

# CS 221: Artificial Intelligence

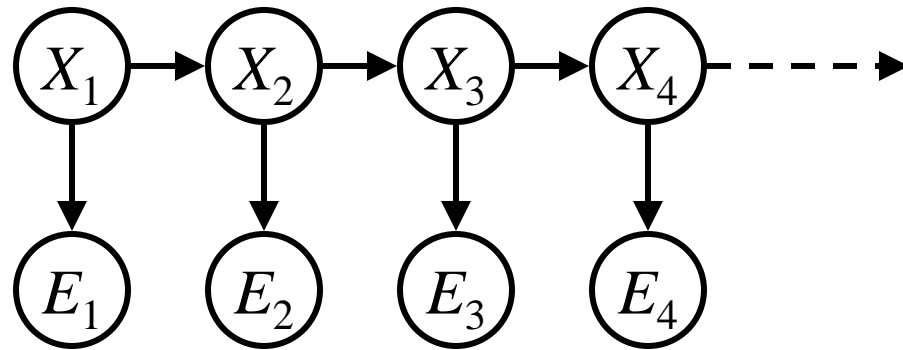
## Lecture 5: Hidden Markov Models and Temporal Filtering

Sebastian Thrun and Peter Norvig

Slide credit: Dan Klein, Michael Pfeiffer

# Class-On-A-Slide

---



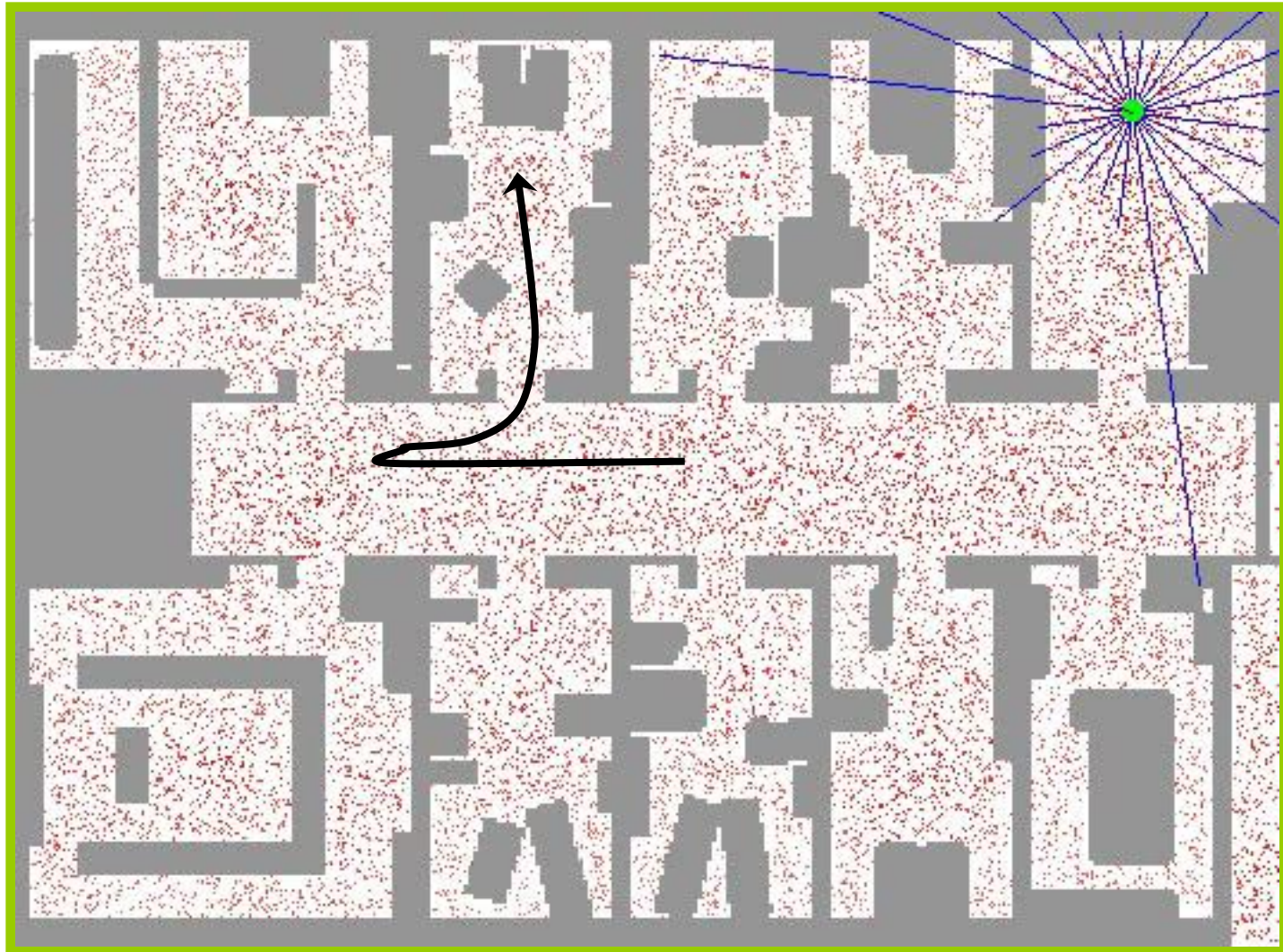
# Example: Minerva

---



# Example: Robot Localization

---



# Example: Groundhog

---



# Example: Groundhog

---



# Example: Groundhog

---







# Overview

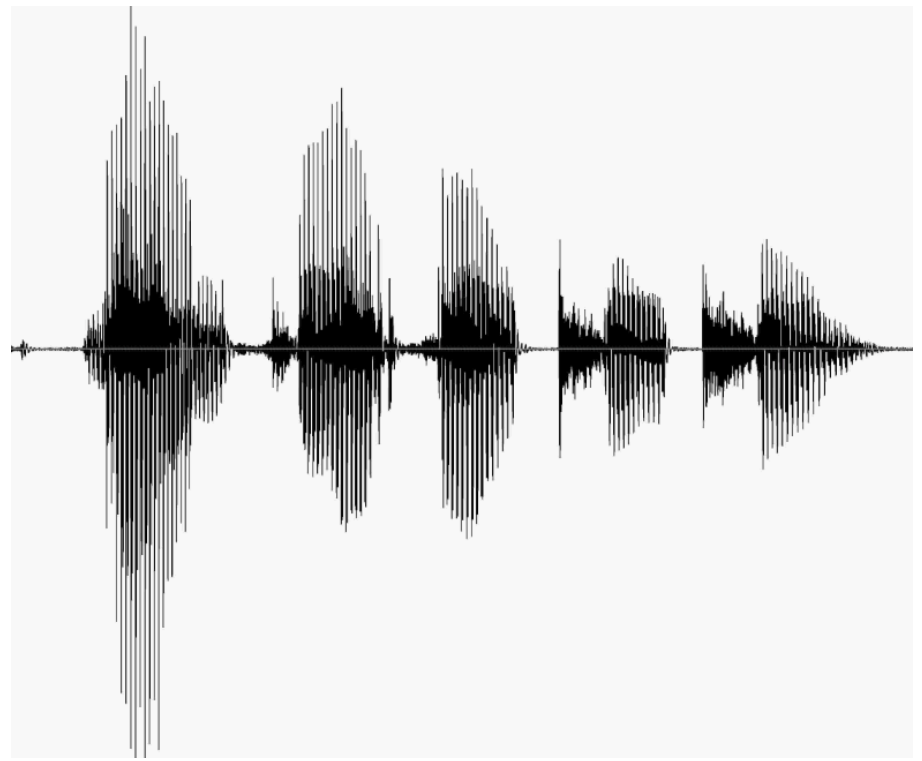
---

- Markov Chains
- Hidden Markov Models
- Particle Filters
- More on HMMs

# Reasoning over Time

---

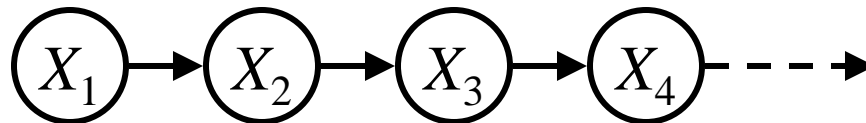
- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring
  - Financial modeling



# Markov Models

---

- A **Markov model** is a chain-structured BN
  - Each node is identically distributed (stationarity)
  - Value of  $X$  at a given time is called the **state**
  - As a BN:

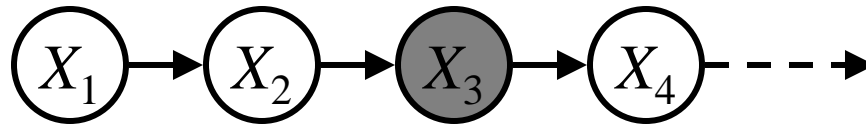


$$P(X_1) \quad P(X|X_{-1})$$

- Parameters: called **transition probabilities** or dynamics, specify how the state evolves over time (also, initial probs)

