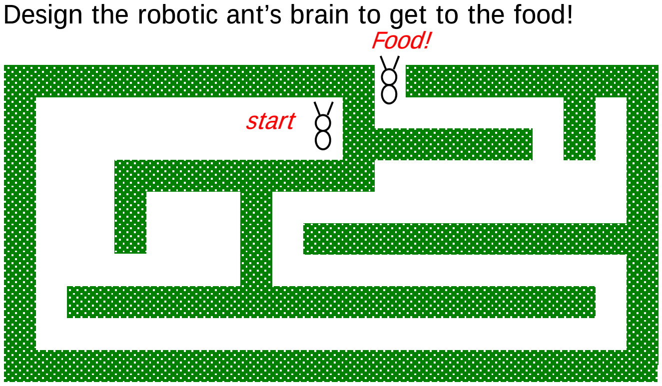
Lecture 19

- Logistics
  - HW7 due now
  - A few days off before HW8 kicks in
  - Midterm review session tomorrow 4:15 EEB125
  - Midterm 2 in class (45min long, starts at 10:35am)
- Last lecture
  - Moore and Mealy machines
- Today
  - A bigger example: Hungry Robot Ant in Maze

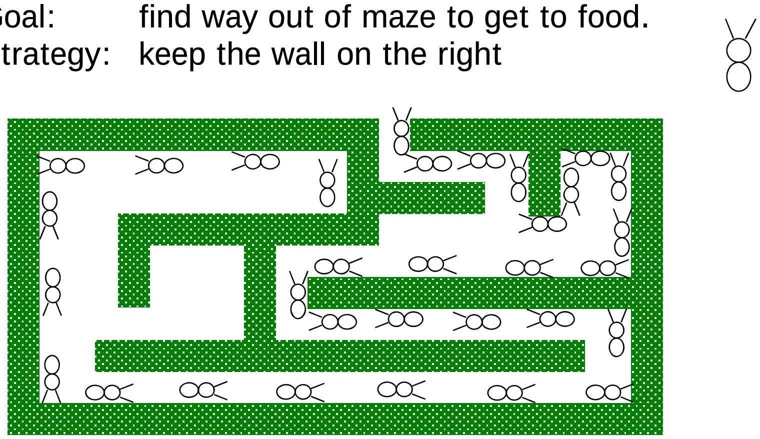
Robotic ant in a maze

- Robot ant, physical maze
  - Maze has no islands
  - Corridors are wider than ant
  - Design the robotic ant's brain to get to the food!



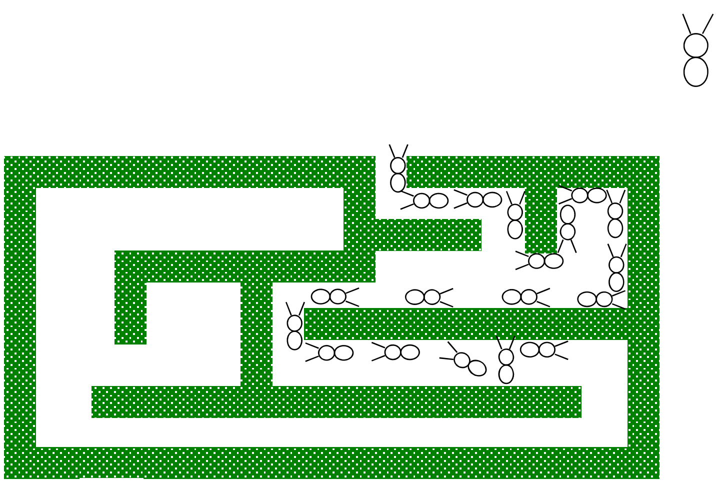
Robot ant specifics

- Sensors: L and R antennae, 1 if touching wall
- Actuators: F - forward step, TL/TR - turn left/right
- Goal: find way out of maze to get to food.
- Strategy: keep the wall on the right



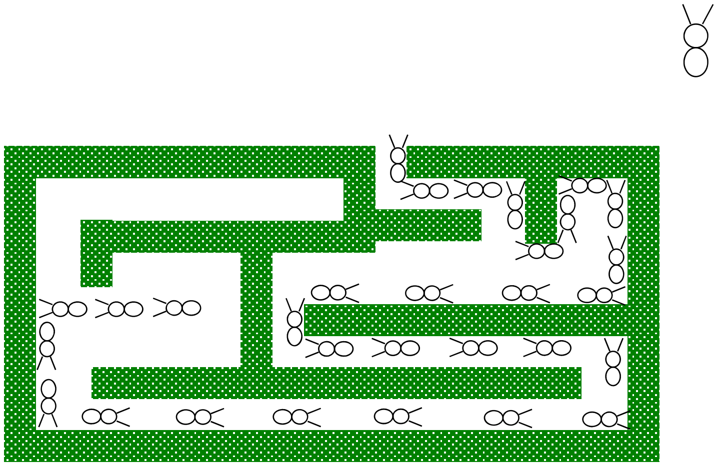
Example: ant brain (special case 1)

