Bayes Filter II

Fast Robots, ECE4160/5160, MAE 4190/5190

E. Farrell Helbling, 4/9/26
Slides adapted from Kirstin Petersen

Class Action Items

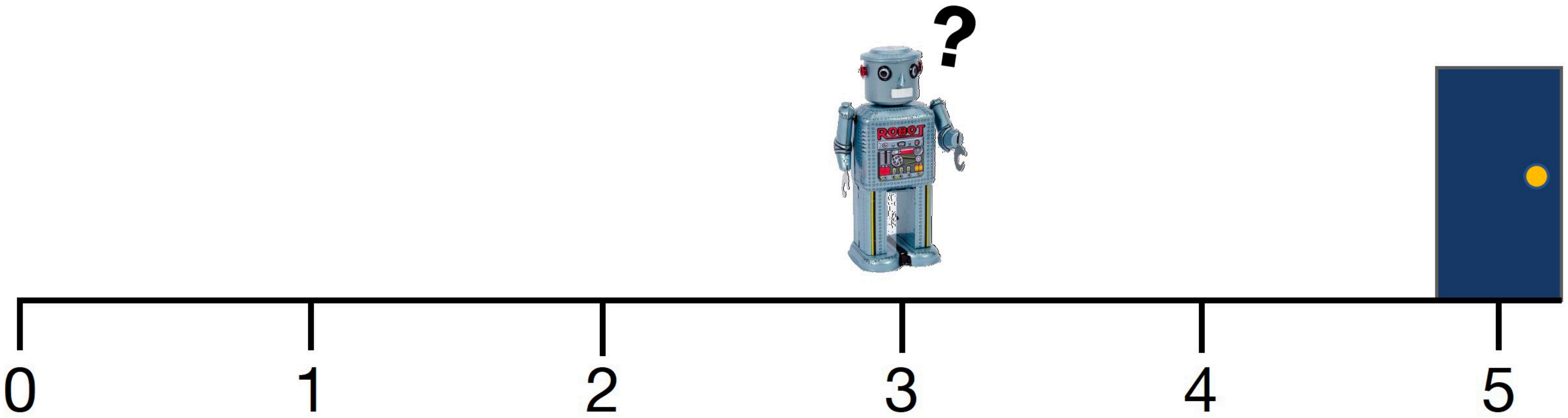
- Next class:
  - Flipped classroom to set up the simulator
  - Please bring your laptop!

Bayes Filter

- The robot performs a series of alternating actions/ measurements
- Given:
  - Sensor model: $p(z_t \mid u_t)$
  - Action model: $p(x_t \mid u_t, x_{t-1})$
  - Initial conditions: $p(x_0)$
- Compute:
  - State of dynamic system
  - Posterior of the state (belief): $bel(x_t) = p(x_t \mid u_1, z_1, \ldots, u_t, z_t)$

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. **for** all x_t do
3. $\overline{bel}(x_t) = \Sigma_{x_{t-1}} \; p(x_t \mid u_t, x_{t-1}) \; bel(x_{t-1})$
4. $bel(x_t) = \eta \; p(z_t \mid x_t) \; \overline{bel}(x_t)$
5. end **for**
6. **return** $bel(x_t)$

Example 1



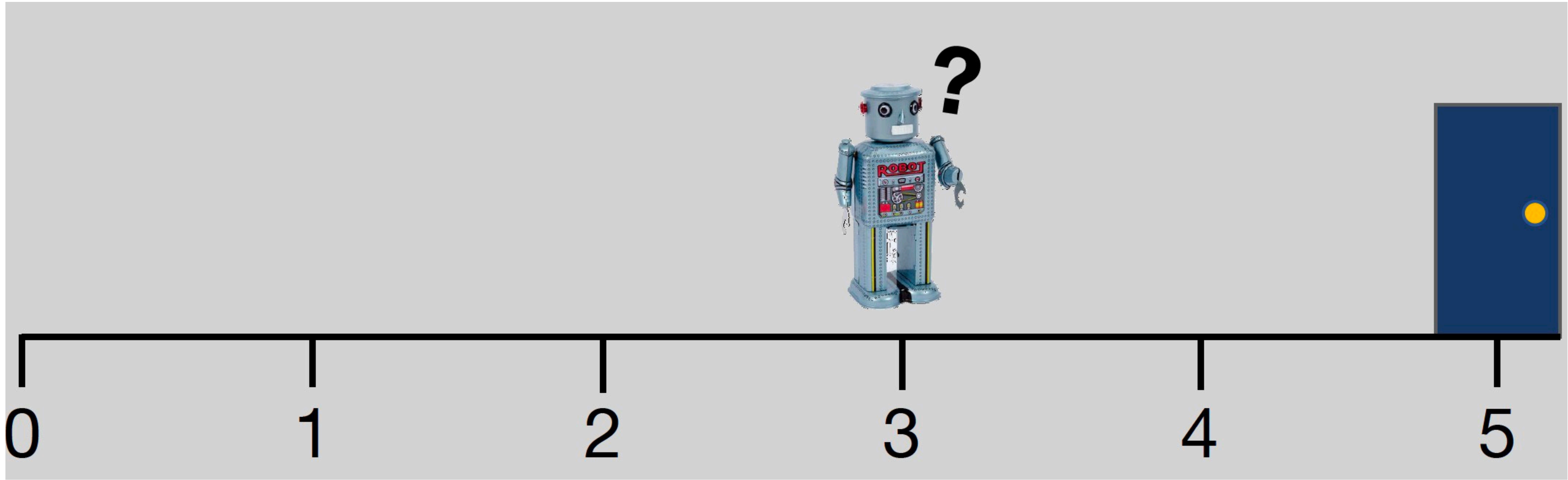
Example 1

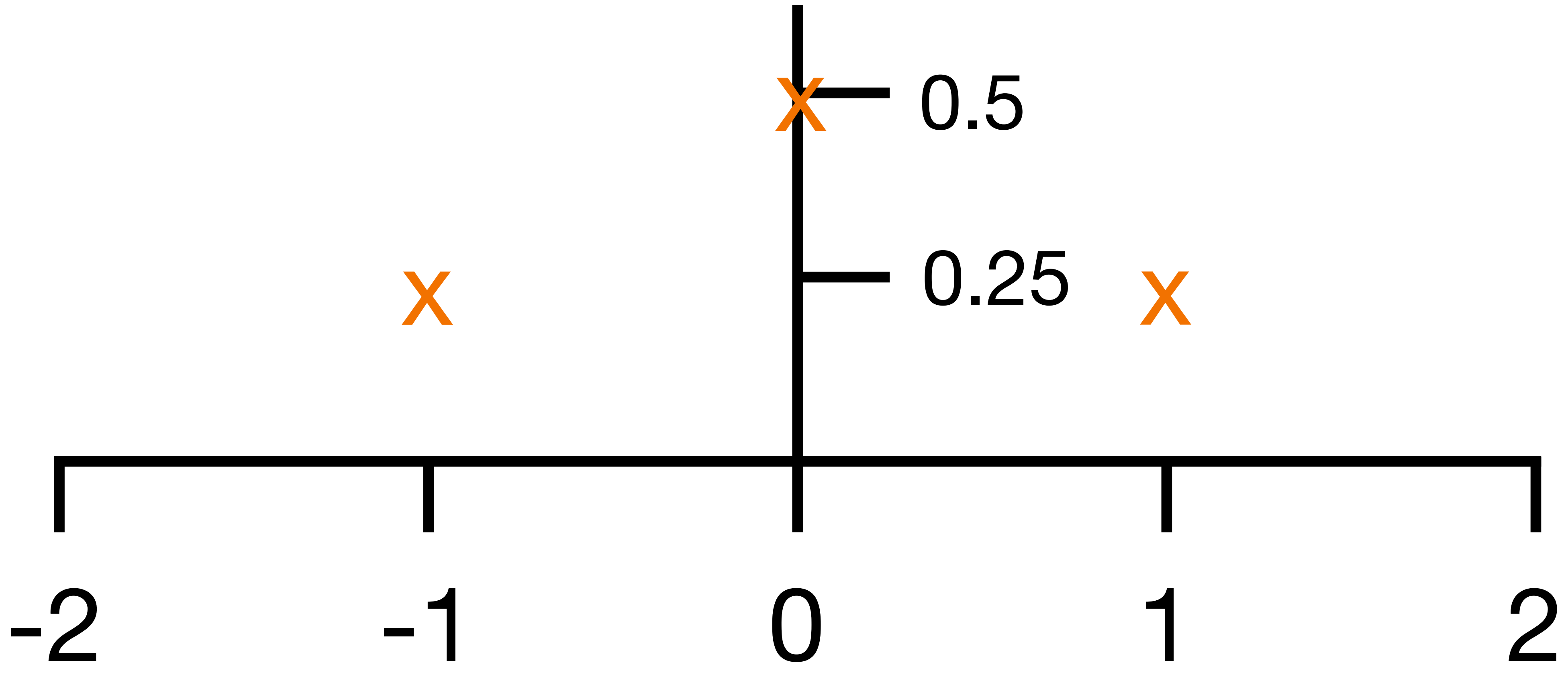
- So, what do we need to run the Bayes Filter?
- Motion model

$$p(x+1\,|\,x,\,u=+1)=0.5$$
$$p(x\,|\,x,\,u=+1)=0.5$$
$$p(x-1\,|\,x,\,u=-1)=0.5$$
$$p(x\,|\,x,\,u=-1)=0.5$$

- Measurement model

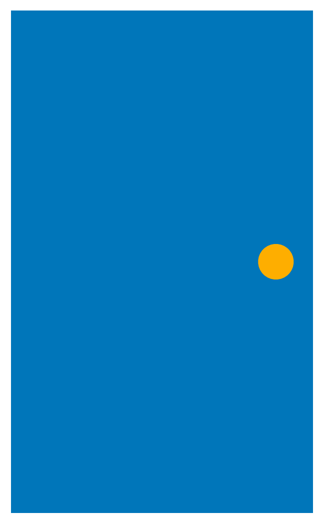
$$p(Z=\text{door}\,|\,X=5)=0.5$$
$$p(Z=\text{door}\,|\,X=4)=0.25$$
$$p(Z=\text{door}\,|\,X=3)=0$$





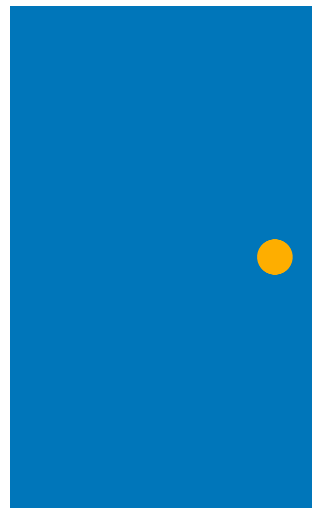
Example 1

At $t = 0$, no information



State	0	1	2	3	4	5
$p(x_0)$						

Example 1



At $t = 0$, no information

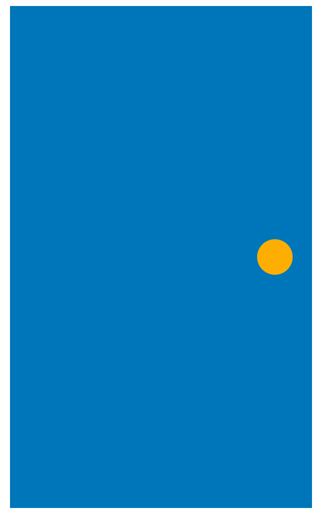
State	0	1	2	3	4	5
$p(x_0)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

At $t = 1,\ U_1 = \mathrm{NOP},\ Z_1 = \mathrm{door}$

State	0	1	2	3	4	5
$p(x_1)$						

Do we need to do the prediction step?

