COMP4620 – Advanced Topics in AI
Partially Observable Markov Decision Processes (POMDP) 1/3

Hanna Kurniawati

http://users.cecs.anu.edu.au/~hannakur/

hanna.kurniawati@anu.edu.au

Australian National University | RESEARCH SCHOOL OF COMPUTER SCIENCE

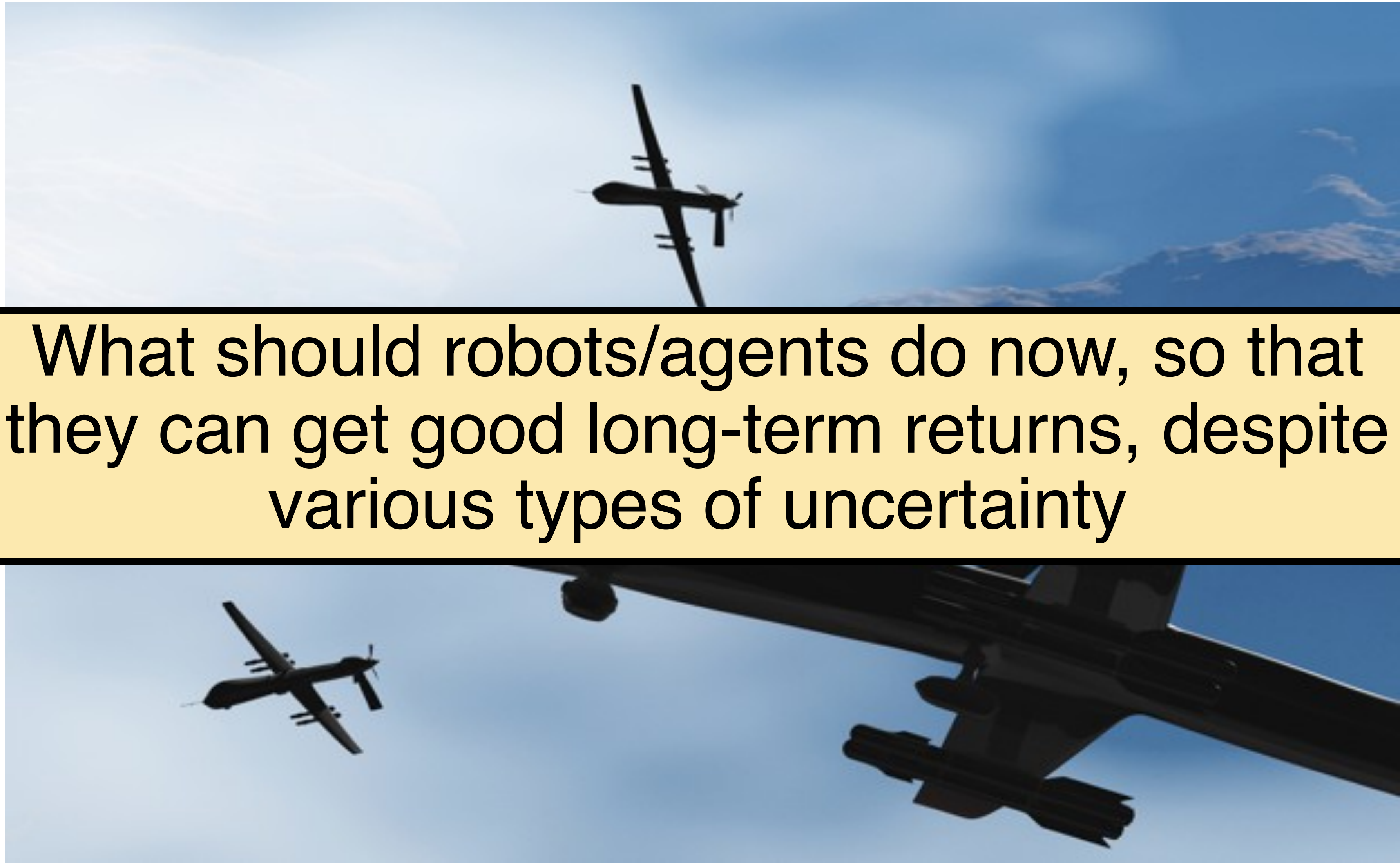
Topics

- Lecture 1: What is POMDPs?
- Lecture 2: How do we solve POMDPs?
- Lecture 3: Applications of POMDPs in Robotics & Cyber

What is POMDPs?