# Conditional Independence

---

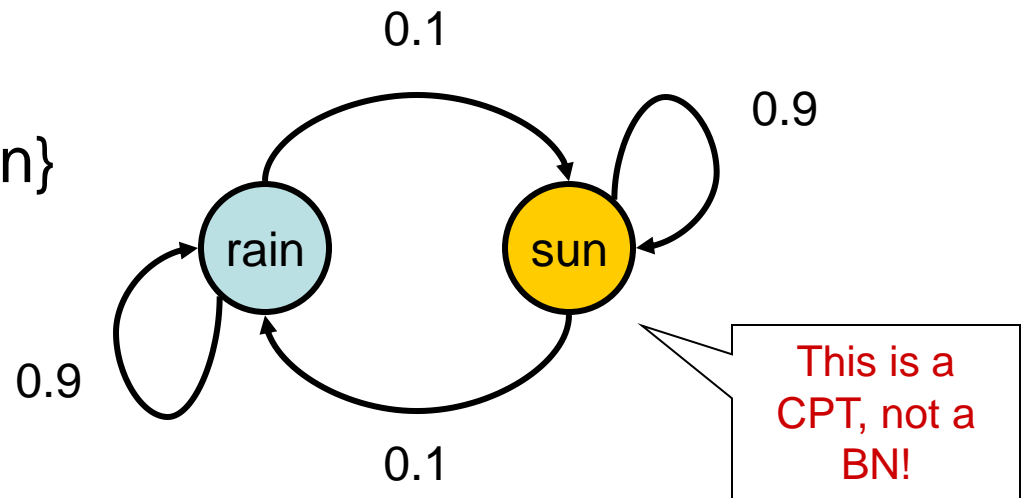


- **Basic conditional independence:**
  - Past and future independent of the present
  - Each time step only depends on the previous
  - This is called the Markov property
- **Note that the chain is just a (growing) BN**
  - We can always use generic BN reasoning on it if we truncate the chain at a fixed length

# Example: Markov Chain

- Weather:

- States:  $X = \{\text{rain}, \text{sun}\}$
- Transitions:

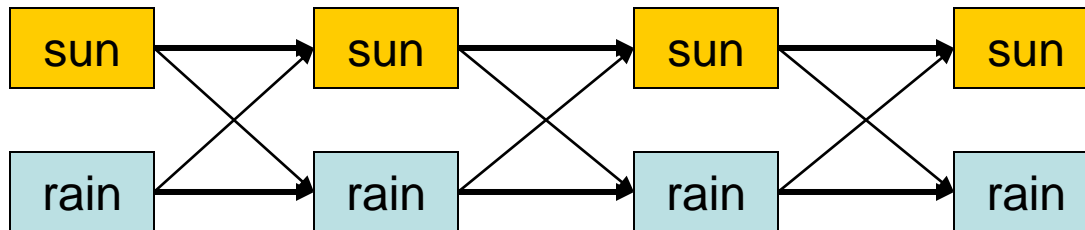


- Initial distribution: 1.0 sun
- What's the probability distribution after one step?

$$\begin{aligned} P(X_2 = \text{sun}) &= P(X_2 = \text{sun} | X_1 = \text{sun})P(X_1 = \text{sun}) + \\ &\quad P(X_2 = \text{sun} | X_1 = \text{rain})P(X_1 = \text{rain}) \\ &= 0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9 \end{aligned}$$

# Mini-Forward Algorithm

- Question: What's  $P(X)$  on some day  $t$ ?
  - An instance of variable elimination!



$$P(x_t) = \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1})$$

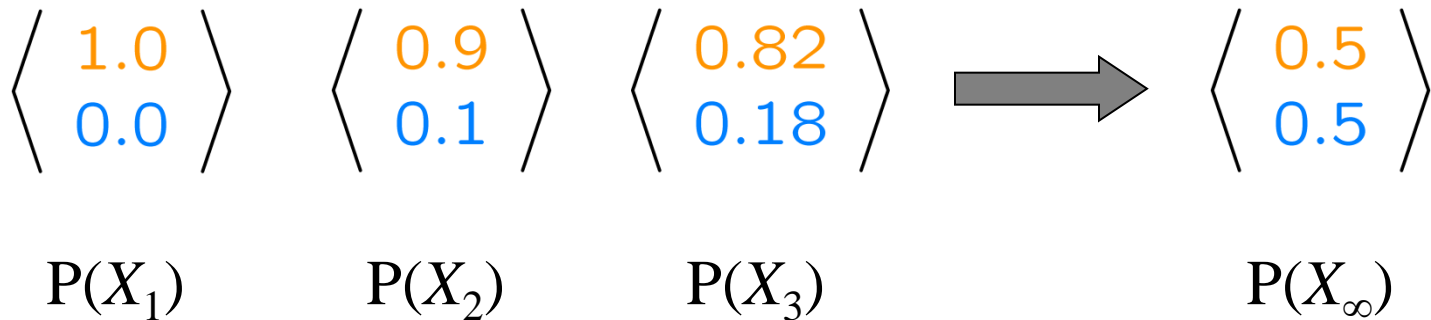
$$P(x_1) = \text{known}$$

*Forward simulation*

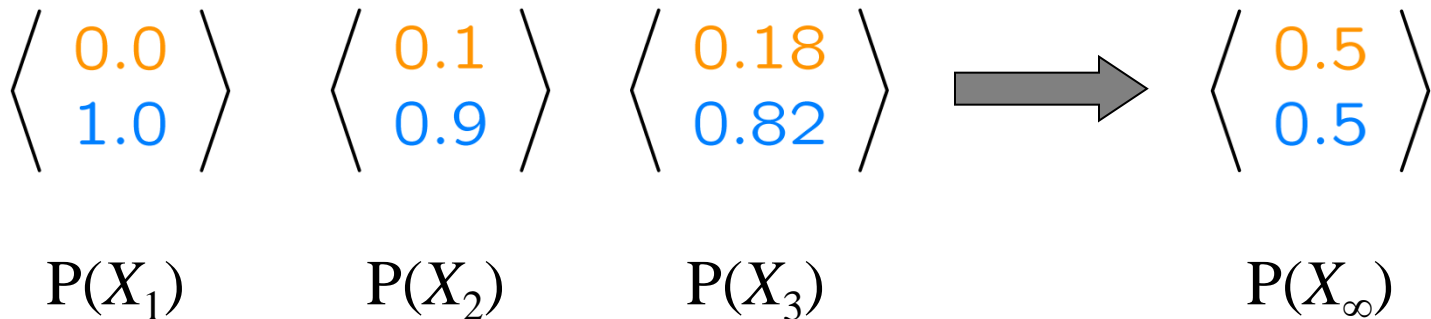
# Example

---

- From initial observation of sun



- From initial observation of rain



# Stationary Distributions

---

- If we simulate the chain long enough:
  - What happens?
  - Uncertainty accumulates
  - Eventually, we have no idea what the state is!
- Stationary distributions:
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the **stationary distribution** of the chain
  - Usually, can only predict a short time out

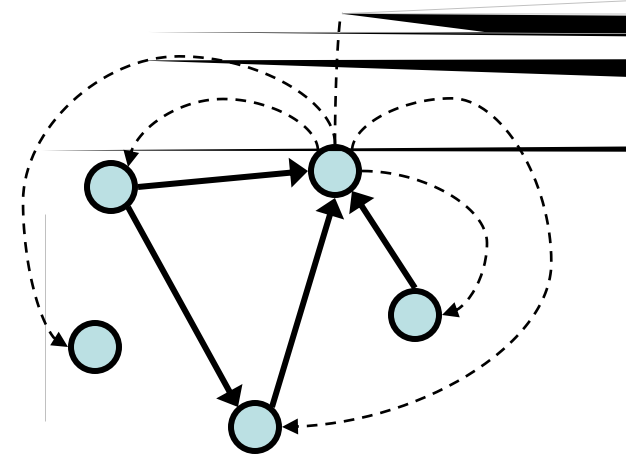


# Example: Web Link Analysis

---

- PageRank over a web graph

- Each web page is a state
- Initial distribution: uniform over pages
- Transitions:
  - With prob.  $c$ , uniform jump to a random page (dotted lines, not all shown)
  - With prob.  $1-c$ , follow a random outlink (solid lines)



- Stationary distribution

- Will spend more time on highly reachable pages
- Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors (rank actually getting less important over time)