Example 1



At $t = 1,\ U_1 = \text{NOP},\ Z_1 = \text{door}$

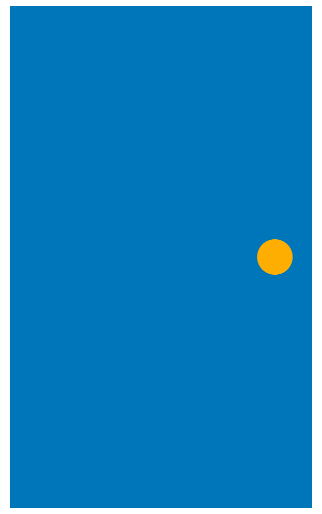
State	0	1	2	3	4	5
$p(x_1)$						

At $t = 2,\ U_2 = -1$

State	0	1	2	3	4	5
$p(x_2)$						

Example 1

At $t = 2,\ U_2 = -1$

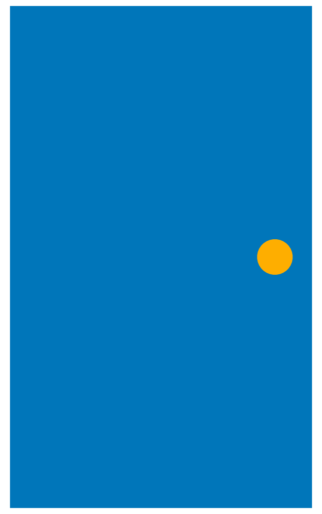


State	0	1	2	3	4	5
$p(x_2)$						

At $t = 2,\ U_2 = -1,\ Z_2 = \text{door}$

State	0	1	2	3	4	5
$p(x_2)$						

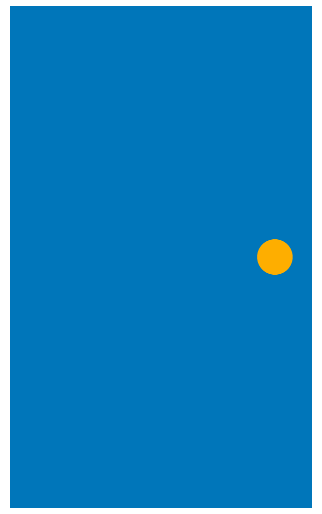
Example 1



At $t = 0$, we are absolutely certain the robot is at state $X_0 = 0$

State	0	1	2	3	4	5
$p(x_0)$						

Example 1



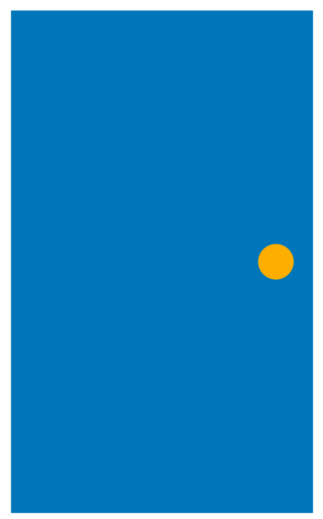
At $t = 0$, we are absolutely certain the robot is at state $X_0 = 0$

State	0	1	2	3	4	5
$p(x_0)$	1	0	0	0	0	0

At $t = 1,\ U_1 = \text{NOP},\ Z_1 = \text{door}$

State	0	1	2	3	4	5
$p(x_1)$						

Example 1



At $t = 0$, we are absolutely certain the robot is at state $X_0 = 0$

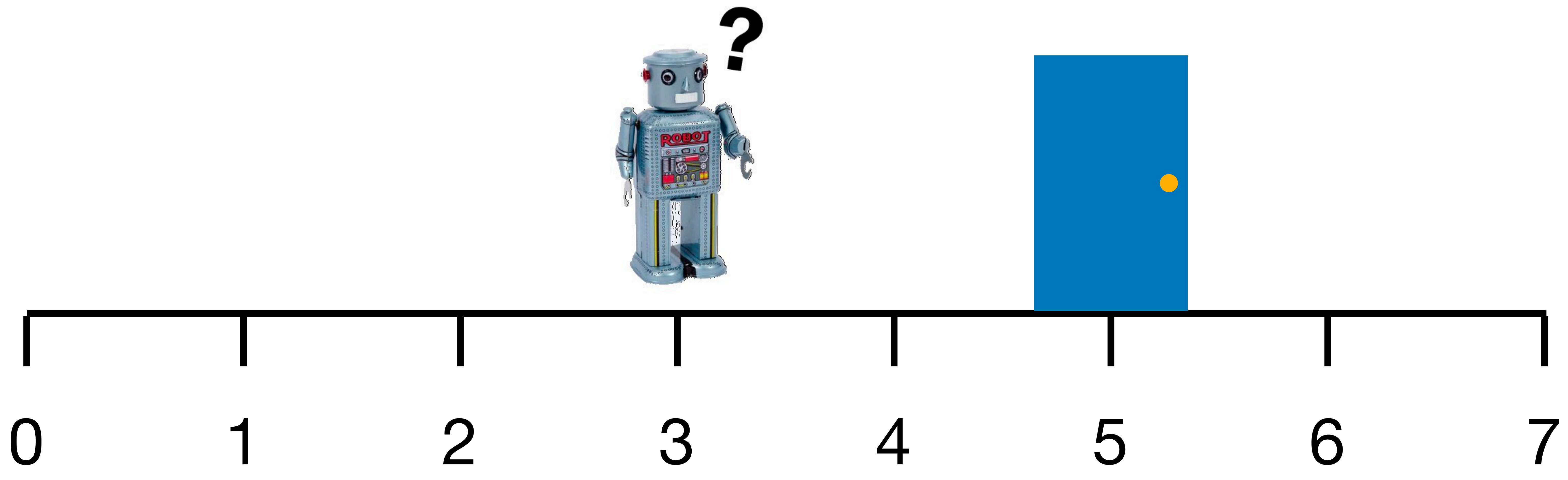
State	0	1	2	3	4	5
$p(x_0)$	$\frac{19}{20}$	$\frac{1}{100}$	$\frac{1}{100}$	$\frac{1}{100}$	$\frac{1}{100}$	$\frac{1}{100}$

At $t = 1,\ U_1 = \text{NOP},\ Z_1 = \text{door}$

State	0	1	2	3	4	5
$p(x_1)$	0	0	0	0	$\frac{1}{3}$	$\frac{2}{3}$

Always believe, even just a little, in the improbable!

Example 2

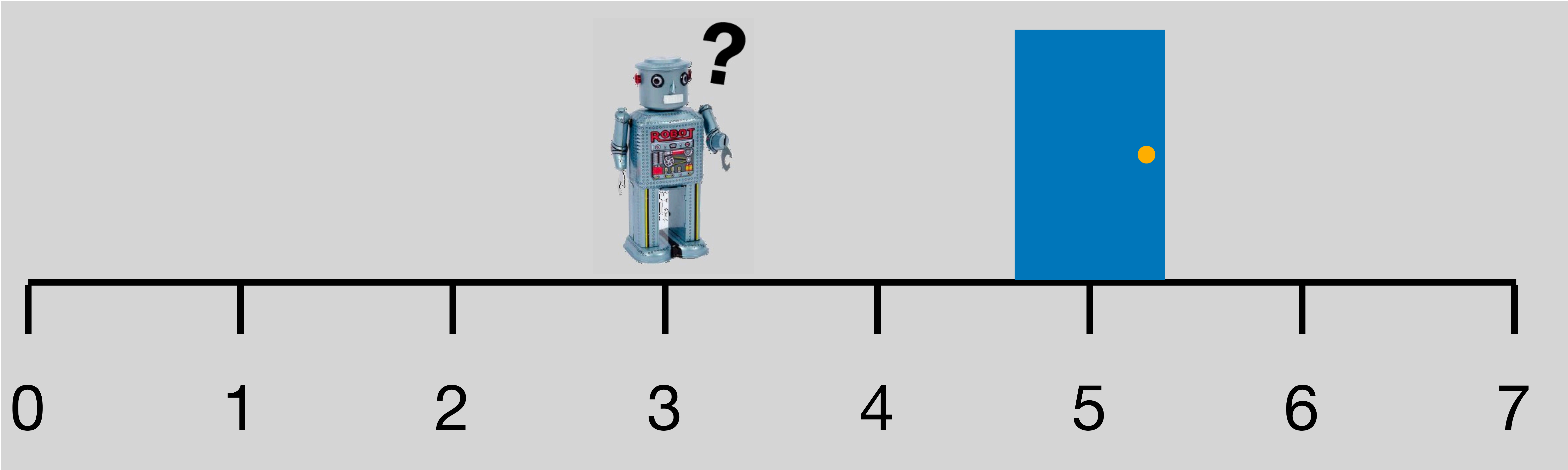


adapted from Prof. Fred Martin at UMass

Example 2

Bayes with Beans

- World
  - 1D continuous, 7 states
  - …door at state 5
- Motion model
  - 80% correct, 20% fail
- Sensor model
  - 90% correct, 10% fail
- Initial belief
- Take an action: +1
- Take a sensor reading: door!

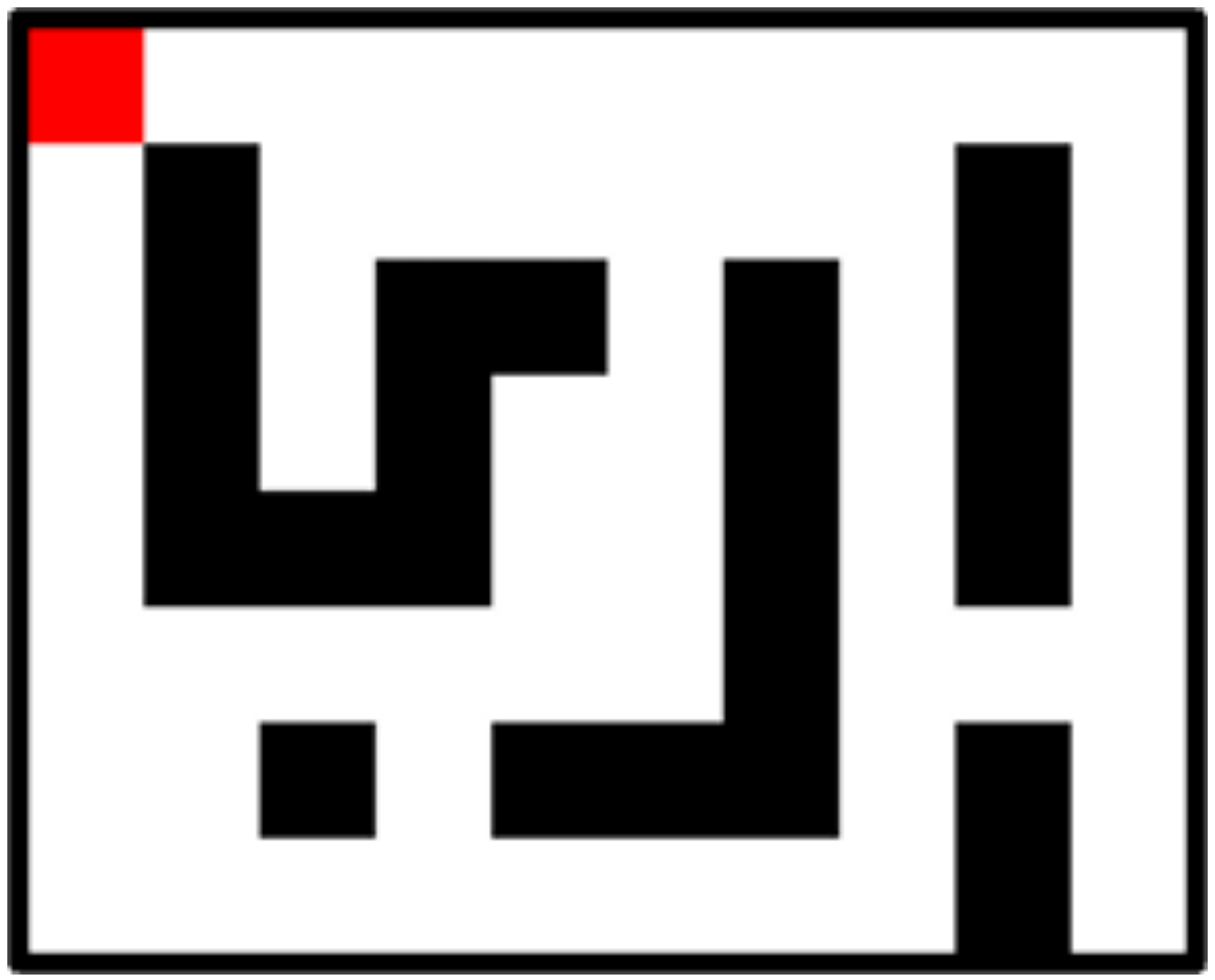


adapted from Prof. Fred Martin at UMass

Example 3

Example 3

- 8x10 discrete world
  - Known map with obstacles and walls
- Robot state
  - Location in the map (no orientation)
  - Initial state is (0,0)