- Intuition
- Formal definitions
- Beliefs

The problem



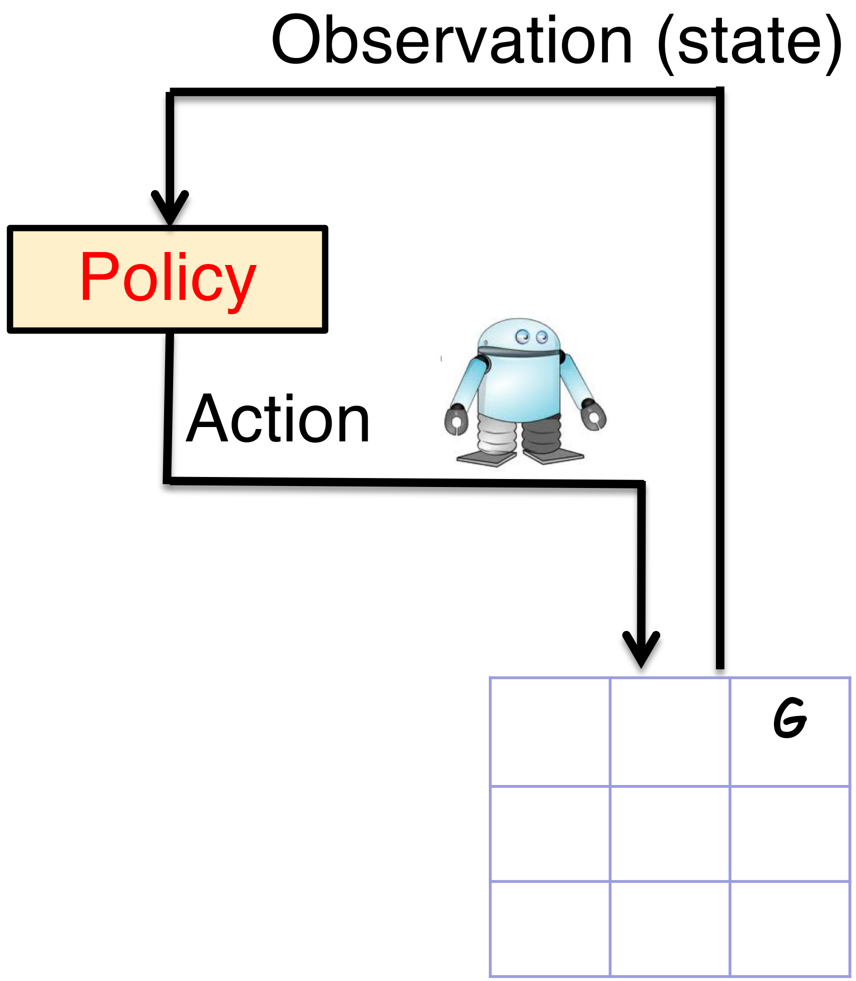
What should robots/agents do now, so that they can get good long-term returns, despite various types of uncertainty