# Overview

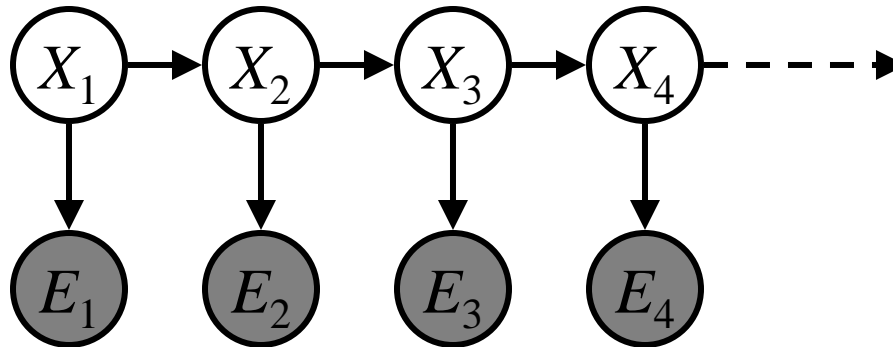
---

- Markov Chains
- Hidden Markov Models
- Particle Filters
- More on HMMs

# Hidden Markov Models

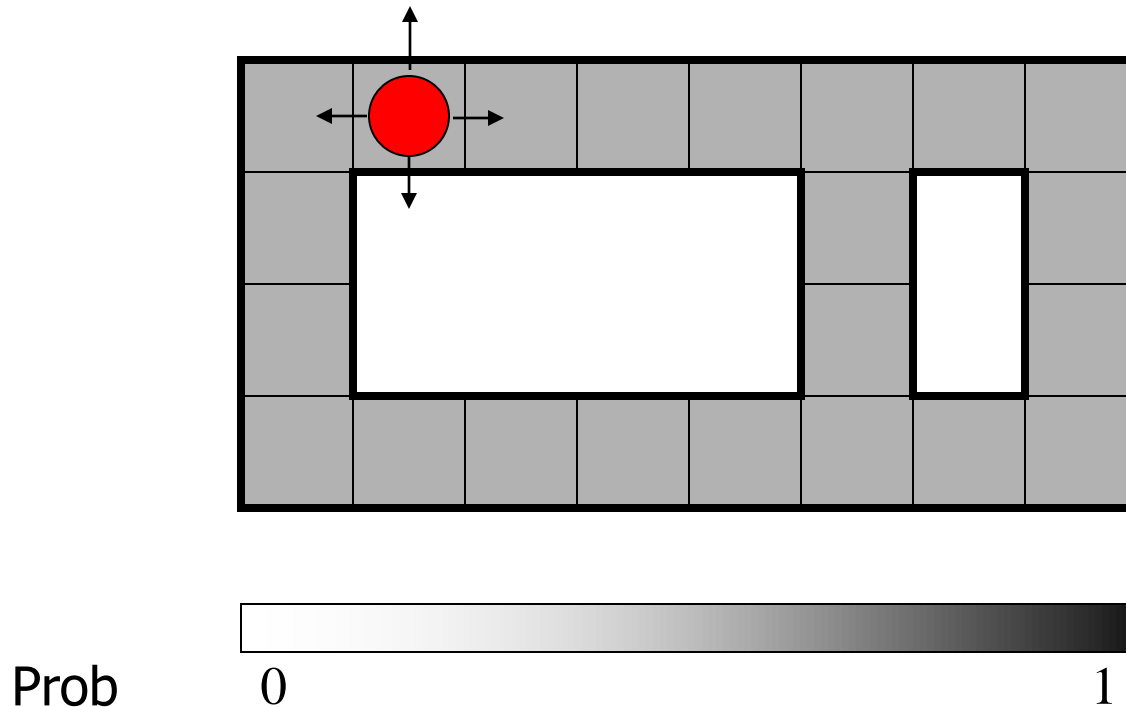
---

- Markov chains not so useful for most agents
  - Eventually you don't know anything anymore
  - Need observations to update your beliefs
- Hidden Markov models (HMMs)
  - Underlying Markov chain over states  $S$
  - You observe outputs (effects) at each time step
  - As a Bayes net:



# Example: Robot Localization

*Example from  
Michael Pfeiffer*



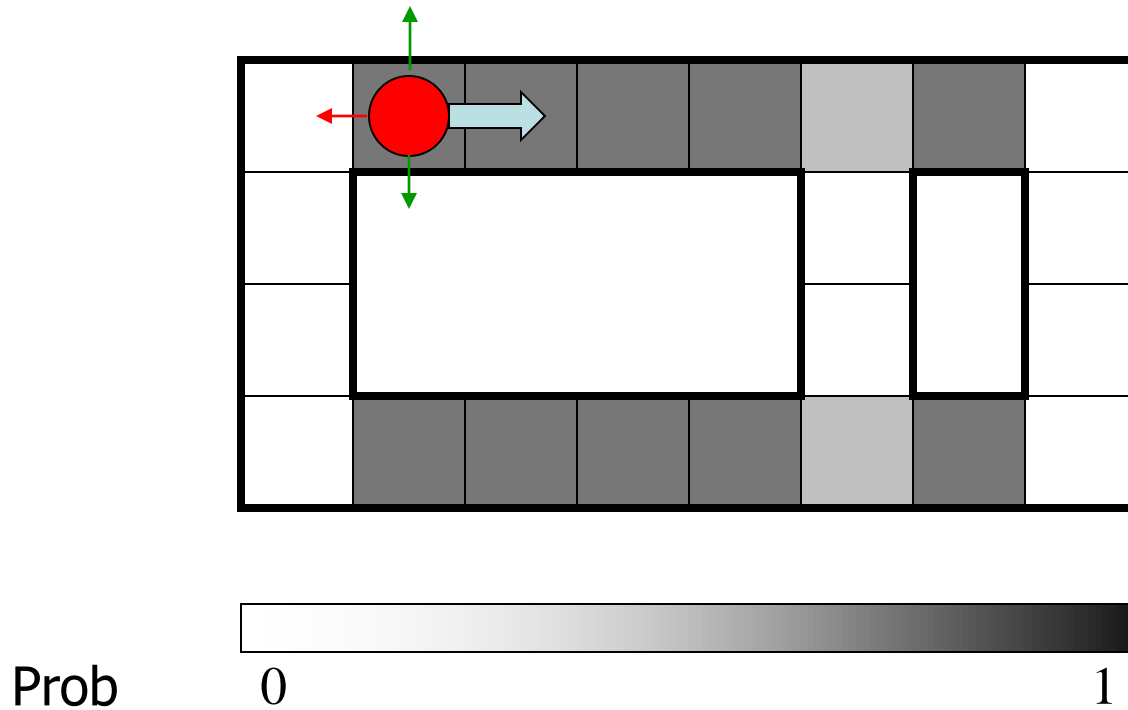
$t=0$

Sensor model: never more than 1 mistake

Motion model: may not execute action with small prob.

