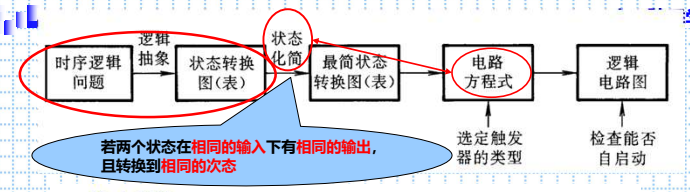
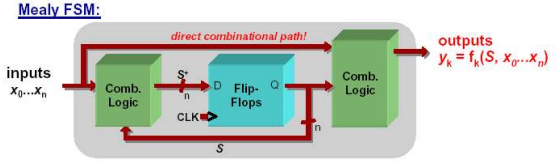
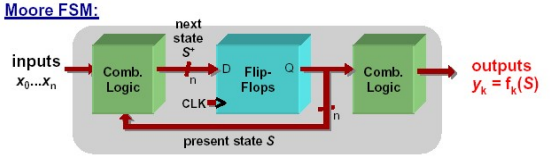


Mealy FSM:

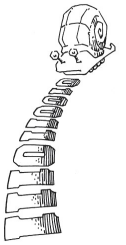


Moore FSM:



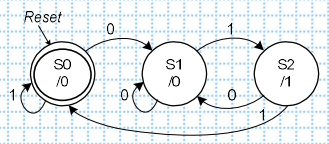
Moore vs. Mealy FSM

- You have a snail that crawls down a paper tape with 1's and 0's on it. The snail smiles whenever the last two digits it has crawled over are 01.
- Design Moore and Mealy FSMs of the snail's brain.

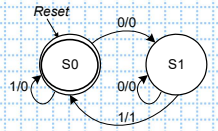


State Transition Diagrams

Moore FSM



Mealy FSM



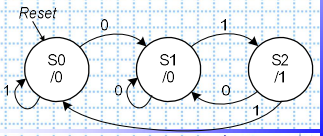
Mealy FSM: arcs indicate input/output

Moore FSM State Transition Table

Current State Q_1	Current State Q_0	Inputs A	Next State Q^*_1	Next State Q^*_0
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		

State	Encoding Q_1Q_0
S0	00
S1	01
S2	10

Moore FSM

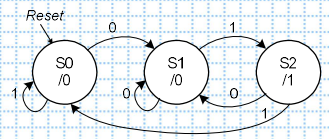


Moore FSM State Transition Table

Current State Q_1	Current State Q_0	Inputs A	Next State Q^*_1	Next State Q^*_0
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

State	Encoding Q_1Q_0
S0	00
S1	01
S2	10

Moore FSM



$Q_1^* = Q_1'Q_0A$

$Q_0^* = Q_1A'+Q_0'A'$

$D_1 = Q_1'Q_0A$

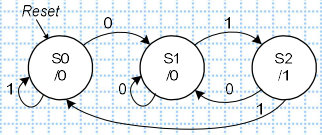
$D_0 = Q_1A'+Q_0'A'$

wang_hong@tsinghua.edu.cn

Moore FSM Output Table

Current State Q_1	Current State Q_0	Output Y
0	0	0
0	1	0
1	0	1
1	1	0

Moore FSM



$Y = Q_1Q_0'$

$D_1 = Q'_0Q_0A$

$D_0 = Q_1A'+Q_0'A'$

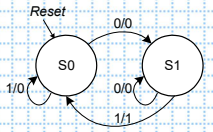
wang_hong@tsinghua.edu.cn

Mealy FSM State Transition & Output Table

Current State Q_0	Input A	Next State Q^*_0	Output Y
0	0		
0	1		
1	0		
1	1		

State	Encoding Q_0
S0	0
S1	1

Mealy FSM



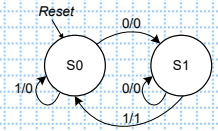
wang_hong@tsinghua.edu.cn

Mealy FSM State Transition & Output Table

Current State Q_0	Input A	Next State Q^*_0	Output Y
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	1

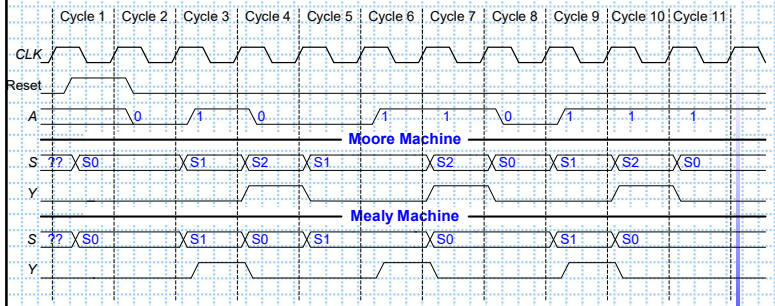
State	Encoding Q_0
S0	0
S1	1

Mealy FSM



wang_hong@tsinghua.edu.cn

Moore & Mealy Timing Diagram



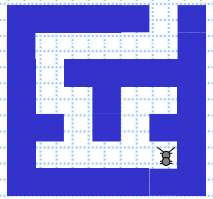
wang_hong@tsinghua.edu.cn

Lets build an *Ant*

8 legs?

- SENSORS: antennae L and R, each 1 if in contact with something.
- ACTUATORS: Forward Step F, ten-degree turns TL and TR (left, right).

