00	S3	S0	S0	0	0
01	S4	S0	S0	1	0
10	S5	S0	S0	0	0
11	S6	S0	S0	1	0

(S0 S1 S2 S3 S4 S5 S6)

↓

(S0) (S1 S2) (S3 S5) (S4 S6)

↓

(S0) (S1‘) (S3’) (S4’

State Minimization Example

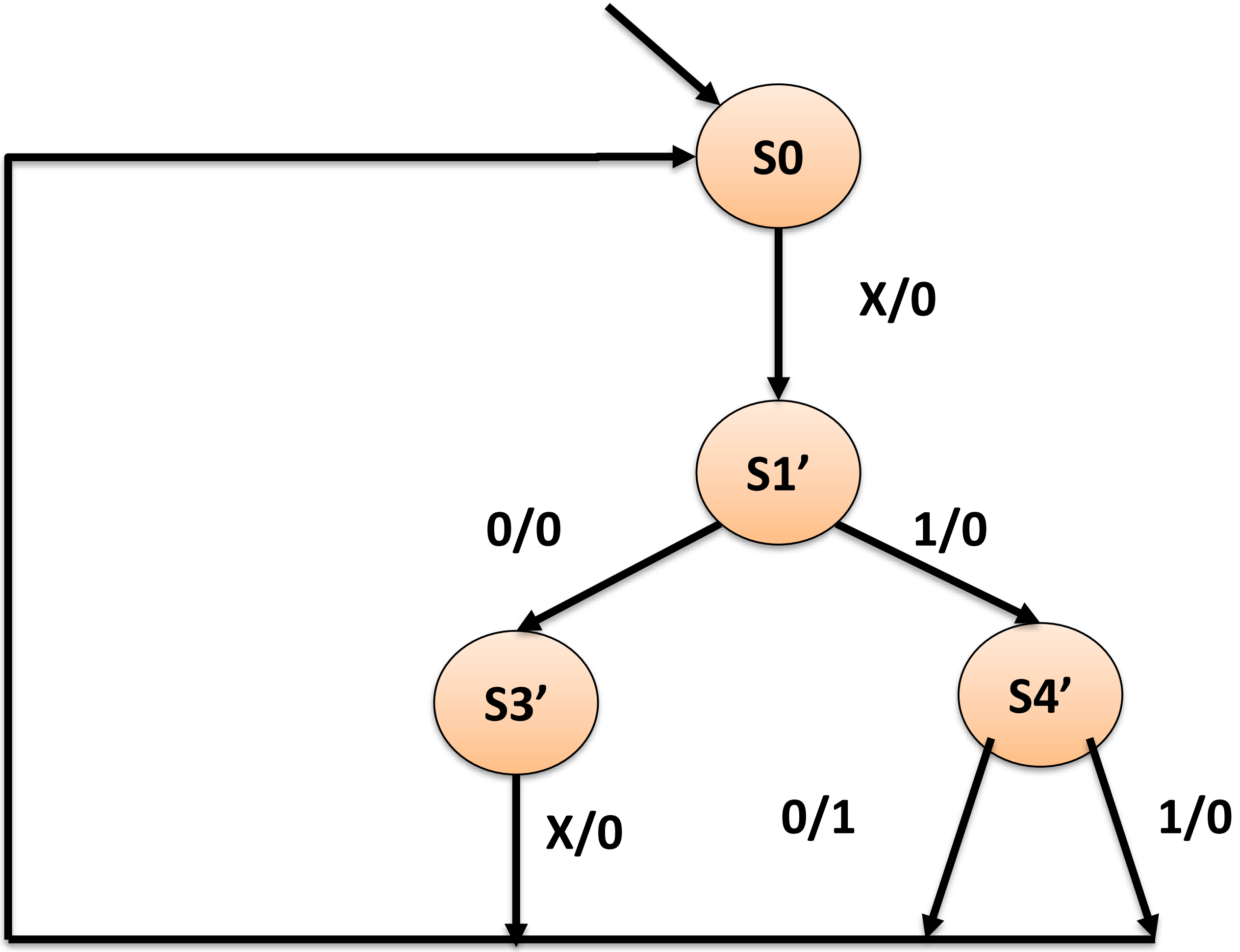
Input	PS	NS		OUTPUT	
		X=0	X=1	X=0	X=1
Reset	S0	S1’	S1’	0	0
0	S1’	S3	S4’	0	0
1	S2	S5	S4’	0	0
00	S3	S0	S0	0	0
01	S4’	S0	S0	1	0
10	S5	S0	S0	0	0
11	S4’	S0	S0	1	0