- Left (L) Antenna touching the wall



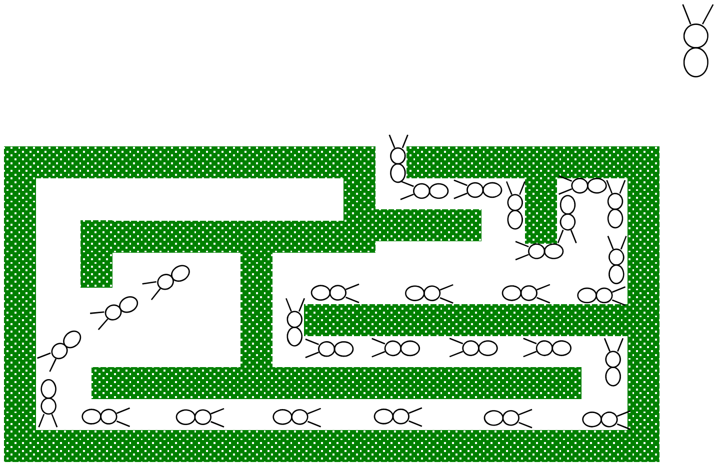
Example: ant brain (special case 2)

- Ant Lost

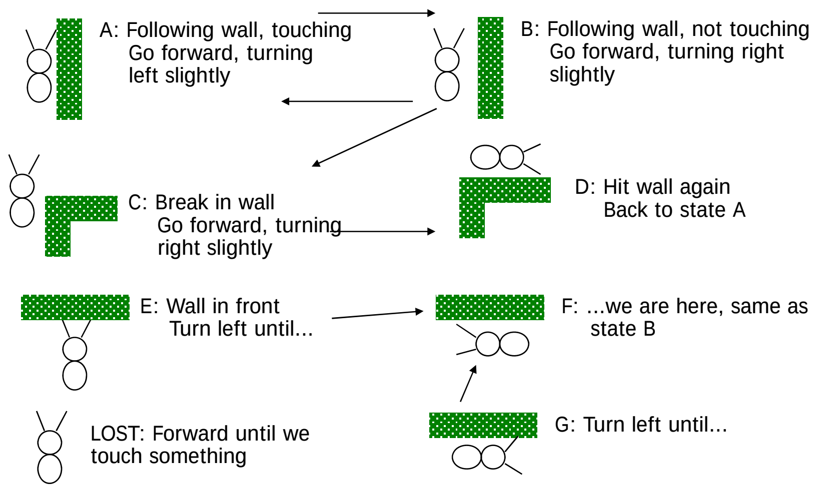


Example: ant brain (special case 2)

- Ant Lost (another example)



Robot Ant behavior



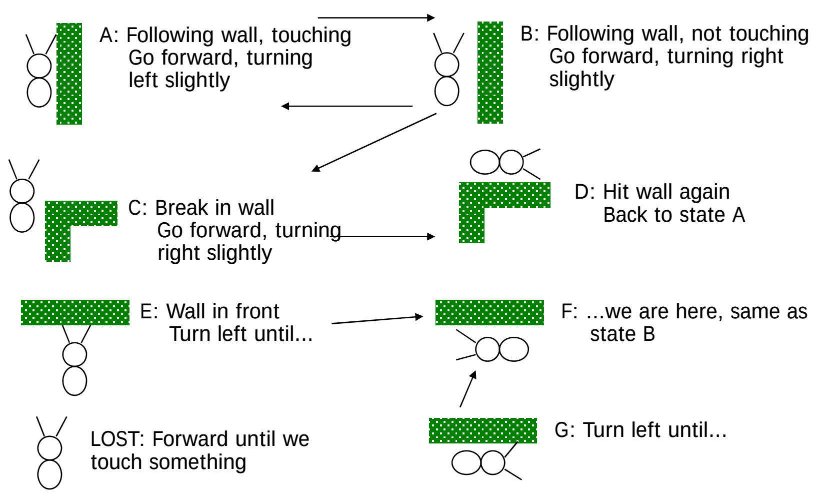
Notes & strategy

- Notes
  - Maze has no islands
  - Corridors are wider than ant
  - Don't worry about startup
  - Assume a Moore machine
  - Assume D flip-flops
- Strategy
  - Keep the wall on the right

Design the ant-brain FSM

1. State diagram
2. State-transition table
3. State minimization
4. State encoding
5. Next-state logic minimization
6. Implement the design