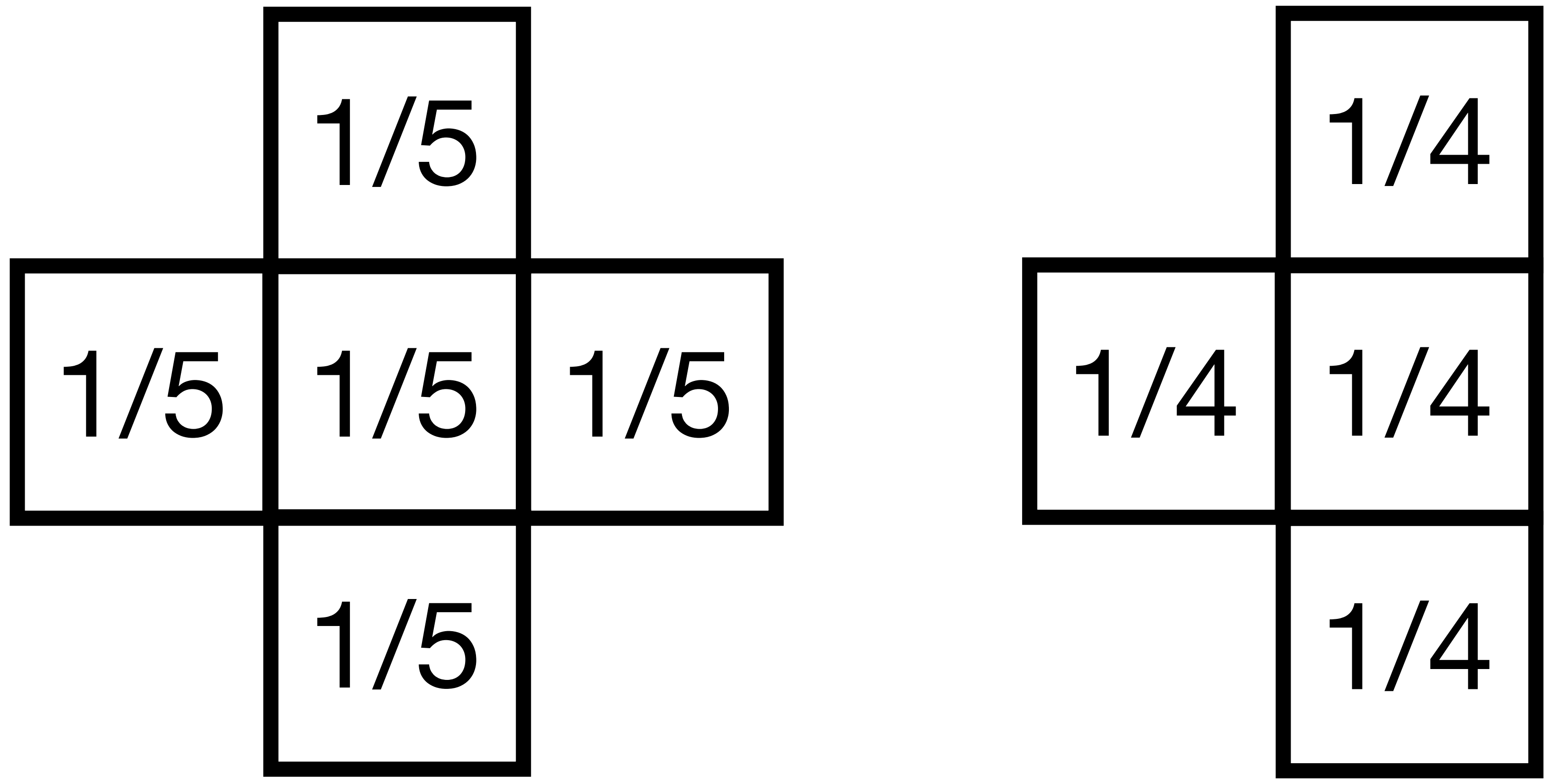
x is the set of possible locations

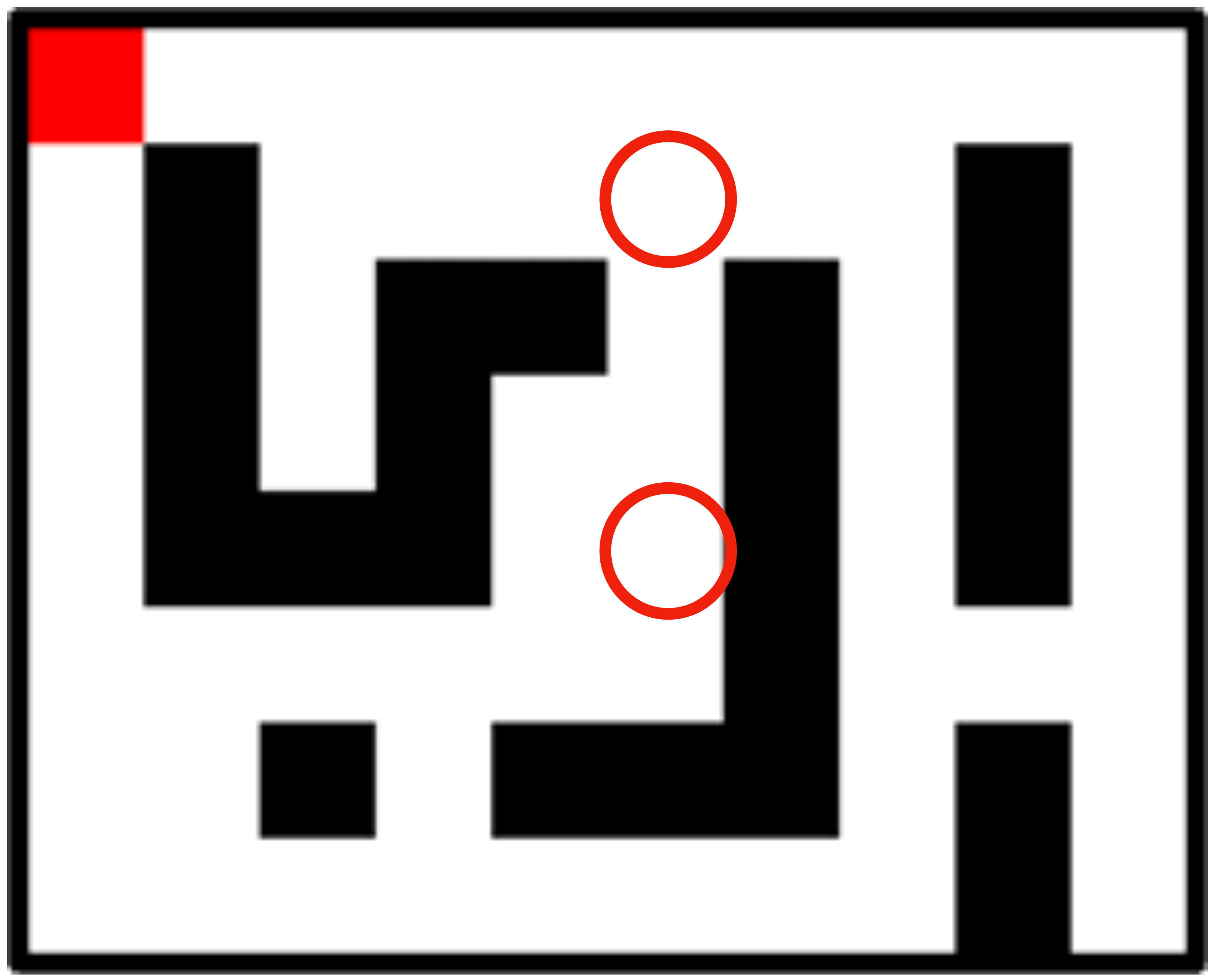
X is one location

Example 3

Transition model

- No matter what I tell my robot to do, it makes a random move or stays in place!