# Example: Robot Localization

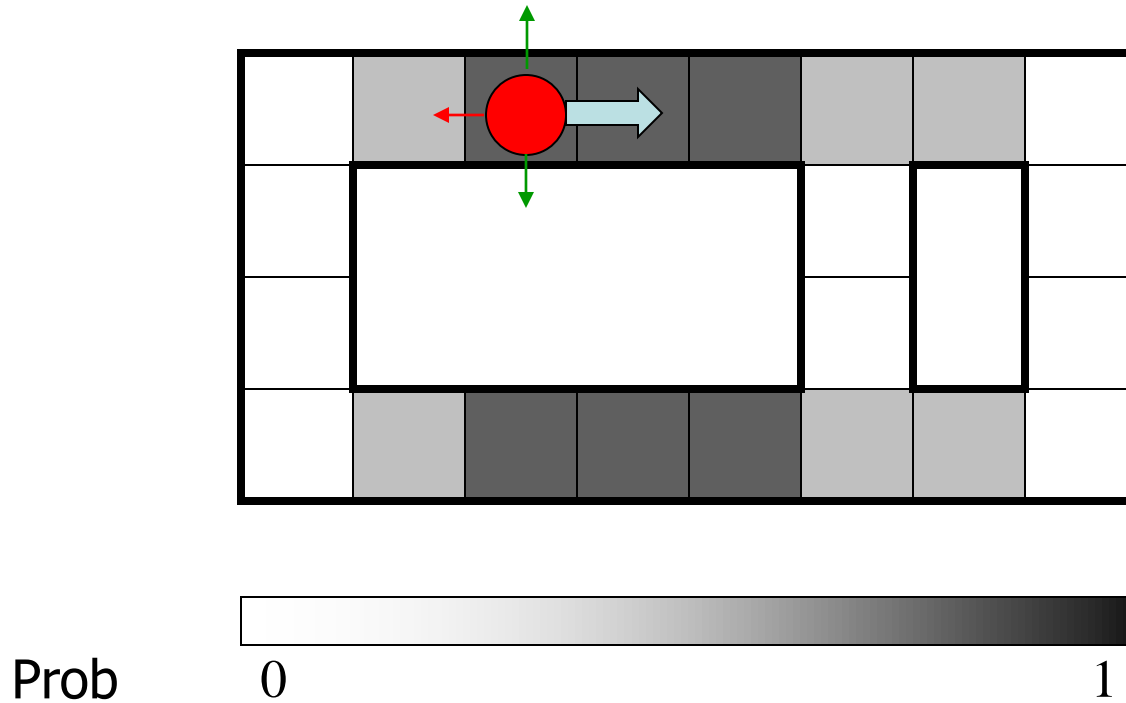
---



t=1

# Example: Robot Localization

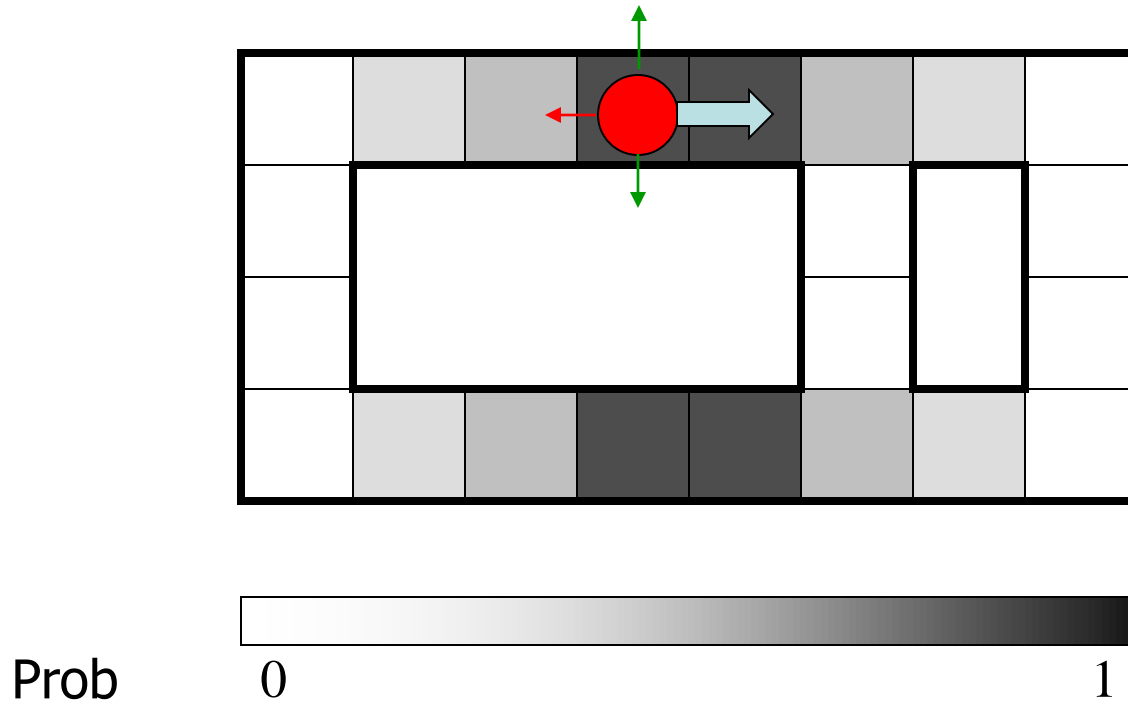
---



t=2

# Example: Robot Localization

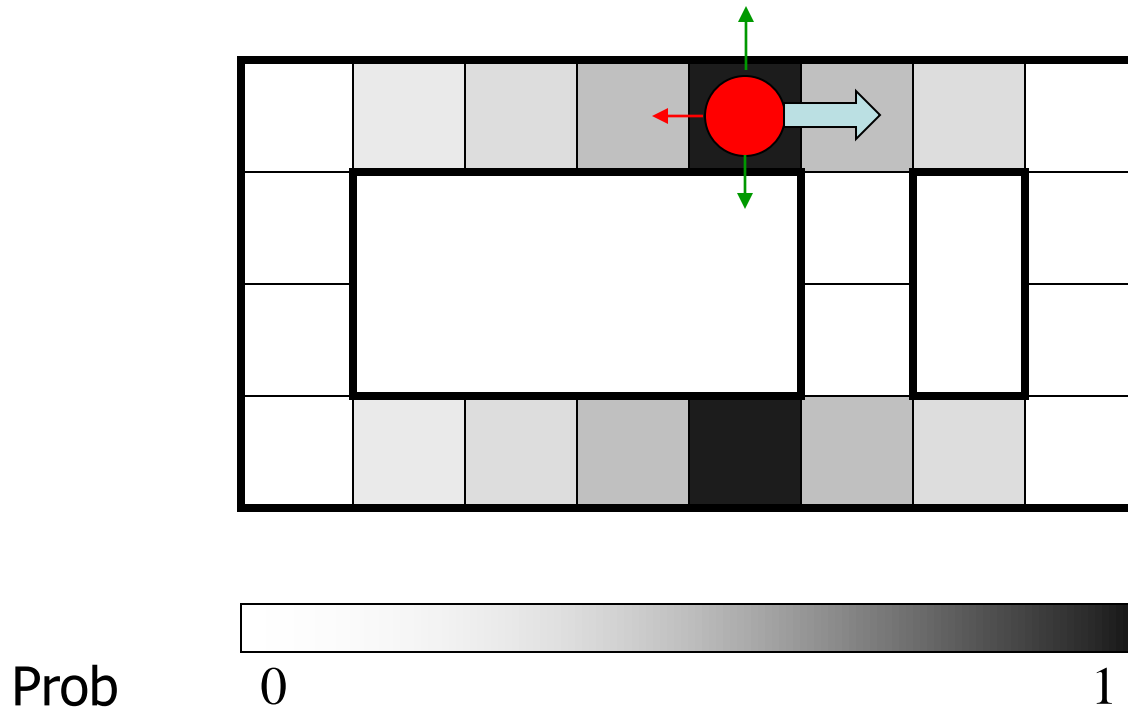
---



$t=3$

# Example: Robot Localization

---

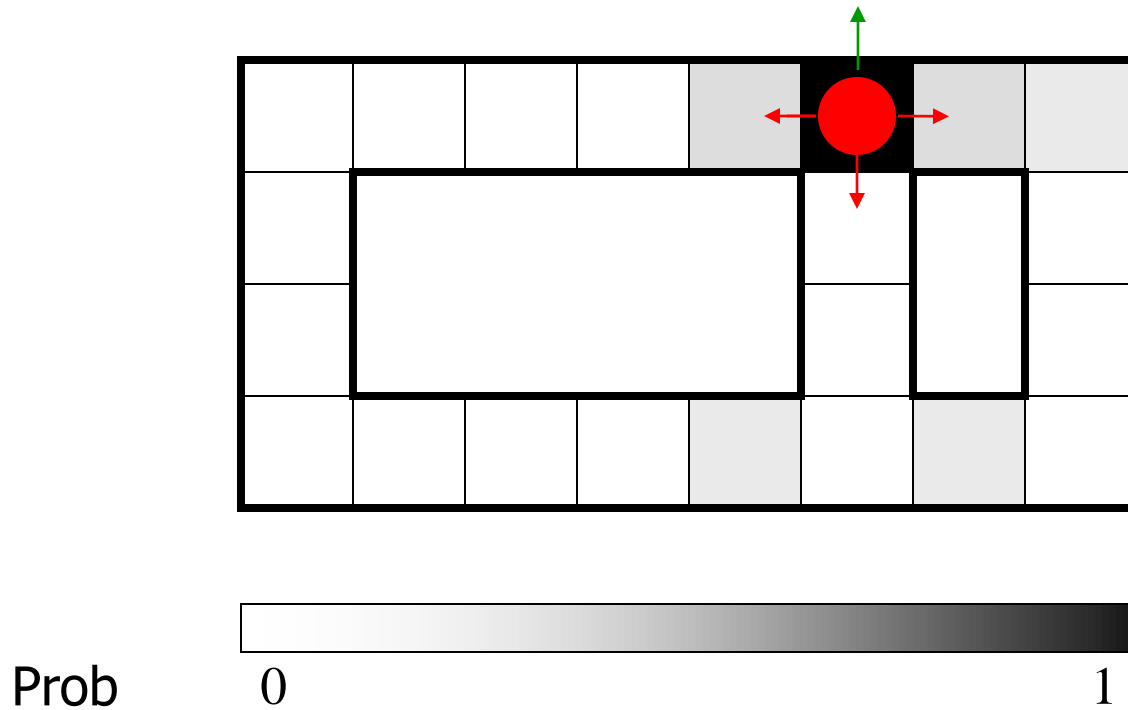


$t=4$



# Example: Robot Localization

---

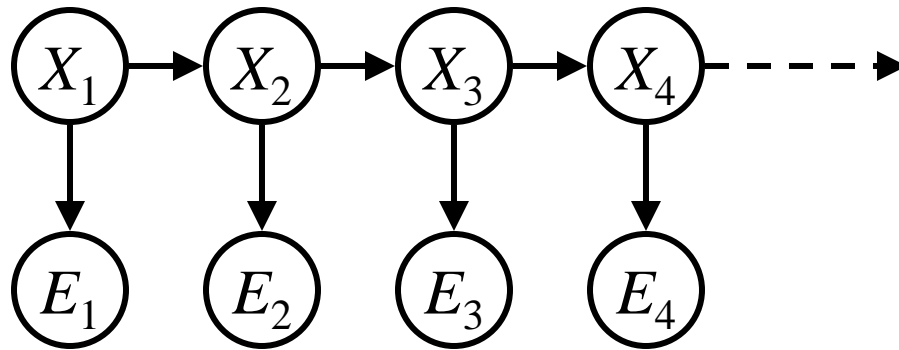


$t=5$

# Hidden Markov Model

---

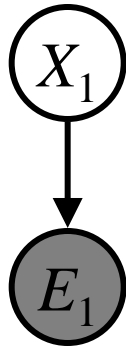
- HMMs have two important independence properties:
  - Markov hidden process, future depends on past via the present
  - Current observation independent of all else given current state



- Quiz: does this mean that observations are mutually independent?
  - [No, correlated by the hidden state]

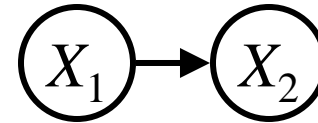
# Inference in HMMs (Filtering)

---



$$P(X_1 | e_1)$$

$$P(x_1 | e_1) = \frac{P(e_1 | x_1)P(x_1)}{\sum P(e_1 | x_1)P(x_1)}$$

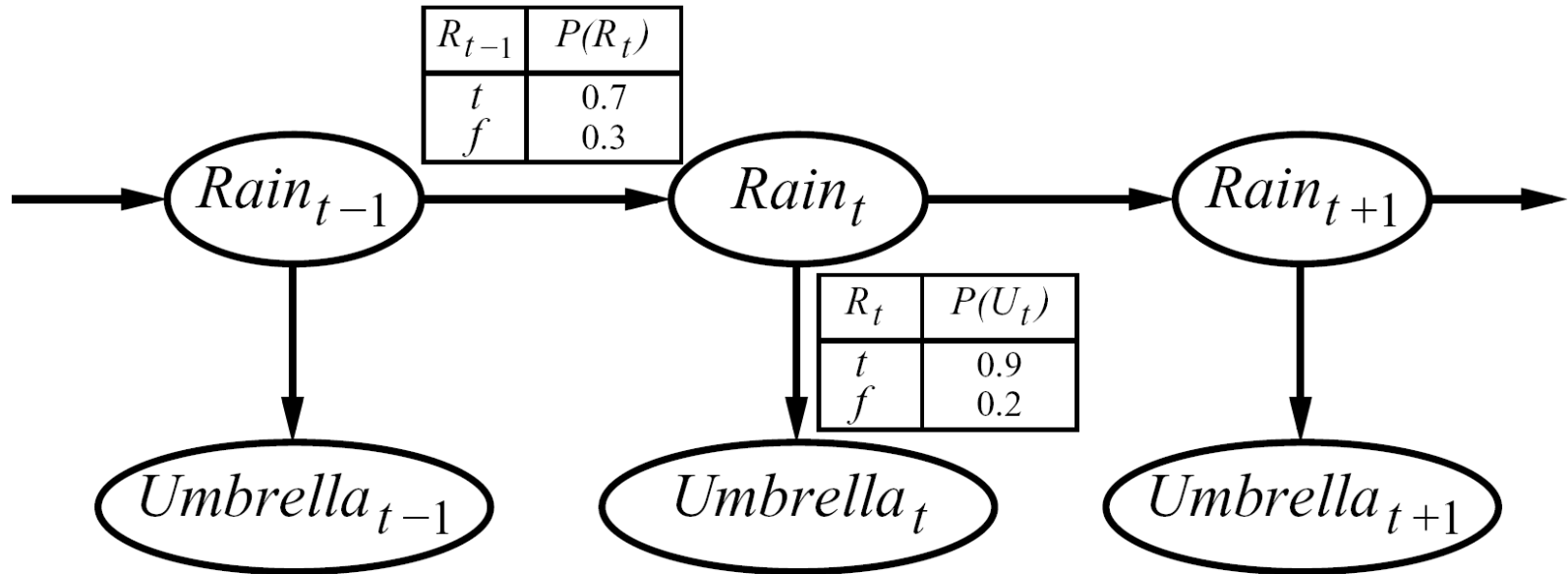


$$P(X_2)$$

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1)P(x_2 | x_1) \end{aligned}$$