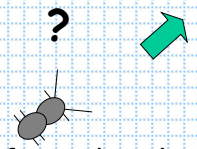
GOAL: Make our ant smart enough to get out of a maze like:



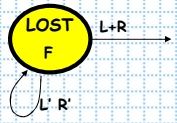
STRATEGY: "Right antenna to the wall"

wang_hong@tsinghua.edu.cn

Lost in space



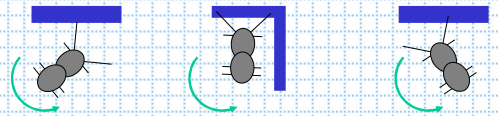
Action: Go forward until we hit something.



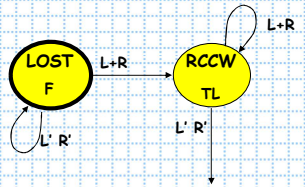
"lost" is the initial state

wang_hong@tsinghua.edu.cn

Bonk!

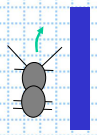


Action: Turn left (CCW) until we don't touch anymore

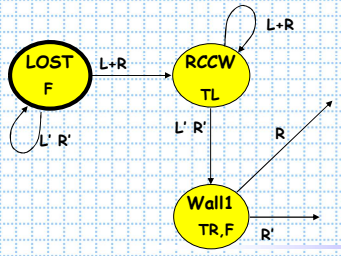


wang_hong@tsinghua.edu.cn

A little to the right…

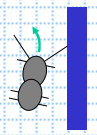


Action: Step and turn right a little, look for wall

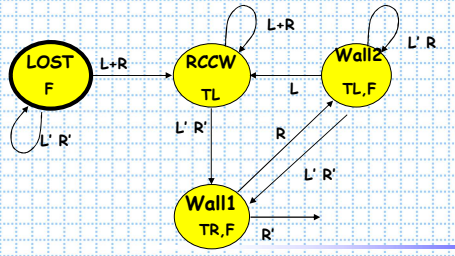


wang_hong@tsinghua.edu.cn

Then a little to the left

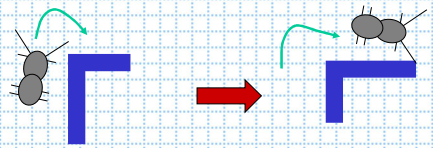


Action: Step and turn left a little, till not touching (again)

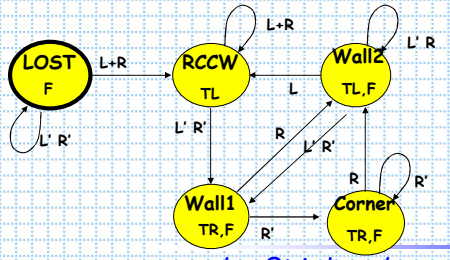


wang_hong@tsinghua.edu.cn

Dealing with corners



Action: Step and turn right until we hit perpendicular wall



wang_hong@tsinghua.edu.cn

Equivalent State Reduction

Observation: $S_i \equiv S_j$ if

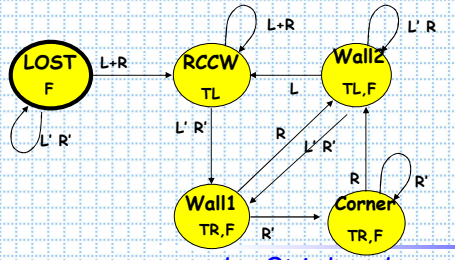
1. States have identical outputs;

AND

2. Every input →equivalent states.

Reduction Strategy:

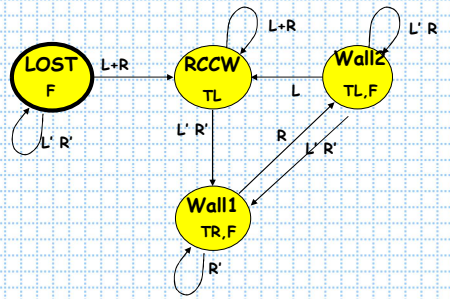
Find pairs of equivalent states, MERGE them.



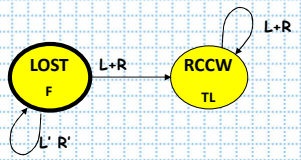
wang_hong@tsinghua.edu.cn

An Evolutionary Step

Merge equivalent states Wall1 and Corner into a single new, combined state.



Building the Transition Table



S	L	R	S*	TR	TL	F
00	0	0	00	0	0	1
00	1	–	01	0	0	1
00	0	1	01	0	0	1
01	1	–	01	0	1	0
01	0	1	01	0	1	0

Implementation Details

	S_1S_0 S	L	R	S_1S_0 S*	TR	TL	F
LOST	00	0	0	00	0	0	1
	00	1	–	01	0	0	1
	00	0	1	01	0	0	1
RCCW	01	1	–	01	0	1	0
	01	0	1	01	0	1	0
	01	0	0	10	0	1	0
WALL1	10	–	0	10	1	0	1
	10	–	1	11	1	0	1
WALL2	11	1	–	01	0	1	1
	11	0	0	10	0	1	1
	11	0	1	11	0	1	1

Complete Transition table

S1* S_1S_0

LR \ S_1S_0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	1
10	0	0	0	1

$$S_1^* = S_1S'_0 + L'S_1 + L'R'S_0$$

S0* S_1S_0

LR \ S_1S_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	0

$$S_0^* = R + LS'_1 + LS_0$$

Ant Schematic