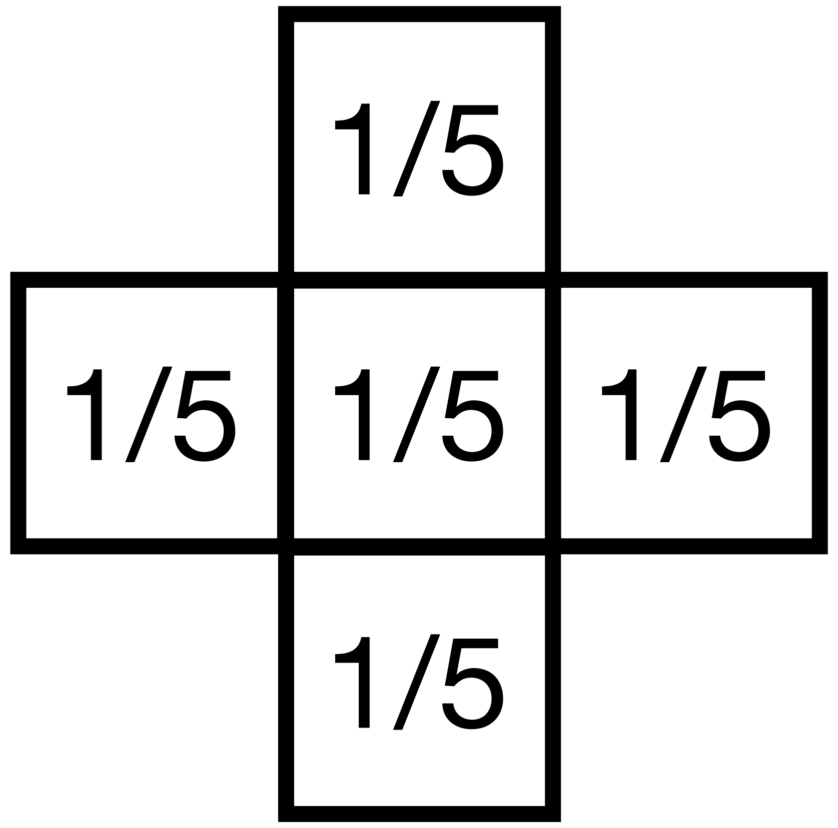
x is the set of possible locations

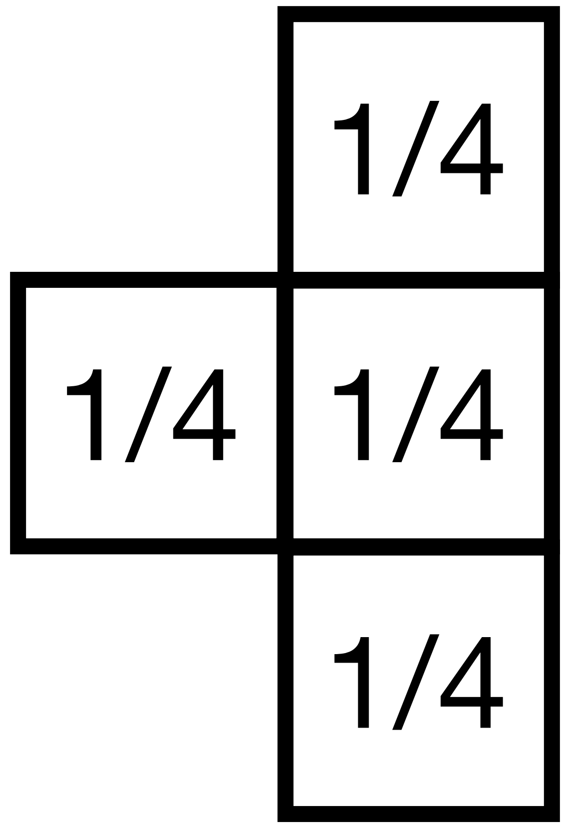
X is one location

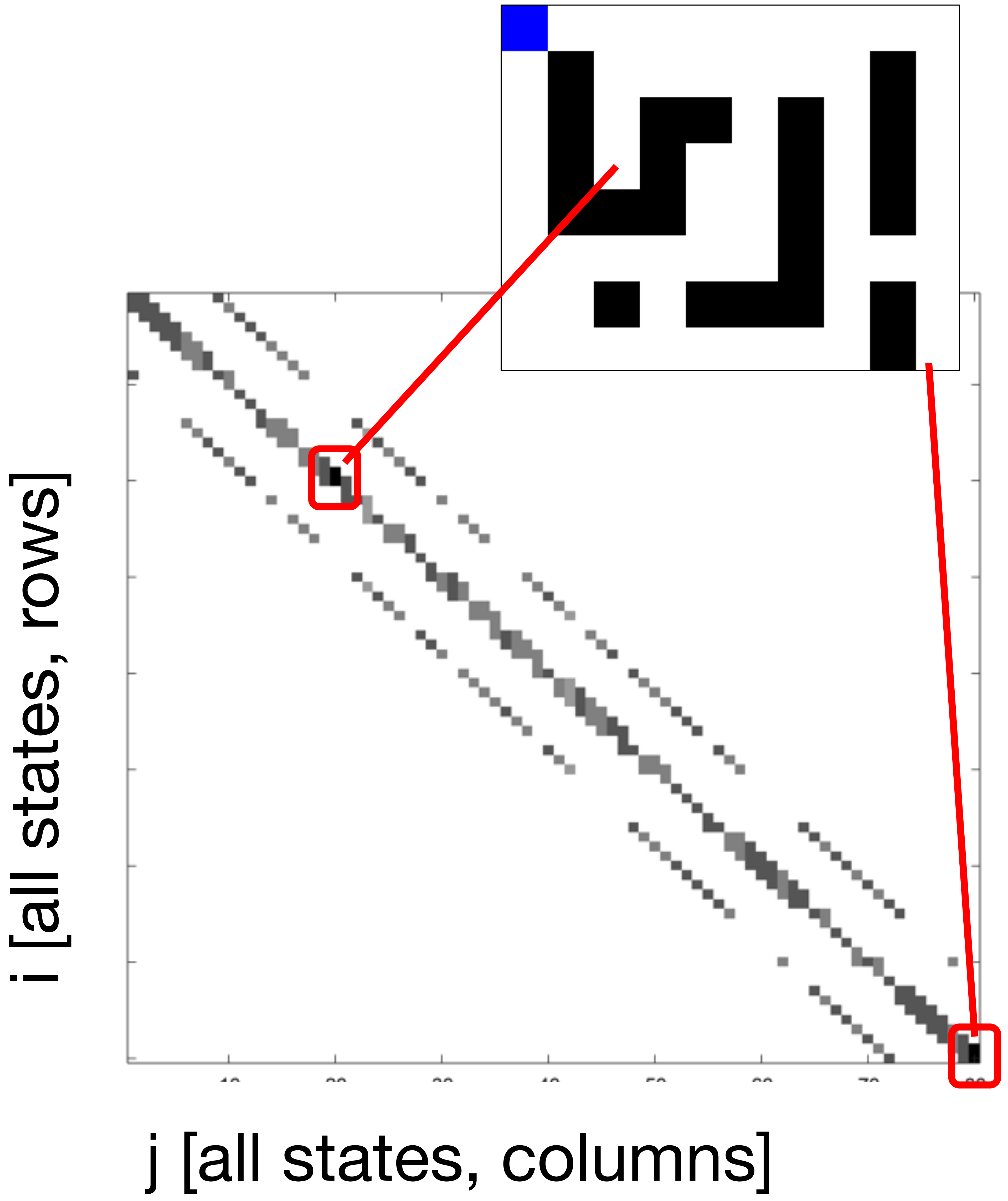
Example 3

Transition model

- No matter what I tell my robot to do, it makes a random move or stays in place!
- Transition matrix, A
  - Probability to move from state *j* to state *i*



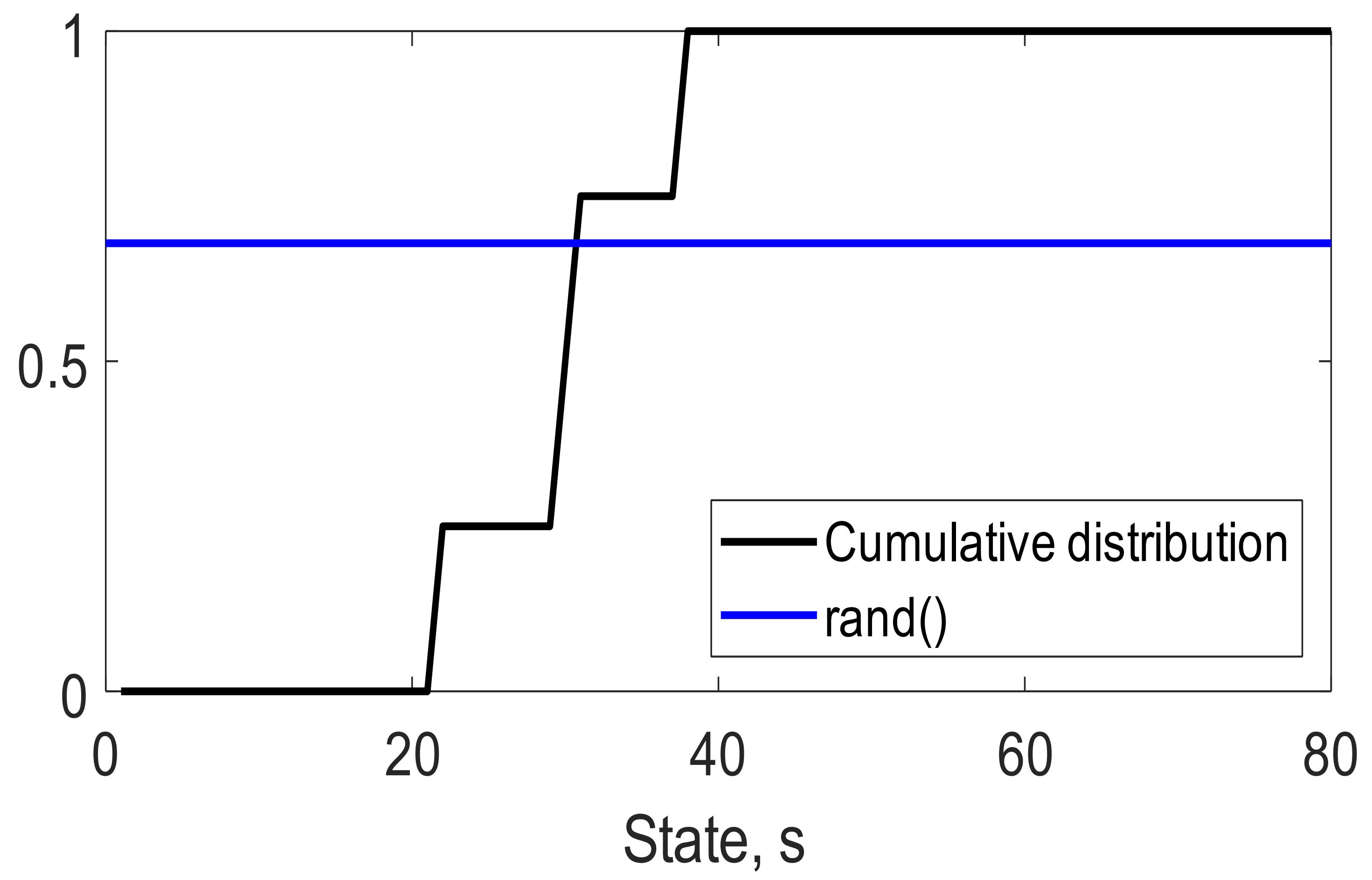


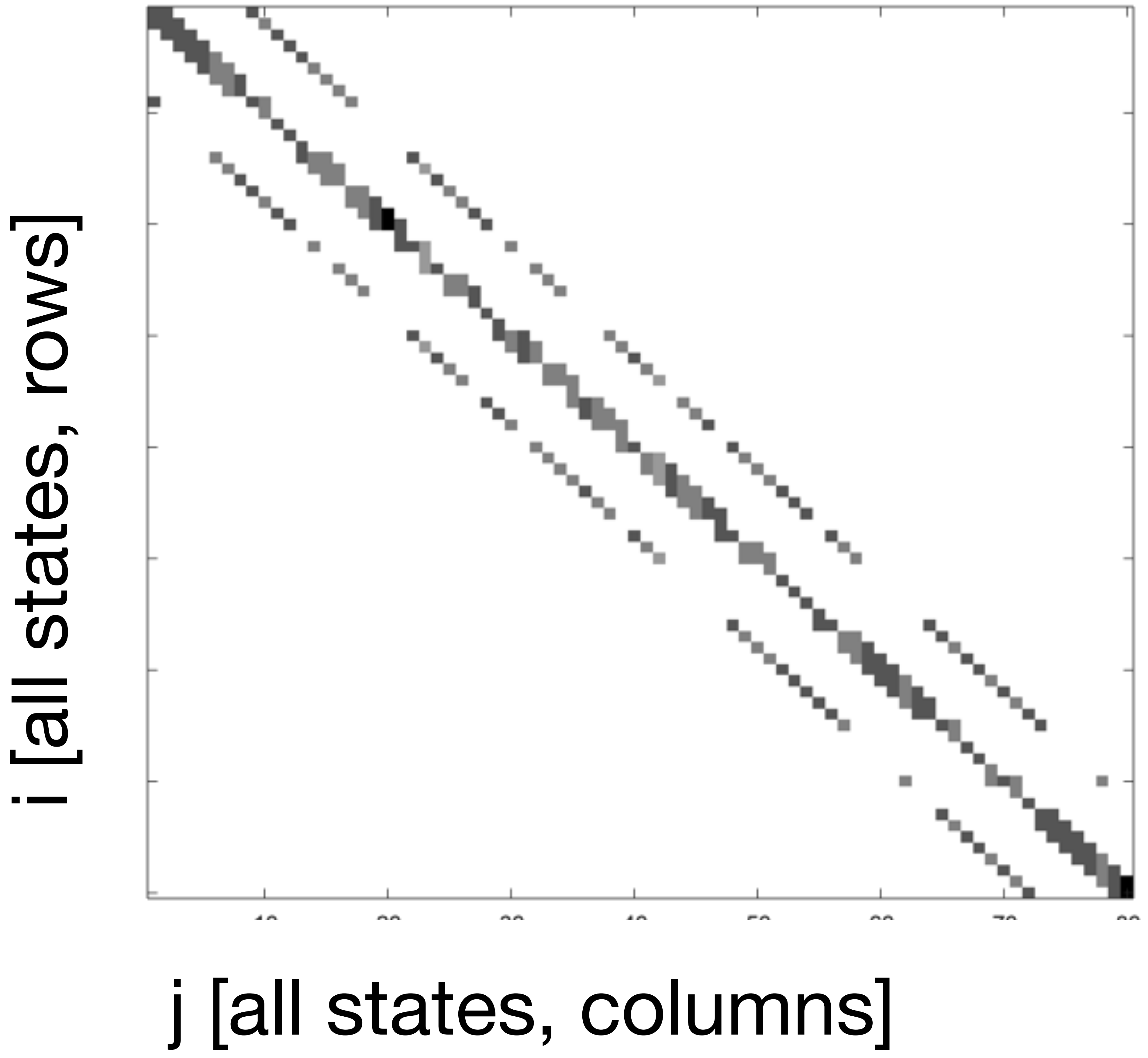


Example 3

Practical implementation

- Set up the world
- Compute the transition matrix, A
- Take actions
  - Cumulative distribution
  - `find(Mtri*A*s >= rand(),1,`first');`