$$Bel(x_t) = \mu P(e_t | x_t) \mathring{a}_{x_{t-1}} P(x_t | u_t, x_{t-1}) Bel(x_{t-1})$$

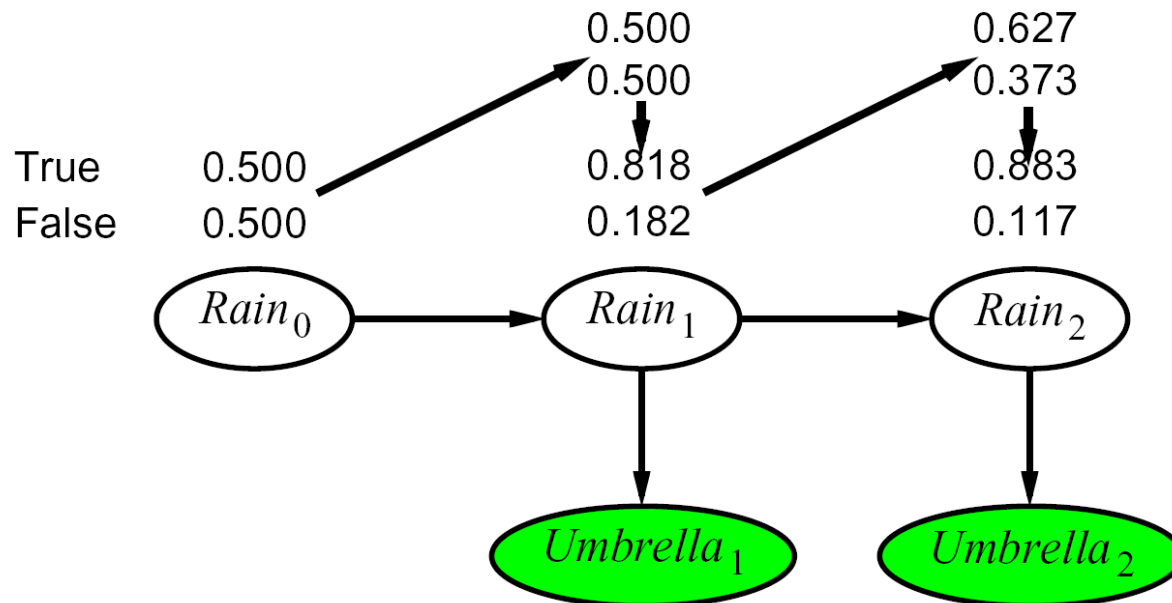
# Example



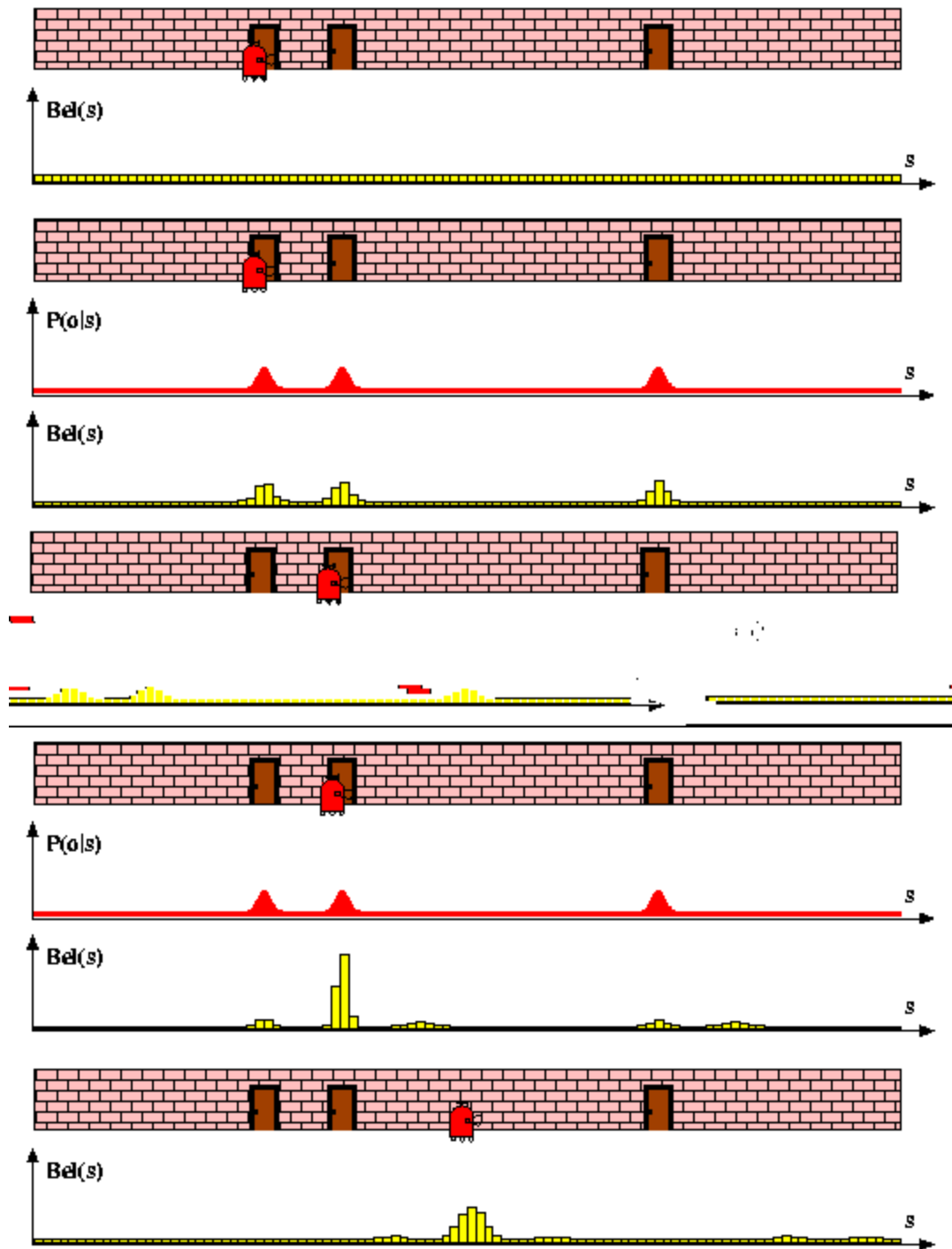
- An HMM is defined by:
  - Initial distribution:  $P(X_1)$
  - Transitions:  $P(X|X_{-1})$
  - Emissions:  $P(E|X)$