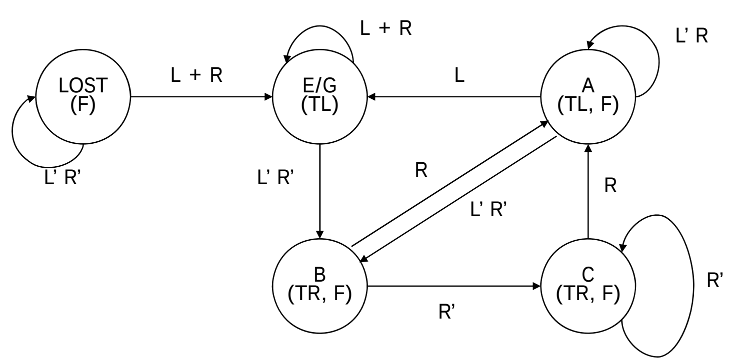
Robot Ant behavior



Notations

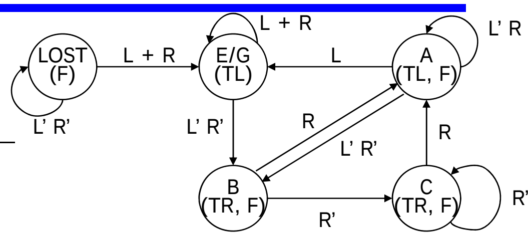
- Sensors on L and R antennae
  - Sensor = “1” if touching wall; “0” if not touching wall
    - L'R' ≡ no wall
    - L'R ≡ wall on right
    - LR' ≡ wall on left
    - LR ≡ wall in front
- Movement
  - F ≡ forward one step
  - TL ≡ turn left slightly
  - TR ≡ turn right slightly

1. State Diagram



2. State Transition Table

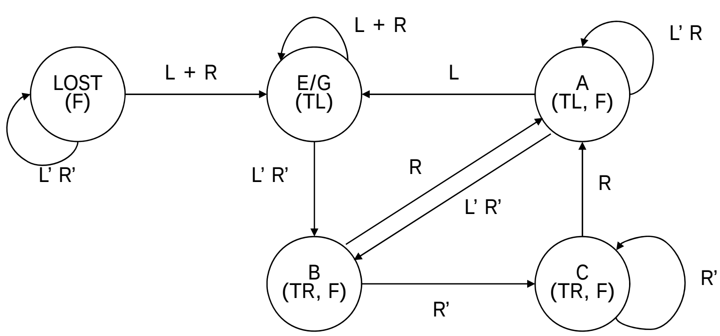
- Using symbolic states and outputs



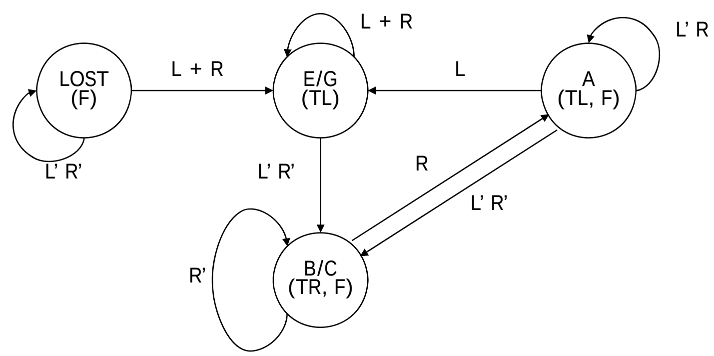
state	L	R	next state	outputs
LOST	0	0	LOST	F
LOST	–	1	E/G	F
LOST	1	–	E/G	F
E/G	0	0	B	TL
E/G	–	1	E/G	TL
E/G	1	–	E/G	TL
B	–	0	C	TR, F
B	–	1	A	TR, F
A	0	0	B	TL, F
A	0	1	A	TL, F
A	1	–	E/G	TL, F
C	–	0	C	TR, F
C	–	1	A	TR, F

3. State minimization

- Any equivalent states?



Sure! Now you can represent states with 2 bits



4. State encoding

state	L	R	next state	outputs
LOST	0	0	LOST	F
LOST	–	1	E/G	F
LOST	1	–	E/G	F
E/G	0	0	B/C	TL
E/G	0	1	E/G	TL
E/G	1	–	E/G	TL
A	0	0	B	TL, F
A	–	1	A	TL, F
A	1	–	E/G	TL, F
B/C	–	0	B/C	TR, F
B/C	–	1	A	TR, F

state	inputs		next state	outputs		
X,Y	L	R	X+,Y+	F	TR	TL
0 0	0	0	0 0	1	0	0
0 0	-	1	0 1	1	0	0
0 0	1	-	0 1	1	0	0
0 1	0	0	1 1	0	0	1
0 1	0	1	0 1	0	0	1
0 1	1	-	0 1	0	0	1
1 0	0	0	1 1	1	0	1
1 0	-	1	1 0	1	0	1
1 0	1	-	0 1	1	0	1
1 1	-	0	1 1	1	1	0
1 1	-	1	1 0	1	1	0