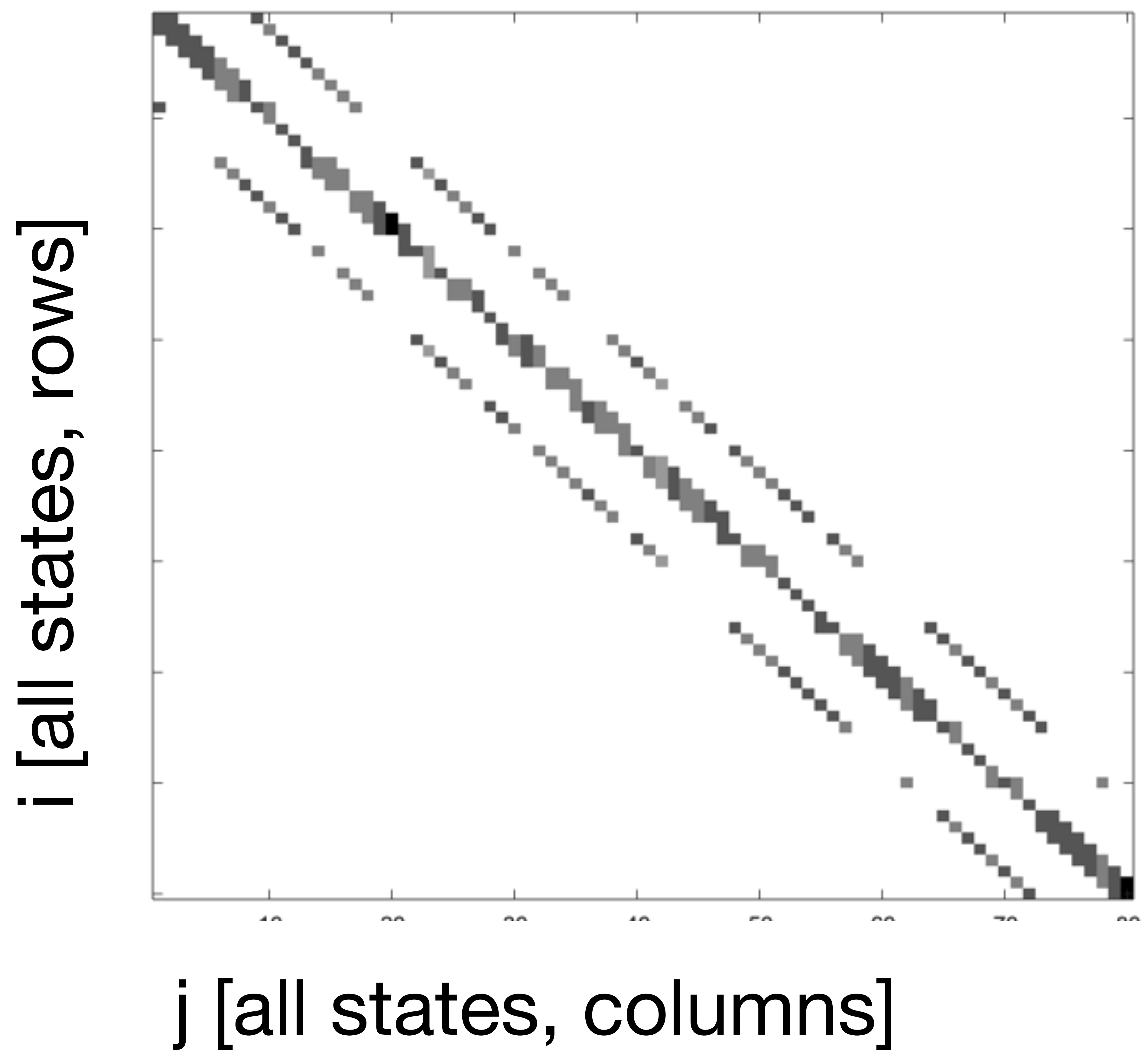
Example 3

Prediction Step

1. **Prediction step** $(bel(x_{t-1}), \not{u_t})$:
2. **for** all x_t do
3. $\overline{bel}(x_t) = \Sigma_{x_{t-1}} \; p(x_t \,|\, \not{u_t}, x_{t-1}) \; bel(x_{t-1})$
4. end **for**

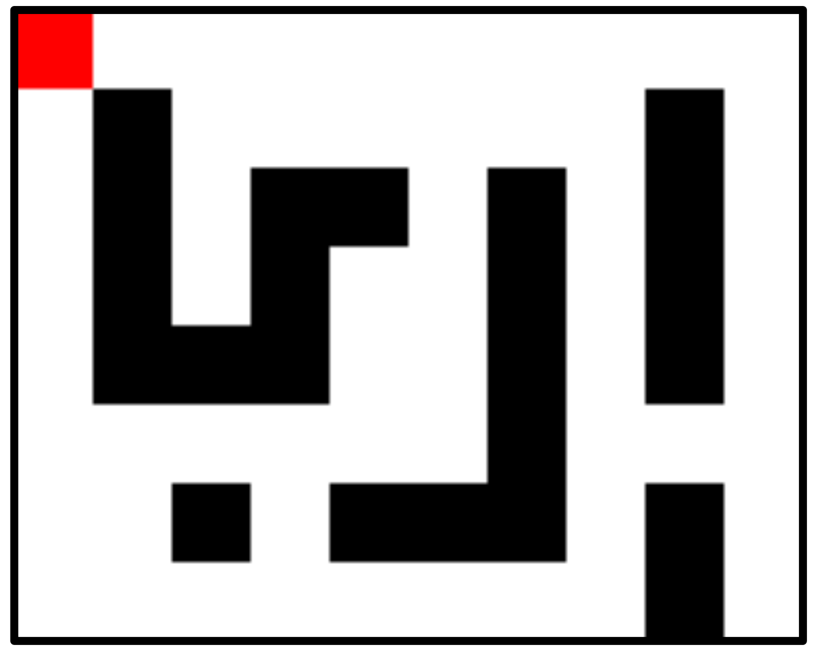
1. **Matrix implementation**
2. $\overline{bel} = A \cdot bel_{t-1}$

where A is the transition matrix (80x80) and bel is the probability distribution over all states (80x1)