# Example HMM

---



# Example: HMMs in Robotics



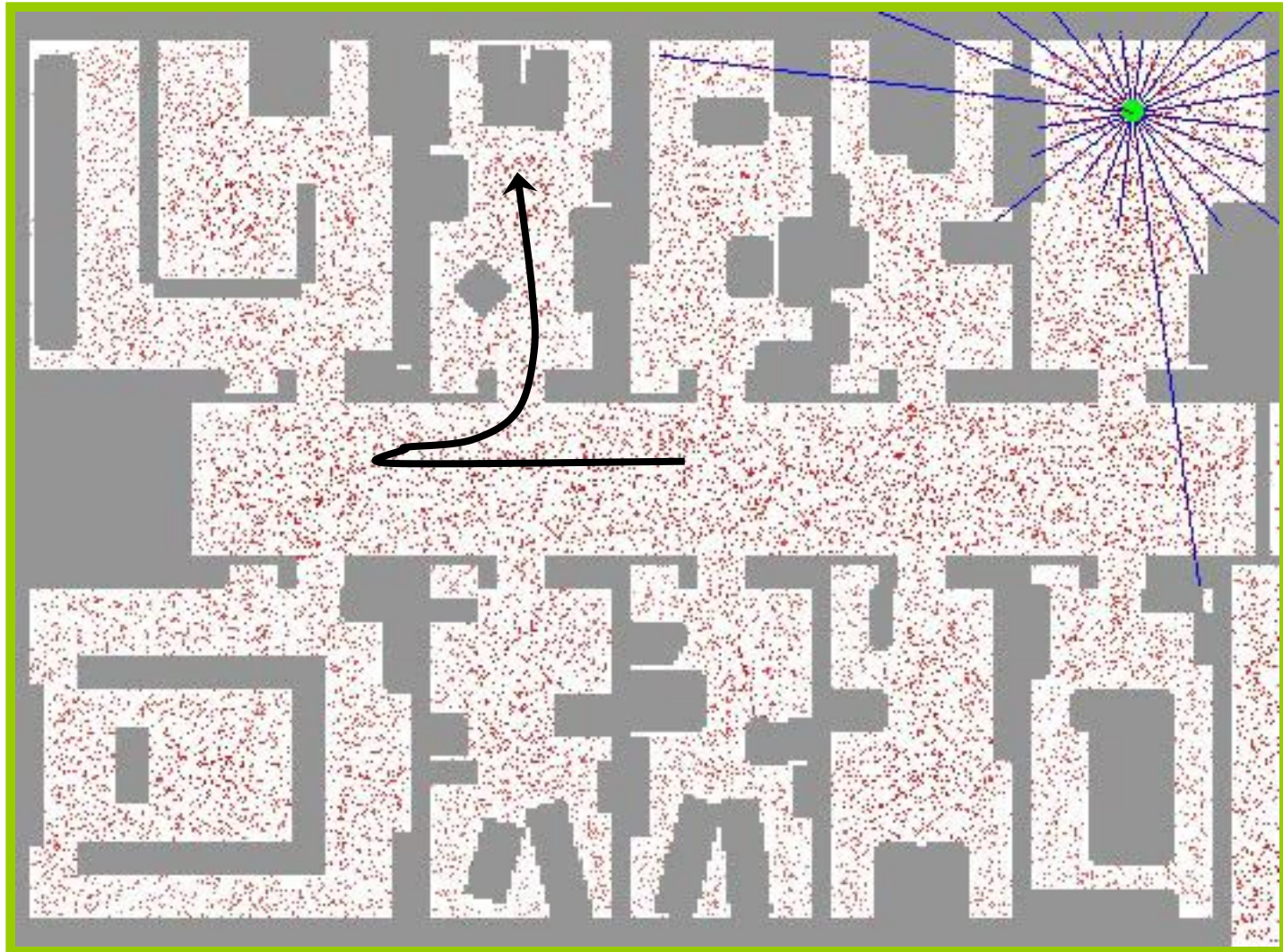
# Overview

---

- Markov Chains
- Hidden Markov Models
- Particle Filters
- More on HMMs

# Example: Robot Localization

---





# Particle Filtering

- Sometimes  $|X|$  is too big to use exact inference
  - $|X|$  may be too big to even store  $B(X)$
  - E.g.  $X$  is continuous
  - $|X|^2$  may be too big to do updates
- Solution: approximate inference
  - Track samples of  $X$ , not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states
- This is how robot localization works in practice

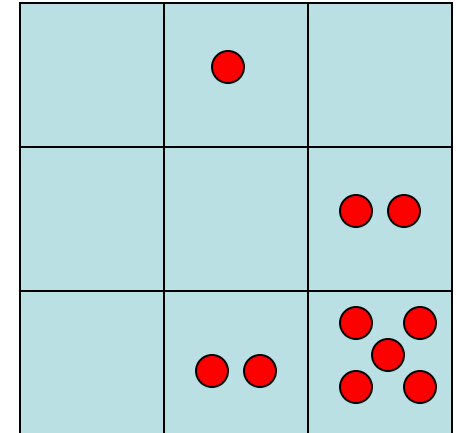
0.0	0.1	0.0
0.0	0.0	0.2
0.0	0.2	0.5



	●	
		● ●
	● ●	● ● ● ●

# Representation: Particles

- Our representation of  $P(X)$  is now a list of  $N$  particles (samples)
  - Generally,  $N \ll |X|$
  - Storing map from  $X$  to counts would defeat the point
- $P(x)$  approximated by number of particles with value  $x$ 
  - So, many  $x$  will have  $P(x) = 0$ !
  - More particles, more accuracy
- For now, all particles have a weight of 1



Particles:

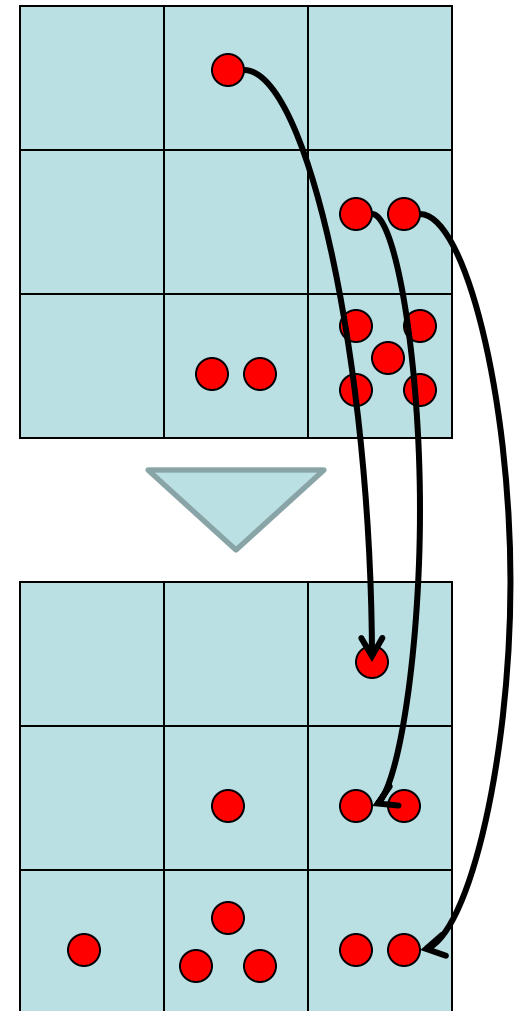
(3,3)  
(2,3)  
(3,3)  
(3,2)  
(3,3)  
(3,2)  
(2,1)  
(3,3)  
(3,3)  
(2,1)

# Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model

$$x' = \text{sample}(P(X'|x))$$

- This is like prior sampling – samples' frequencies reflect the transition probs
  - Here, most samples move clockwise, but some move in another direction or stay in place
- This captures the passage of time
    - If we have enough samples, close to the exact values before and after (consistent)



# Particle Filtering: Observe

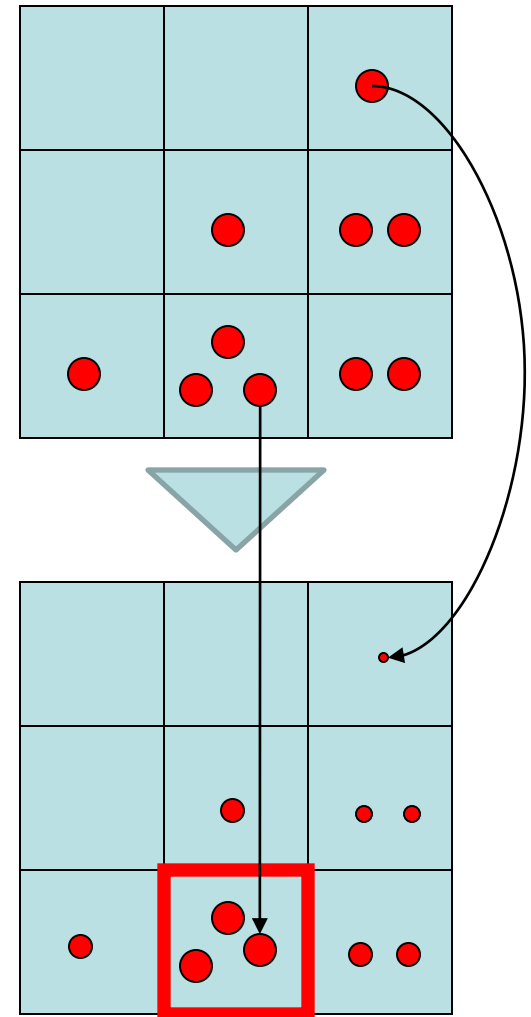
- Slightly trickier:

- Don't do rejection sampling (why not?)
- We don't sample the observation, we fix it
- This is similar to likelihood weighting, so we downweight our samples based on the evidence

$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

- Note that, as before, the probabilities don't sum to one, since most have been downweighted (in fact they sum to an approximation of  $P(e)$ )

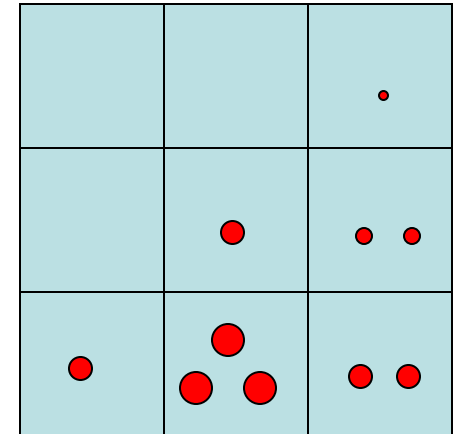


# Particle Filtering: Resample

- Rather than tracking weighted samples, we resample
- N times, we choose from our weighted sample distribution (i.e. draw with replacement)
- This is analogous to renormalizing the distribution
- Now the update is complete for this time step, continue with the next one

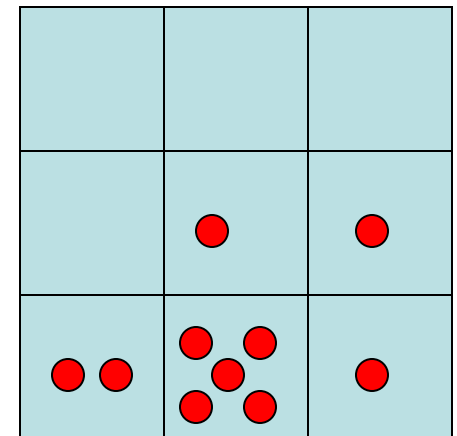
Old Particles:

(3,3)  $w=0.1$   
(2,1)  $w=0.9$   
(2,1)  $w=0.9$   
(3,1)  $w=0.4$   
(3,2)  $w=0.3$   
(2,2)  $w=0.4$   
(1,1)  $w=0.4$   
(3,1)  $w=0.4$   
(2,1)  $w=0.9$   
(3,2)  $w=0.3$



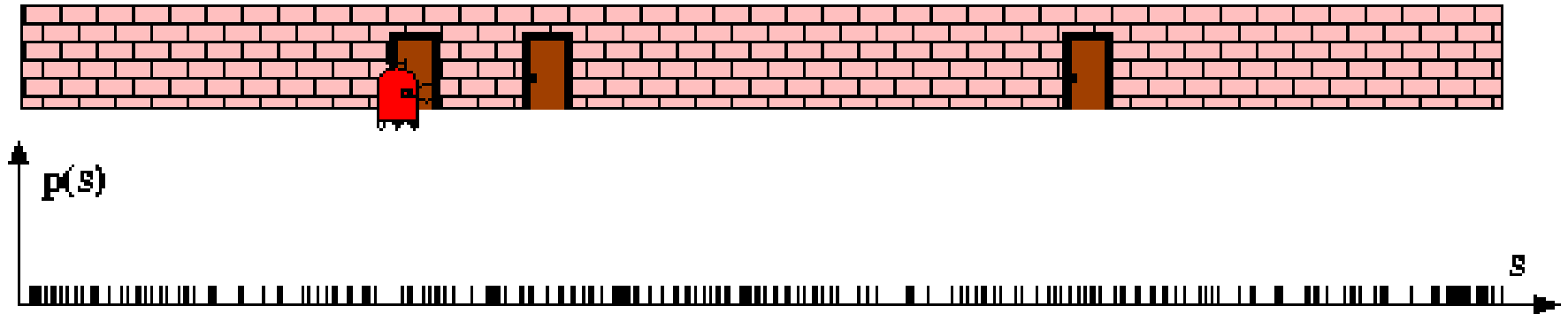
New Particles:

(2,1)  $w=1$   
(2,1)  $w=1$   
(2,1)  $w=1$   
(3,2)  $w=1$   
(2,2)  $w=1$   
(2,1)  $w=1$   
(1,1)  $w=1$   
(3,1)  $w=1$   
(2,1)  $w=1$   
(1,1)  $w=1$



# Particle Filters

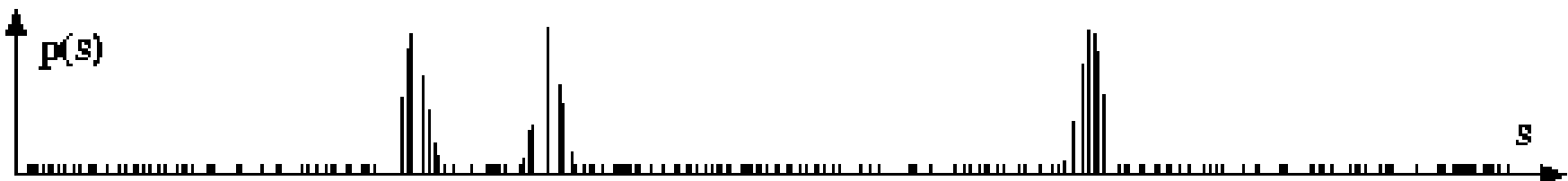
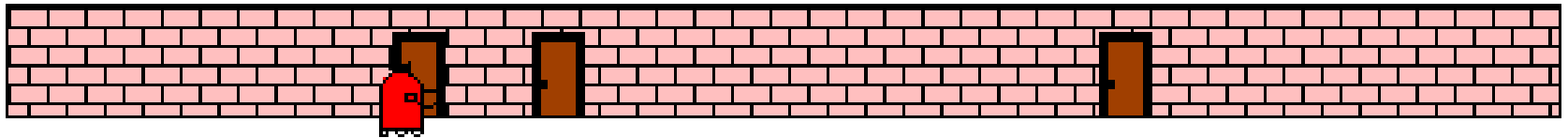
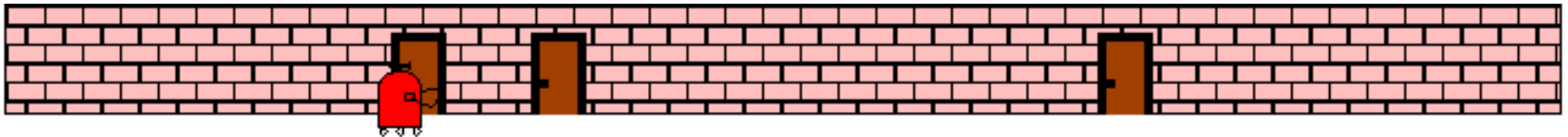
---



# Sensor Information: Importance Sampling

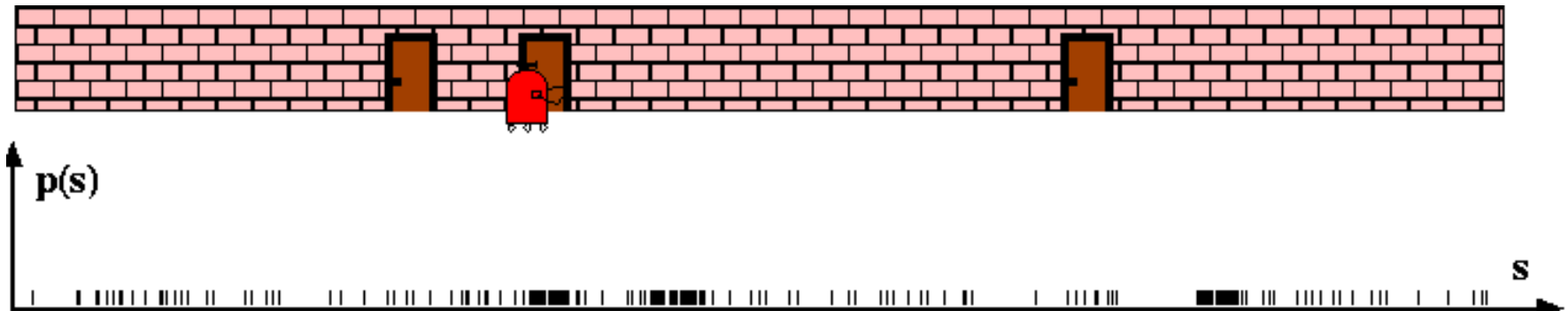
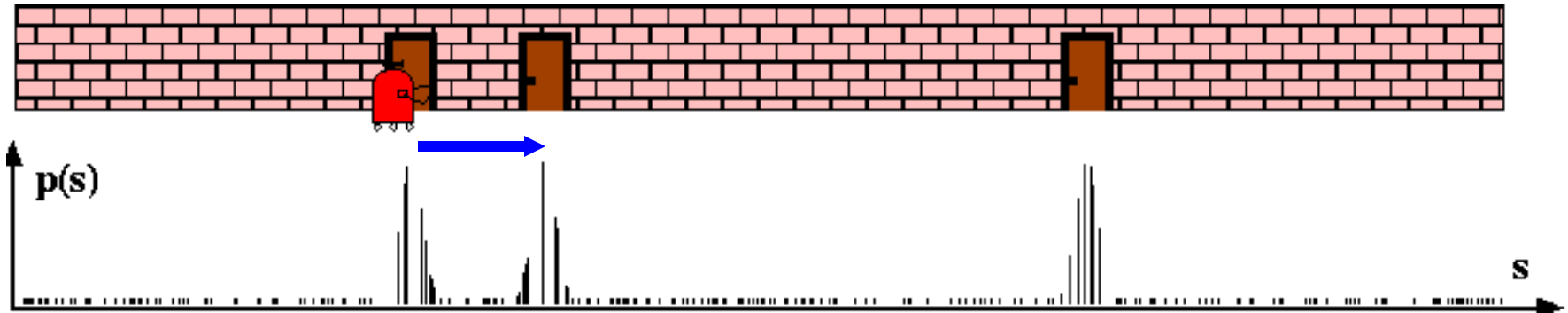
$$Bel(x) \leftarrow \alpha p(z|x) Bel^-(x)$$