5. Next state logic minimization

state X,Y	inputs L R	next state X',Y'	outputs F TR TL
0 0	0 0	0 0	1 0 0
0 0	- 1	0 1	1 0 0
0 0	1 -	0 1	1 0 0
0 1	0 0	1 1	0 0 1
0 1	0 1	0 1	0 0 1
0 1	1 -	0 1	0 0 1
1 0	0 0	1 1	1 0 1
1 0	- 1	1 0	1 0 1
1 0	1 -	0 1	1 0 1
1 1	- 0	1 1	1 1 0
1 1	- 1	1 0	1 1 0

X+

0	1	1	1
0	0	1	1
0	0	1	0
0	0	1	0

Y+

0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	1

F

1	0	1	1
1	0	1	1
1	0	1	1
1	0	1	1

TR

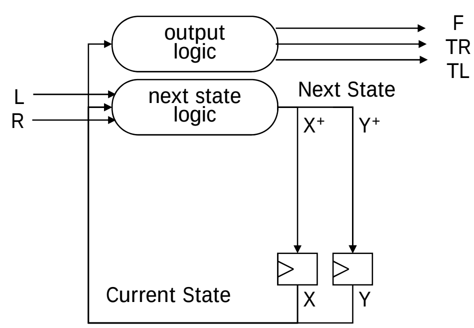
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0

TL

0	1	0	1
0	1	0	1
0	1	0	1
0	1	0	1

6. Circuit Implementation

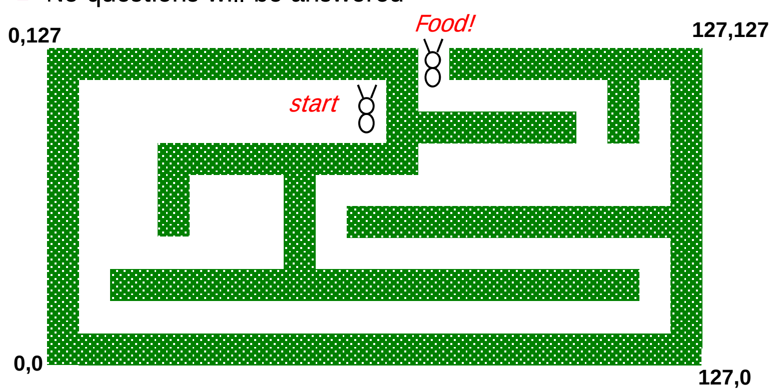
- Outputs are a function of the current state only - Moore machine



Extra credit
(worth 15pts equivalent in a midterm)

Design the robotic ant's brain with virtual maze representation

- Due last day in class, Friday, June 6; printouts only
- Graded on clarity and completeness of explanation
- No questions will be answered



The maze

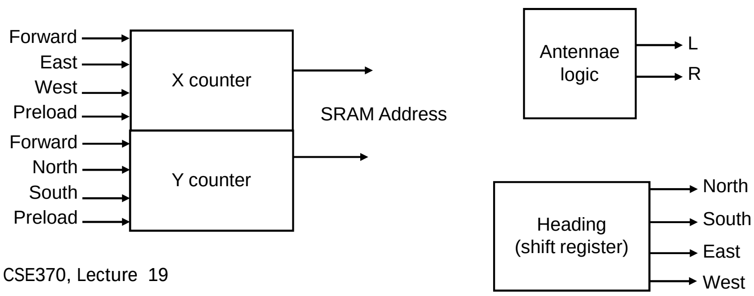
- Virtual maze
  - 128 × 128 grid
    - Stored in memory
    - 16384 8-bit words
  - *YX* is maze addresses
    - *X* is the ant's horizontal position (7 bits)
    - *Y* is the ant's vertical position (7 bits)
  - Each memory location says
    - 00000001 ≡ No wall
    - 00000010 ≡ North wall
    - 00000100 ≡ West wall
    - 00001000 ≡ South wall
    - 00010000 ≡ East wall
    - 00100000 ≡ Exit

Can have multiple walls
Example: 00001100
⇒ Walls on South and East

Design of different components

Predesigned:

Submit the designs for:



Recommendations

- Memory controller
  - Move horizontally: Increment or decrement *X*
  - Move vertically: Increment or decrement *Y*
- Shift register for heading
  - N: 0001
  - W: 0010
  - S: 0100
  - E: 1000
  - Rotate right when ant turns right
  - Rotate left when ant turns left
- Combinational logic for antennae logic