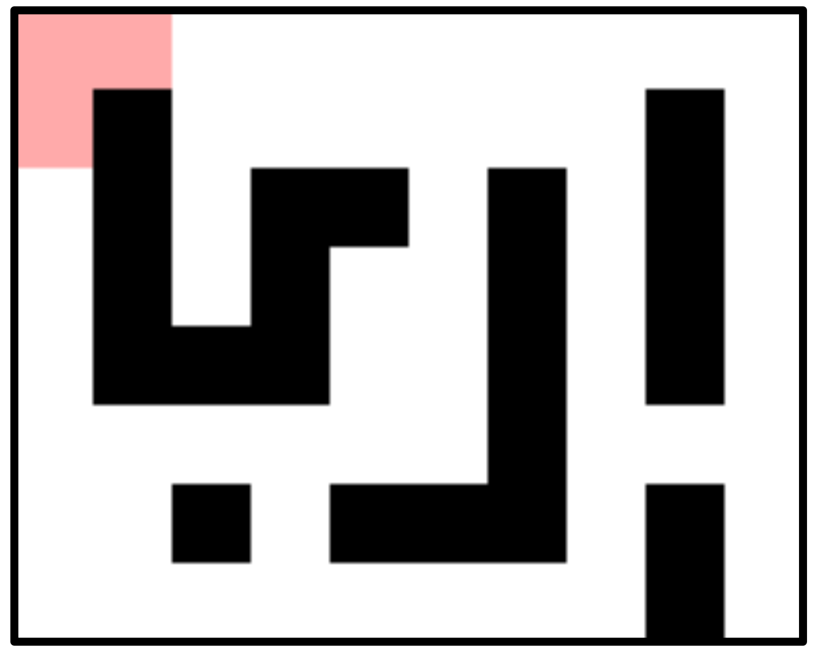


Example 3

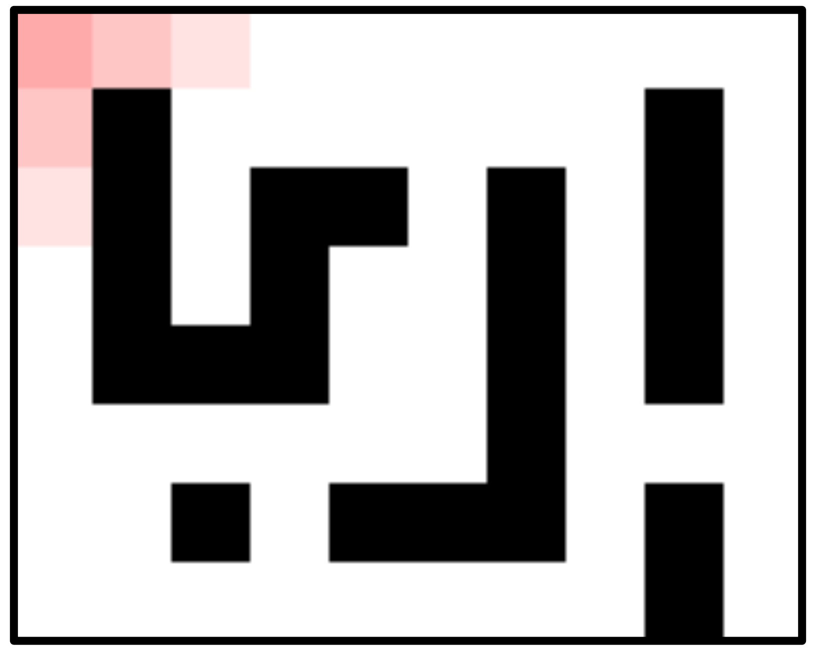
Prediction Step



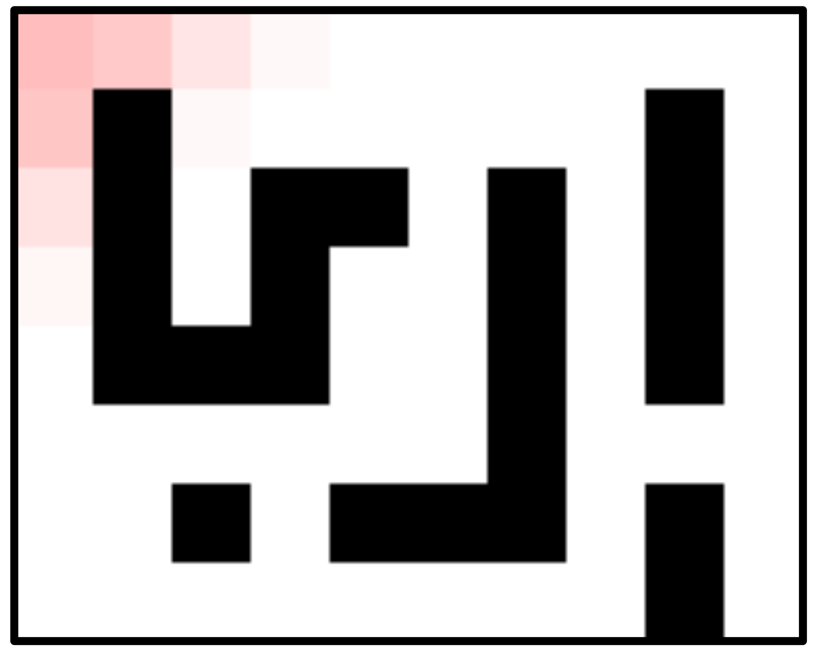
$\overline{bel}_0$



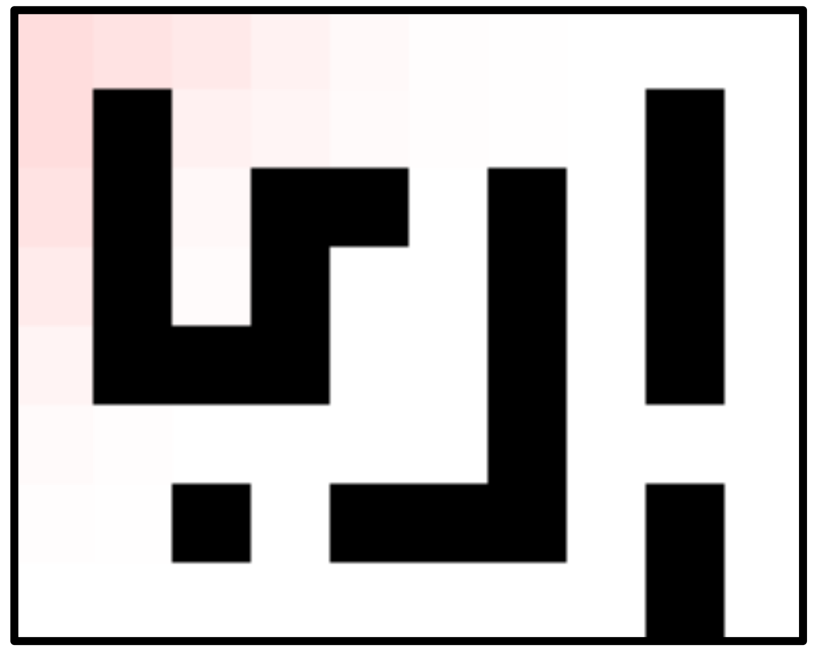
$\overline{bel}_1$



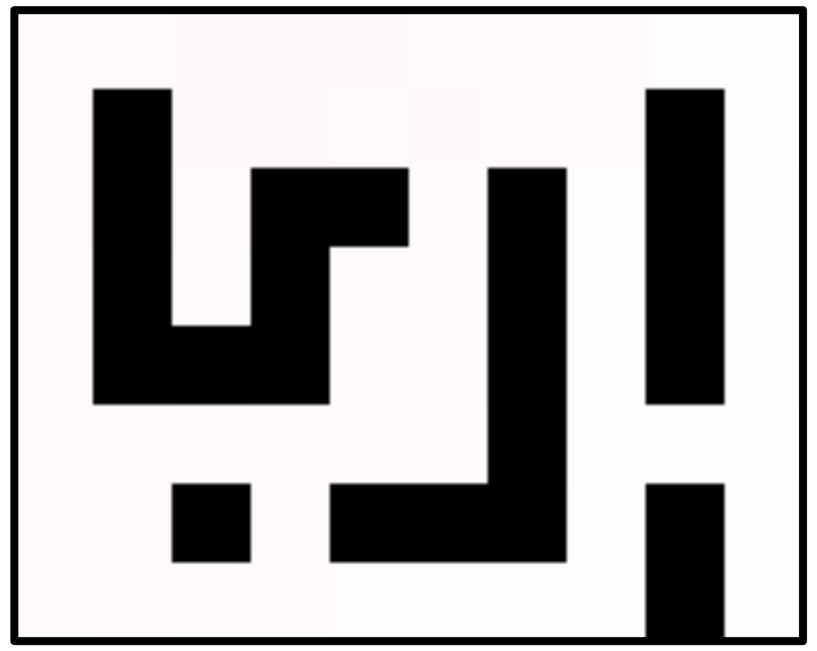
$\overline{bel}_2$



$\overline{bel}_3$



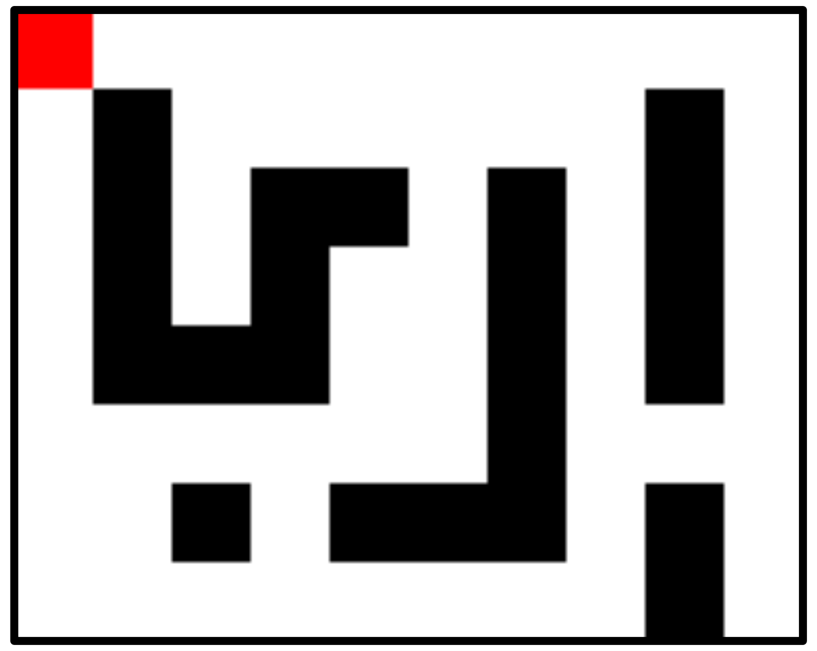
$\ldots\overline{bel}_{10}$



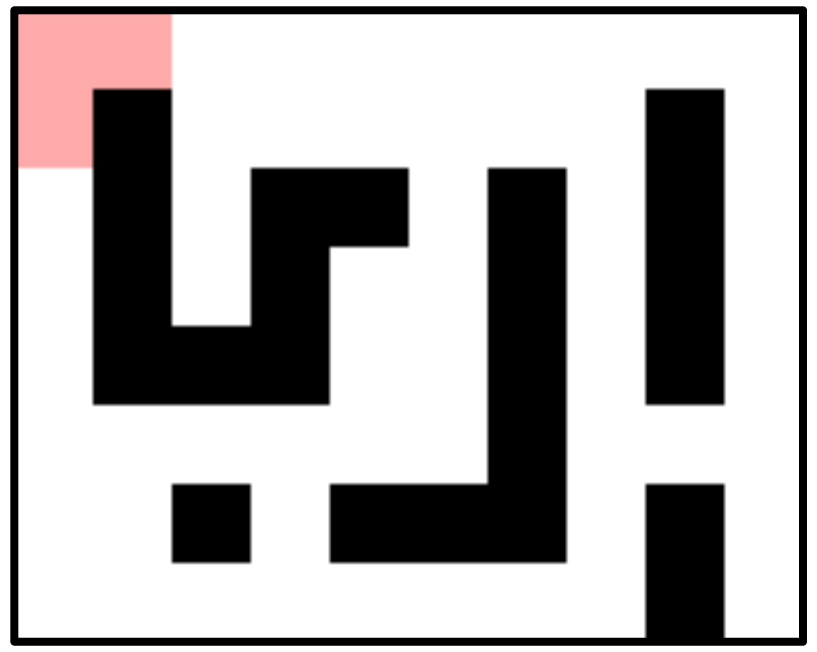
$\ldots\overline{bel}_{100}$

Example 3

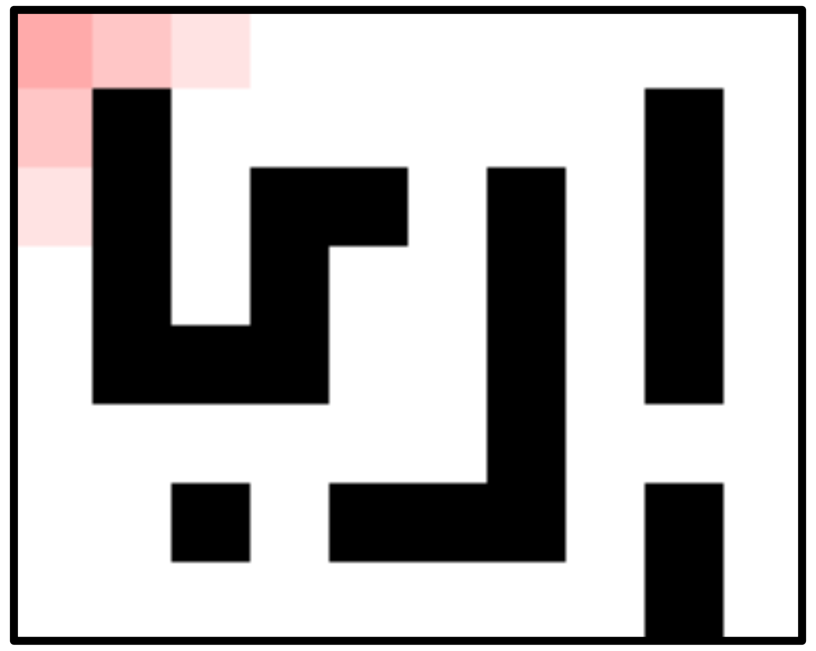
Observations



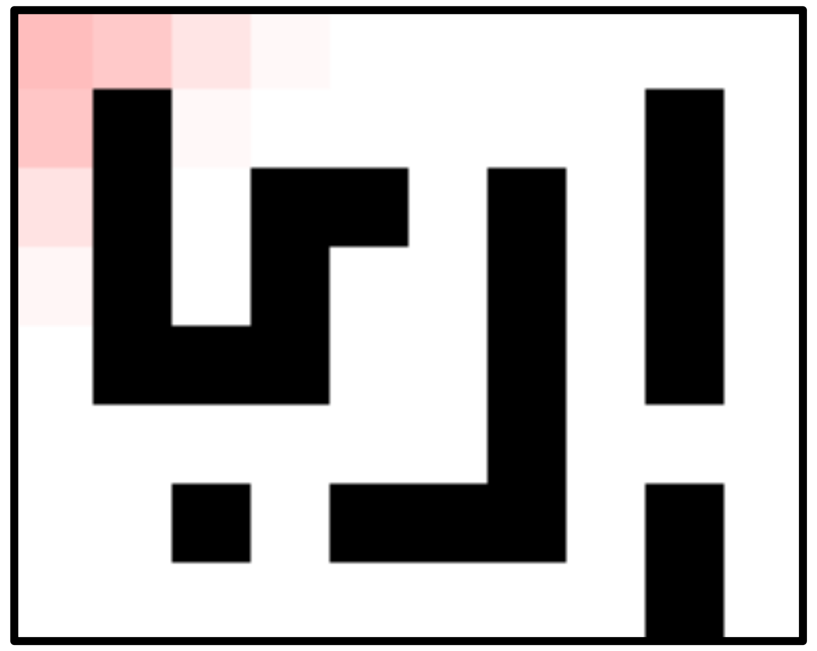
$\overline{bel}_0$