$$w \leftarrow \frac{\alpha p(z|x) Bel^-(x)}{Bel^-(x)} = \alpha p(z|x)$$



# Robot Motion

$$Bel^-(x) \leftarrow \int p(x|u, x') Bel(x') dx'$$

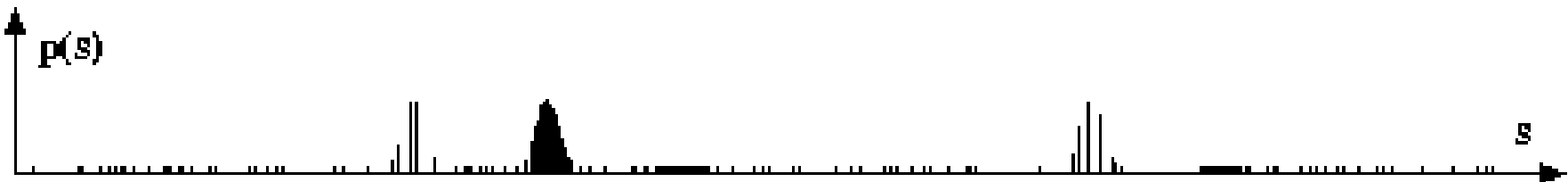
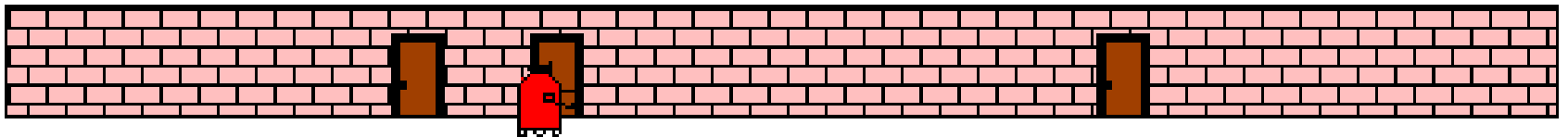
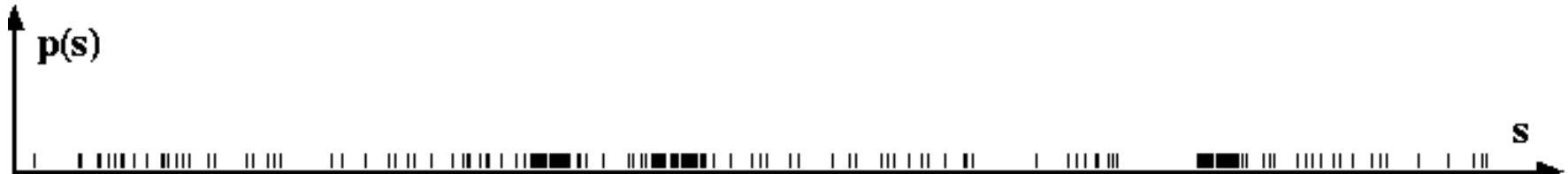
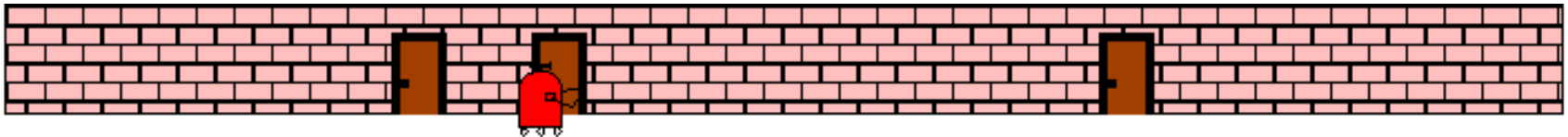




# Sensor Information: Importance Sampling

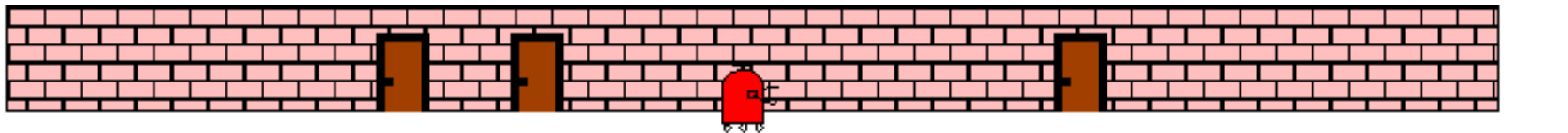
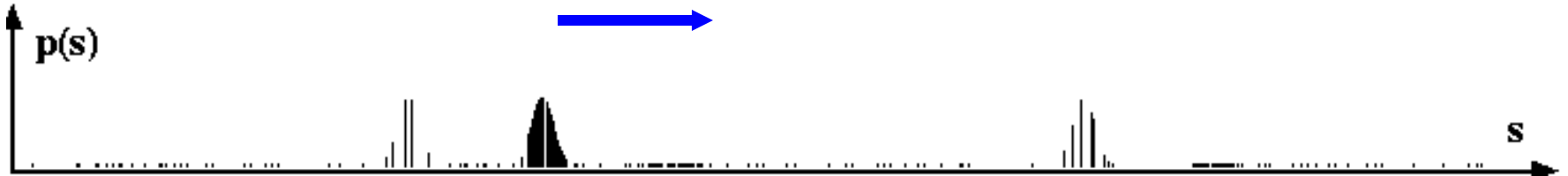
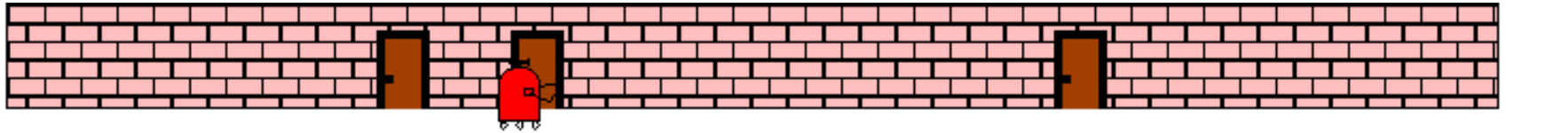
$$Bel(x) \leftarrow \alpha p(z|x) Bel^-(x)$$

$$w \leftarrow \frac{\alpha p(z|x) Bel^-(x)}{Bel^-(x)} = \alpha p(z|x)$$



# Robot Motion

$$Bel^-(x) \leftarrow \int p(x|u, x') Bel(x') dx'$$



# Particle Filter Algorithm

---

- Sample the next generation for particles using the proposal distribution
- Compute the importance weights :  
$$weight = target\ distribution / proposal\ distribution$$
- Resampling: “Replace unlikely samples by more likely ones”

# Particle Filter Algorithm

1. Algorithm **particle\_filter**(  $S_{t-1}, u_{t-1} z_t$ ):

2.  $S_t = \emptyset, \quad \eta = 0$

3. **For**  $i = 1 \dots n$

*Generate new samples*

4. Sample index  $j(i)$  from the discrete distribution given by  $w_{t-1}$

5. Sample  $x_t^i$  from  $p(x_t | x_{t-1}, u_{t-1})$  using  $x_{t-1}^{j(i)}$  and  $u_{t-1}^{j(i)}$

6.  $w_t^i = p(z_t | x_t^i)$

*Compute importance weight*

7.  $\eta = \eta + w_t^i$

*Update normalization*

*factor*  
 $S_t = S_t \cup \{ \langle x_t^i, w_t^i \rangle \}$

8.  $i = 1 \dots n$

*Insert*

9. **For**  $w_t^i = w_t^i / \eta$

10. *Normalize weights*

# Overview

---

- Markov Chains
- Hidden Markov Models
- Particle Filters
- More on HMMs

# Other uses of HMM

---

- Find most likely sequence of states
  - Viterbi algorithm
- Learn HMM parameters from data
  - Baum-Welch (EM) algorithm
- Other types of HMMs
  - Continuous, Gaussian-linear: Kalman filter
  - Structured transition/emission probabilities: Dynamic Bayes network (DBN)

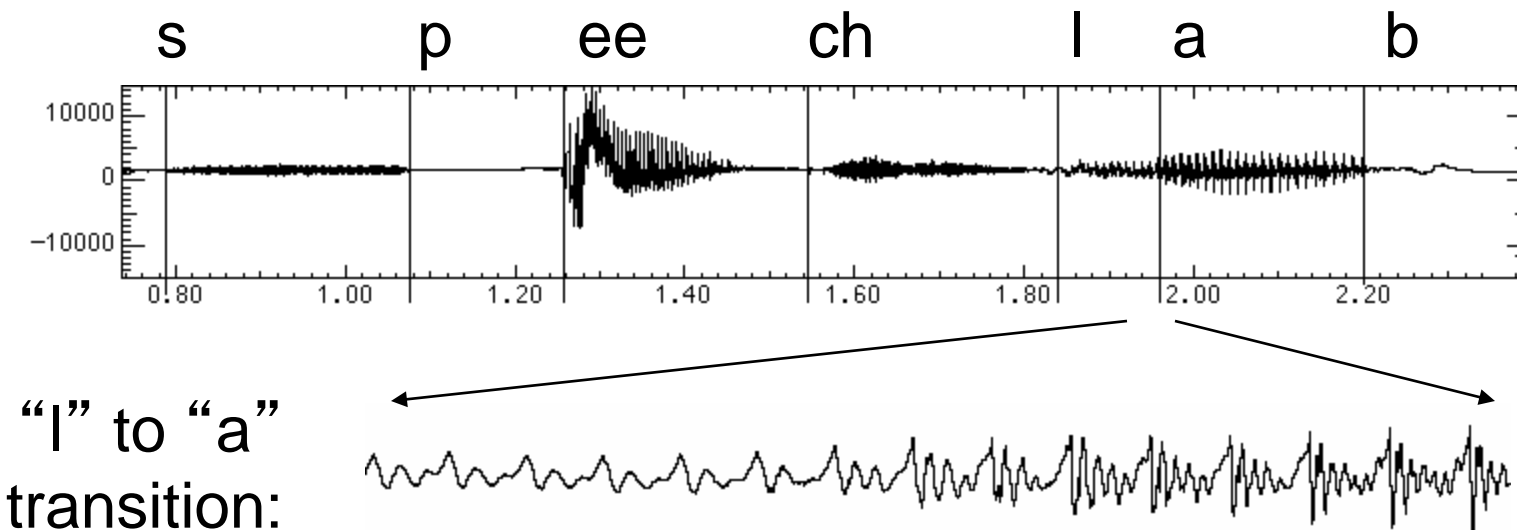
# Real HMM Examples

---

- **Speech recognition HMMs:**
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- **Machine translation HMMs:**
  - Observations are words (tens of thousands)
  - States are translation options (dozens per word)
- **Robot tracking:**
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

# HMM Application Domain: Speech

- Speech input is an acoustic wave form





# Time for Questions

---