(S0) (S1 S2) (S3 S5) (S4 S6)

↓

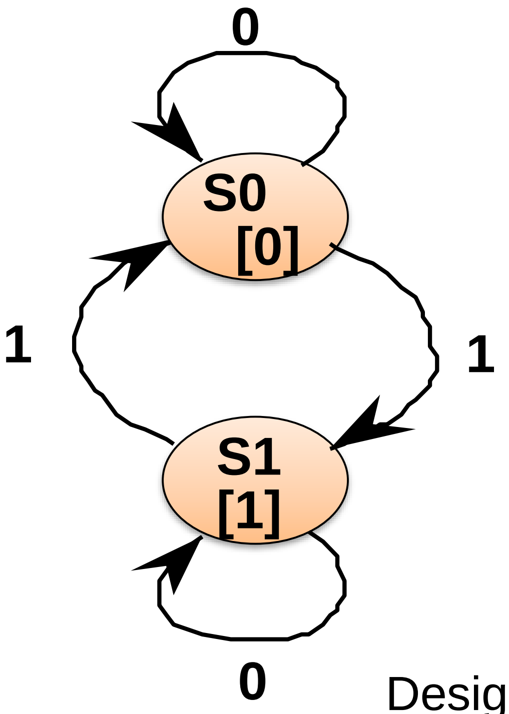
(S0) (S1‘) (S3’) (S4’

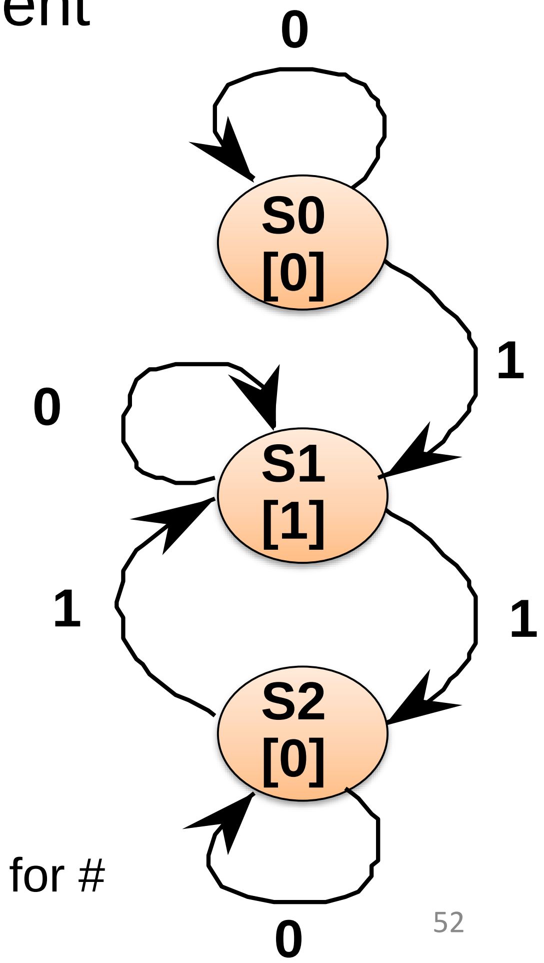
Minimized FSM



Parity Checker Example

- Odd Parity Checker: two alternative state diagrams
  - Identical output behavior on all input strings
  - FSMs are *equivalent*, but require different implementations





Design state diagram without concern for # of states, Reduce later

Partitioning Methd

	Next State		
Present State	X=0	X=1	Output
S_0	S_0	S_1	0
S_1	S_1	S_2	1
S_2	S_2	S_1	0

(S0S1S2)

Based on output

(S0S2) (S1)

Partitioning Methd

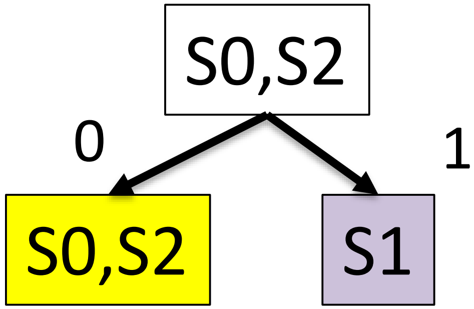
	Next State		
Present State	X=0	X=1	Output
S_0	S_0	S_1	0
S_1	S_1	S_2	1
S_2	S_2	S_1	0

(S0S1S2)

Based on output

(S0S2) **(S1)**

But was not possible with Row matching method



S0,S2 are same

{S0 S2}, {S1} => {S0’}, {S1}