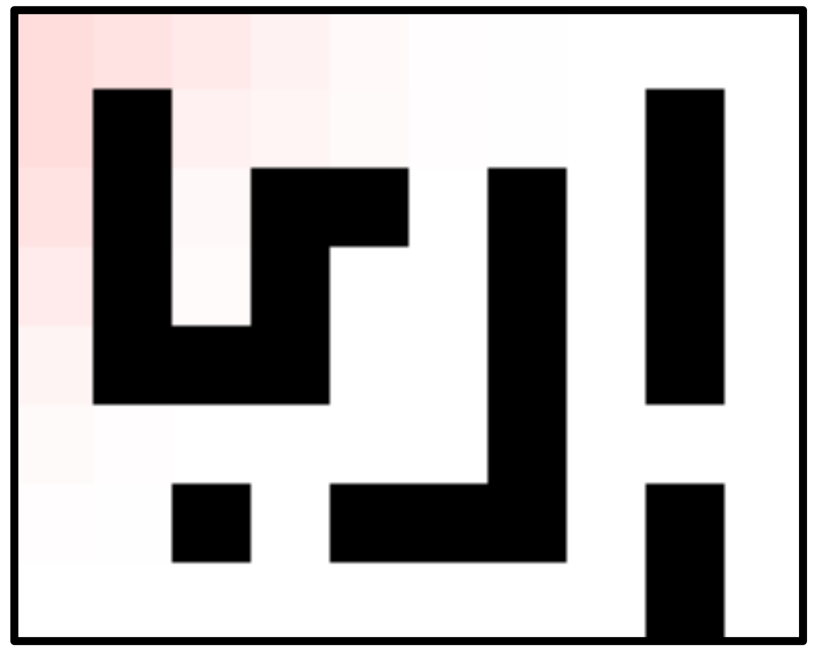
$\overline{bel}_1$



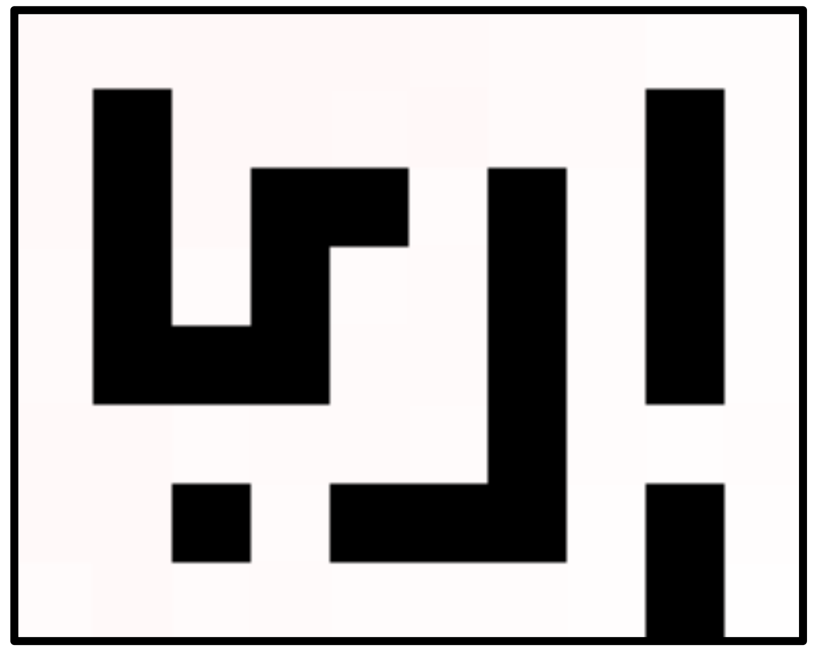
$\overline{bel}_2$



$\overline{bel}_3$

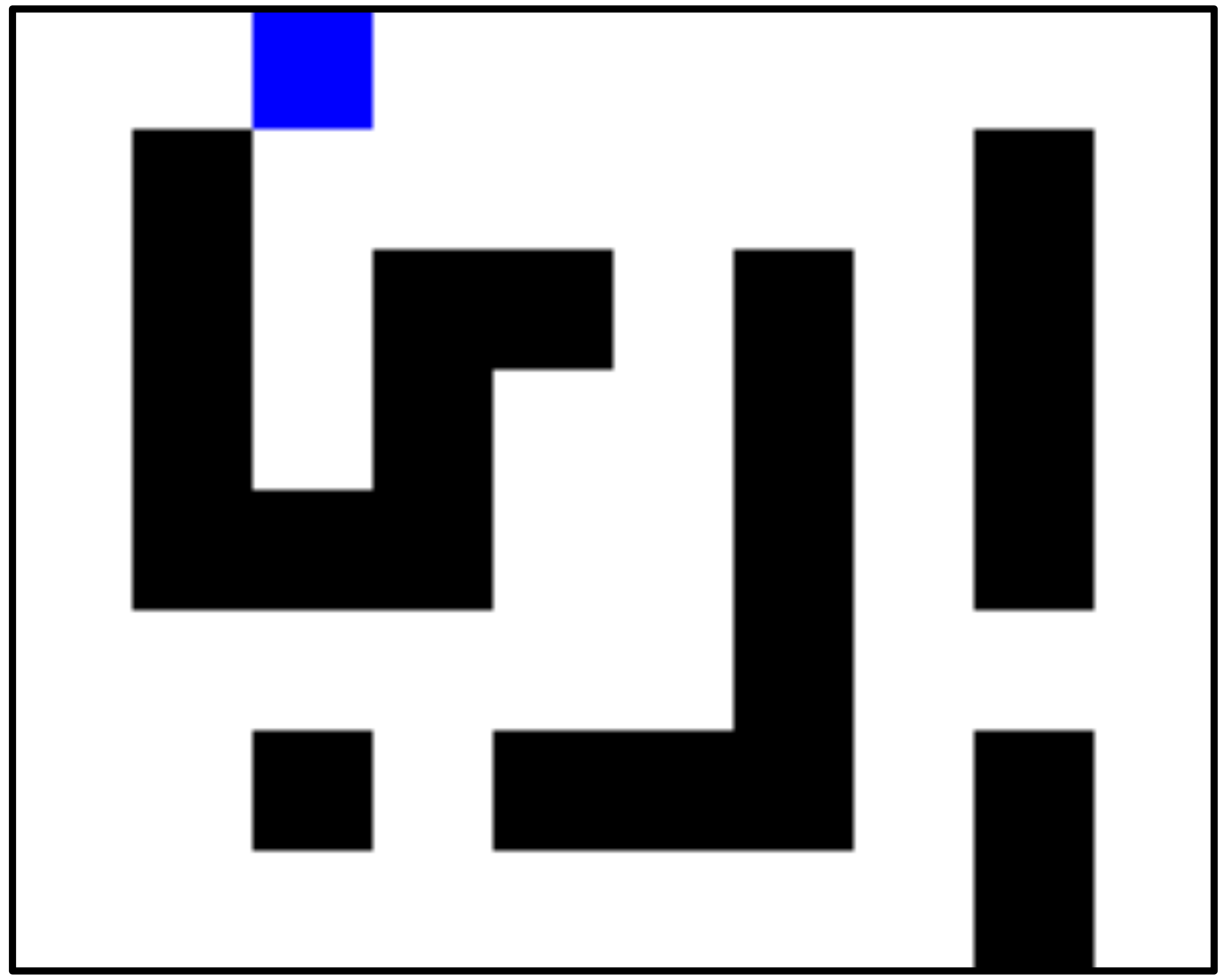


$\ldots\overline{bel}_{10}$



$\ldots\overline{bel}_{100}$

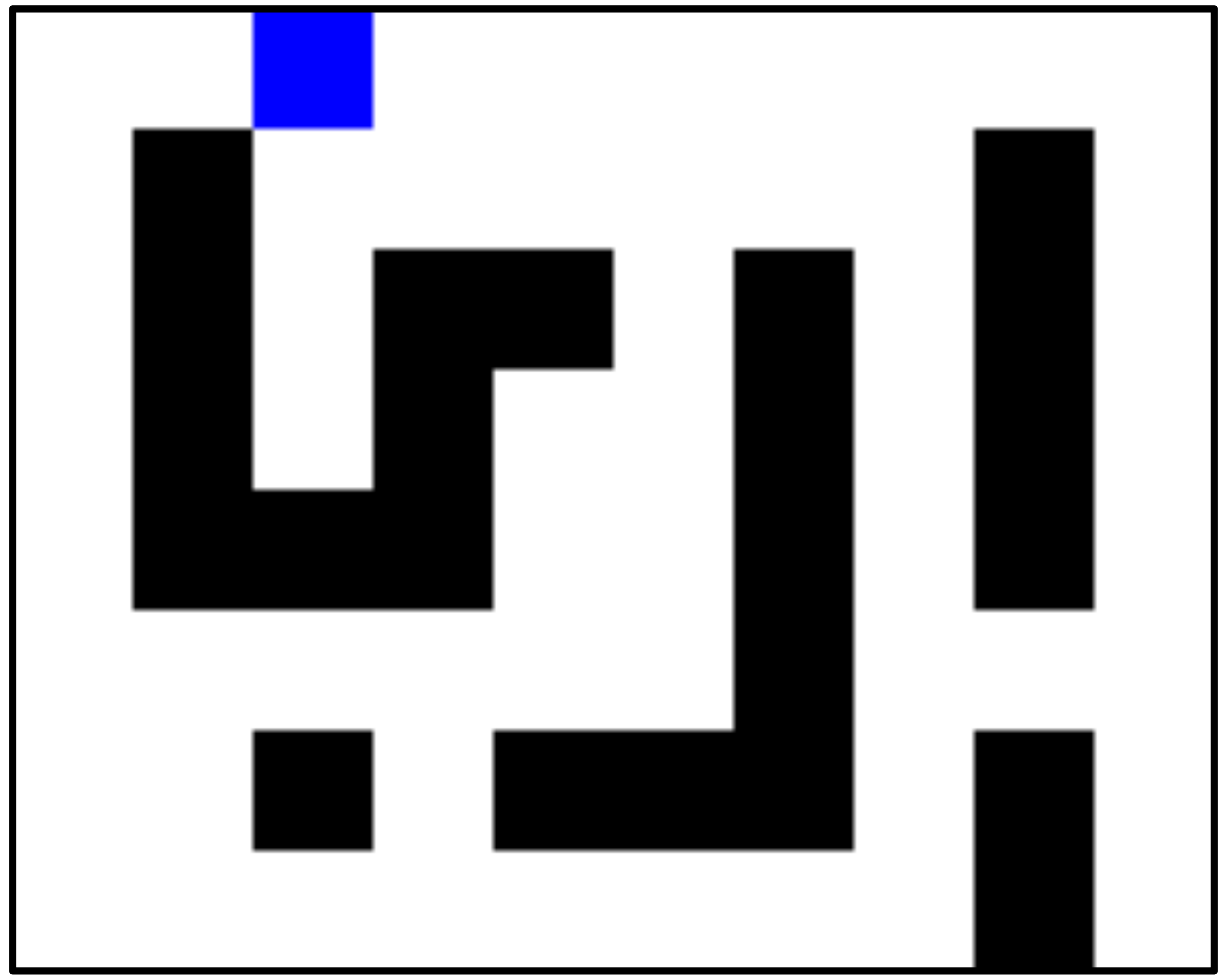
- The robot may not know where it is, but it **does** have a physical state
- It will have observations tied to that state



Example 3

Observation model

- In every time step, we sense each of the four neighboring cells (N, E, S, W)
- In z, each reading is independent and correct with 90% probability



x is the set of possible locations

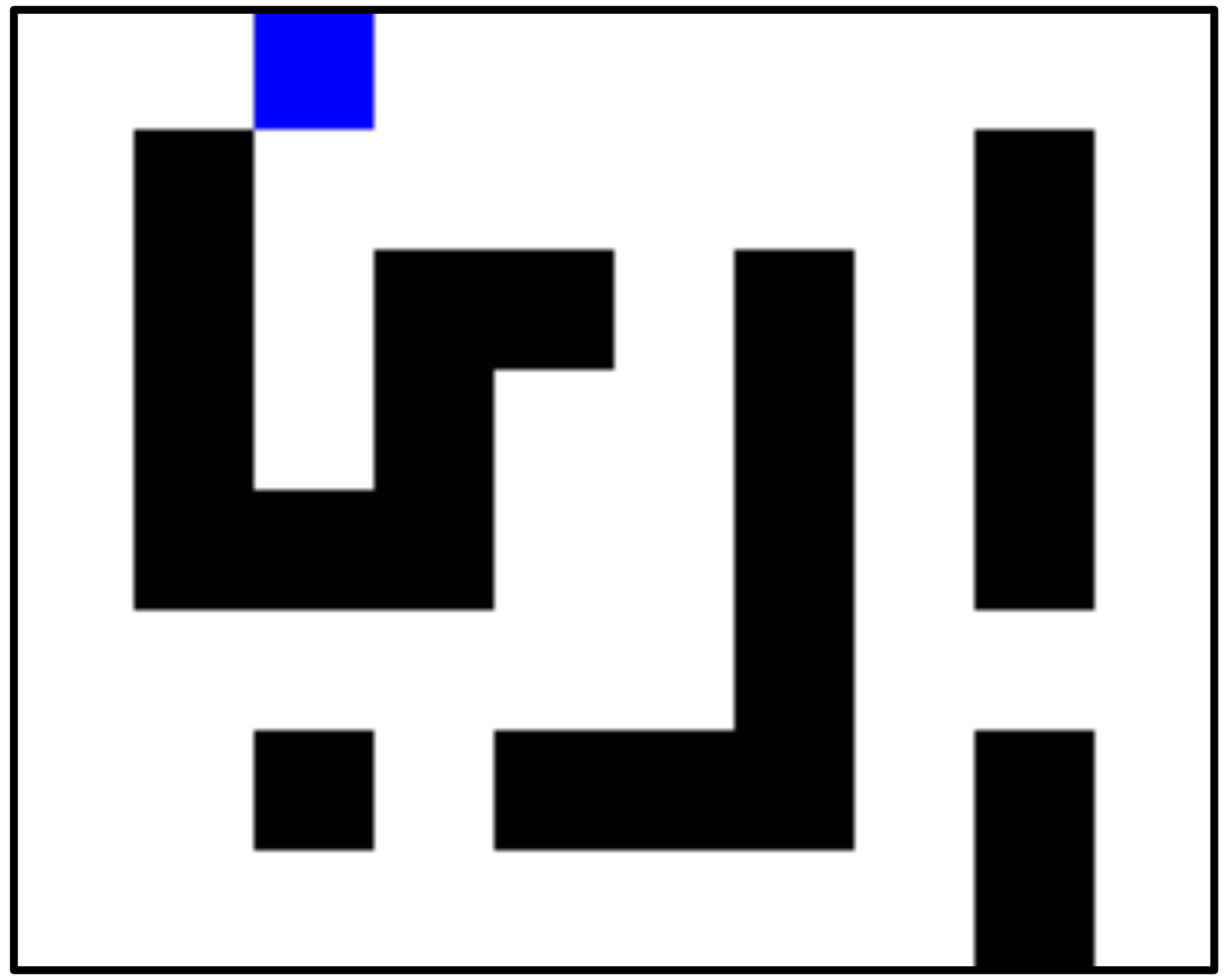
X is one location

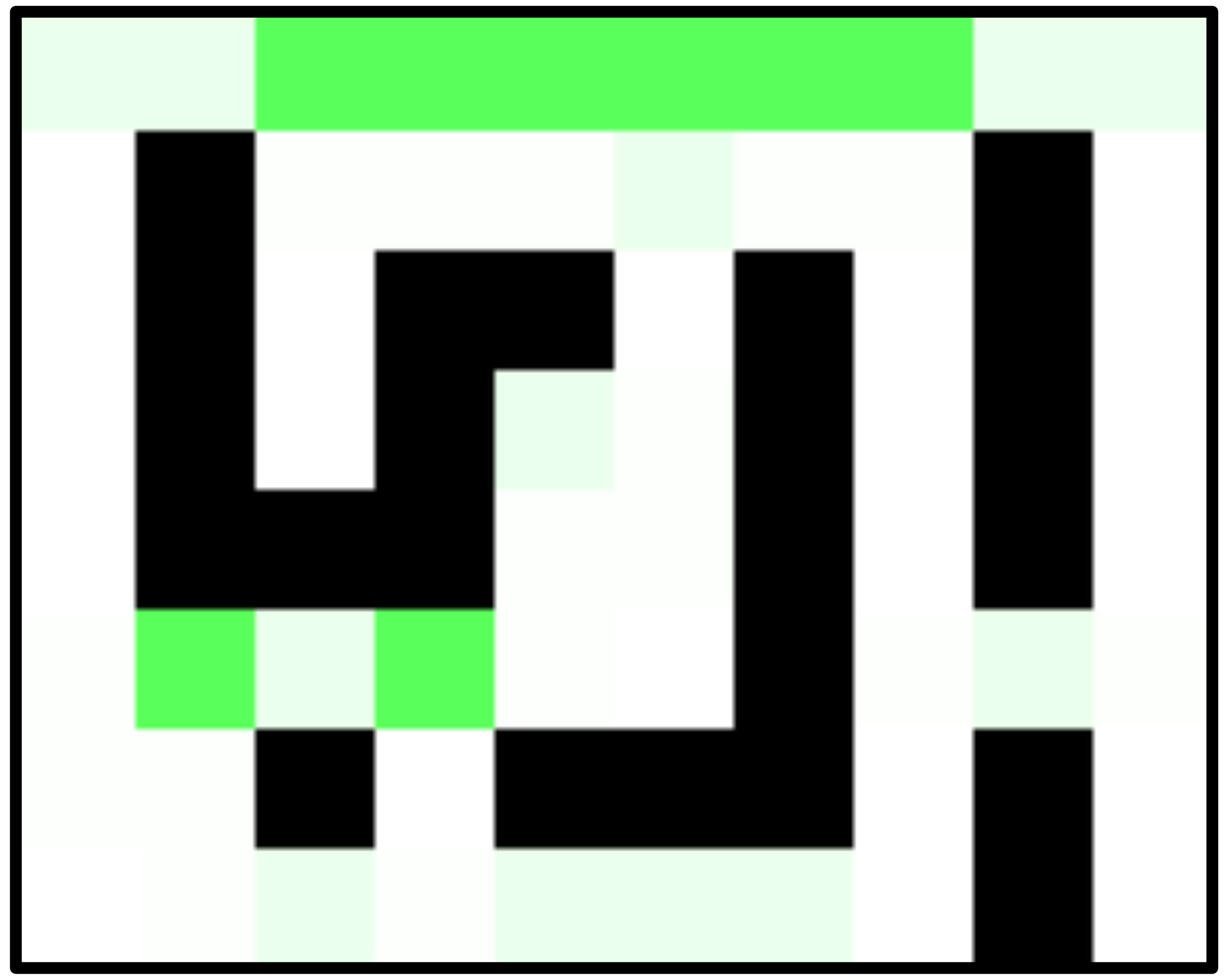
z are the sensor measurements

Example 3

Observation model

- In every time step, we sense each of the four neighboring cells (N, E, S, W)
- In z, each reading is independent and correct with 90% probability





How many combinations are there per state?

Example 3

Observation model

- In every time step, we sense each of the four neighboring cells (N, E, S, W)
- In z, each reading is independent and correct with 90% probability

Algorithm Bayes_Filter $(bel(x_{t-1}), u_t, z_t)$:

  for all x_t do

    $\overline{bel}(x_t) = \Sigma_{x_{t-1}} \, p(x_t \mid u_t, x_{t-1}) \, bel(x_{t-1})$

    $bel(x_t) = \eta \, p(z_t \mid x_t) \, \overline{bel}(x_t)$

  end **for**

return $bel(x_t)$

- If all readings are correct:
  - $\Sigma \mid z_t - z'_{xt} \mid = 0$
  - $p_z(x_t) = 0.6561$
- If all readings are incorrect:
  - $\Sigma \mid z_t - z'_{xt} \mid = 4$
  - $p_z(x_t) = 0.0001$

Likelihood of Observations, p_{zX}:

  for all x_t do

    $p_{zX}(x_t) = 0.9^{4-\Sigma|z_t - z'_{xt}|} \, 0.1^{\Sigma|z_t - z'_{xt}|}$

  end **for**

where p_{zX} is a vector (80x1)

Example 3

Observation model

- In every time step, we sense each of the four neighboring cells (N, E, S, W)
- In z, each reading is independent and correct with 90% probability

Algorithm Bayes_Filter $(bel(x_{t-1}), u_t, z_t)$:

for all x_t do

$\overline{bel}(x_t) = \Sigma_{x_{t-1}} \, p(x_t \mid u_t, x_{t-1}) \, bel(x_{t-1})$

$bel(x_t) = \eta \, p(z_t \mid x_t) \, \overline{bel}(x_t)$

end **for**

return $bel(x_t)$

Compute new belief:

$$bel_t = \frac{p_{zX}\overline{bel}}{\Sigma(p_{zX}\overline{bel})}$$

where $\overline{bel}$ is a vector (80x1)
and p_{zX} is a vector (80x1)

Example 3

Algorithm Bayes_Filter (bel_{t-1}, z_t) :

$\overline{bel} = A \; bel_{t-1}$

for all x_t do

$p_{zX}(x_t) = 0.9^{4-\Sigma|z_t - z'_{xt}|} \; 0.1^{\Sigma|z_t - z'_{xt}|}$

end **for**

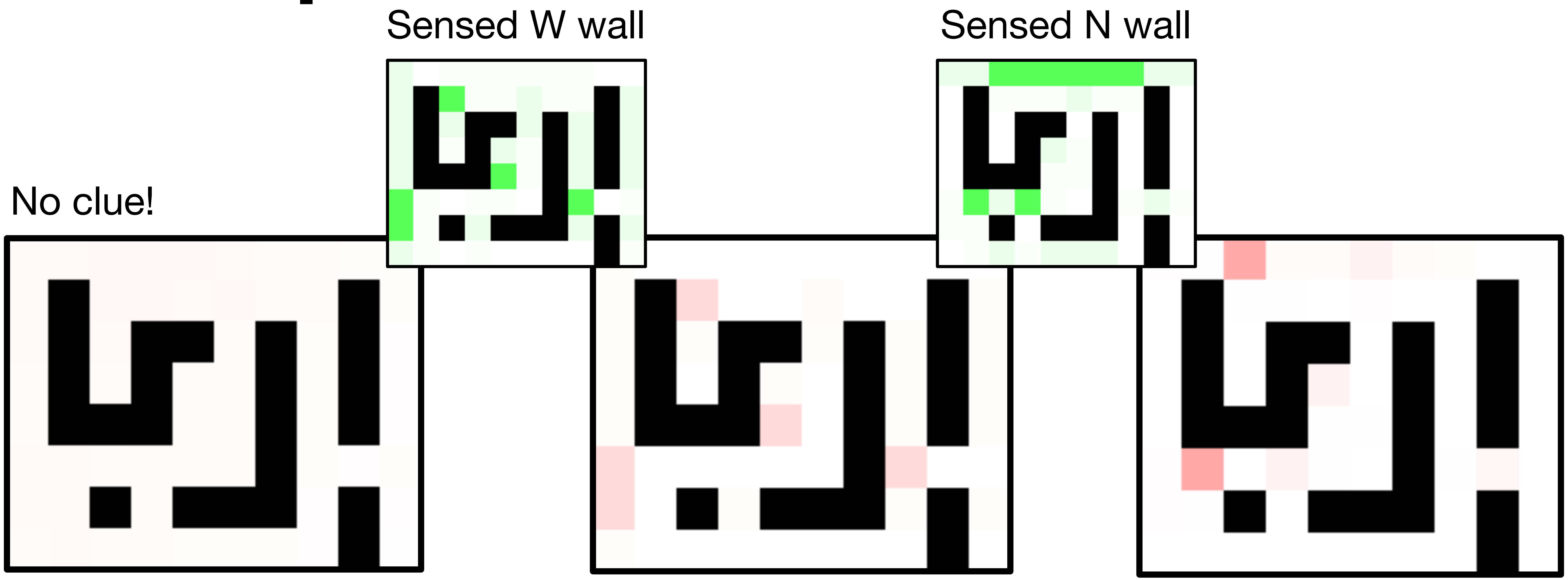
$bel_t = \dfrac{p_{zX}\overline{bel}}{\Sigma(p_{zX}\overline{bel})}$

return $bel(x_t)$

Only do this for states with a belief > threshold

Cache and look up

Example 3



In two steps, we homed in on where we are!

Can we do better?

- Improved transition model
- Deliberately move in directions that give you more information

Today's examples

- Example 1: robot in the 1D world
  - Important to have some belief in all states
- Example 2: Bayes with beans
  - Important to normalize
- Example 3: (x,y) robot in a grid world
  - Important to improve computational efficiency
    - Matrices
    - Pre-cache

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. **for** all x_t do
3. $\overline{bel}(x_t) = \Sigma_{x_{t-1}} \, p(x_t \mid u_t, x_{t-1}) \, bel(x_{t-1})$
4. $bel(x_t) = \eta \, p(z_t \mid x_t) \, \overline{bel}(x_t)$
5. end **for**
6. **return** $bel(x_t)$

Summary of Bayes Filter

- Use temporal consistency between observations that are poor estimates
- Localization can work with…
  - completely random motion
  - noisy sensors
  - Remember to…
    - remain probabilistic
    - normalize
    - improve efficiency

Algorithm Bayes_Filter $(bel(x_{t-1}), u_t, z_t)$:

    for all x_t do

        $\overline{bel}(x_t) = \Sigma_{x_{t-1}}\, p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})$

        $bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$

    end **for**

return $bel(x_t)$