Intelligent Agent: Types of environments

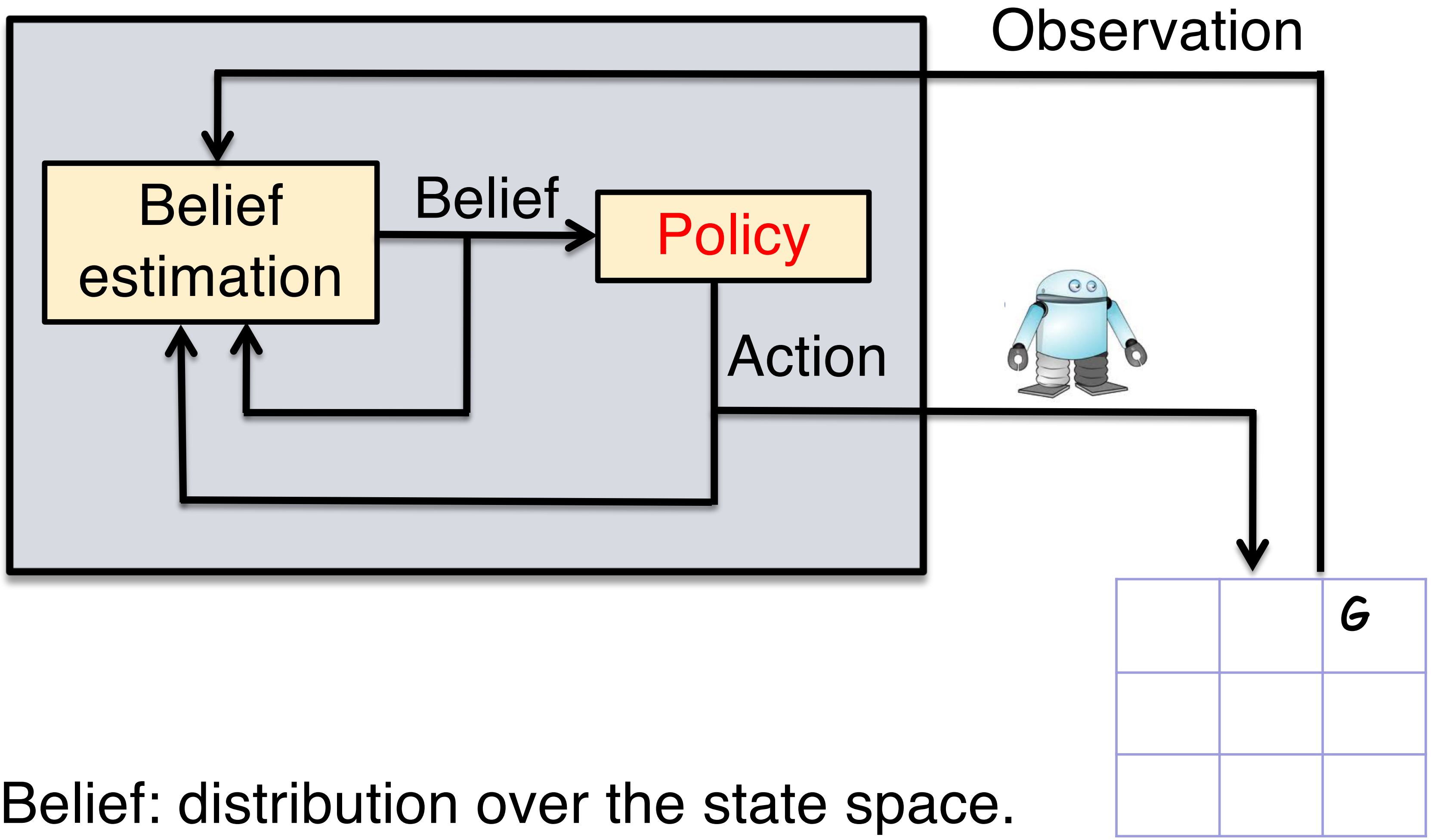
- Fully observable vs. partially observable
  - Does the agent know the state of the world exactly?
- Deterministic vs. non-deterministic
  - Does an action map one state into a single other state?
- Static vs. dynamic
  - Can the world change while the agent is “thinking”?
- Discrete vs. continuous
  - Are the sets of actions & percepts/observations discrete?

MDP: Non-Deterministic & ~~Fully~~ Partially Observable

1. Starts from the initial state.
2. Move according to the policy.
3. The agent moves to a new state.
   - The new state the agent ends up may be different in different runs.
4. Repeat to 2 until stopping criteria is satisfied (e.g., goal is reached)



Partially Observable Markov Decision Processes (POMDPs)



- Belief: distribution over the state space.
- Strategy/policy: mapping from beliefs to actions.

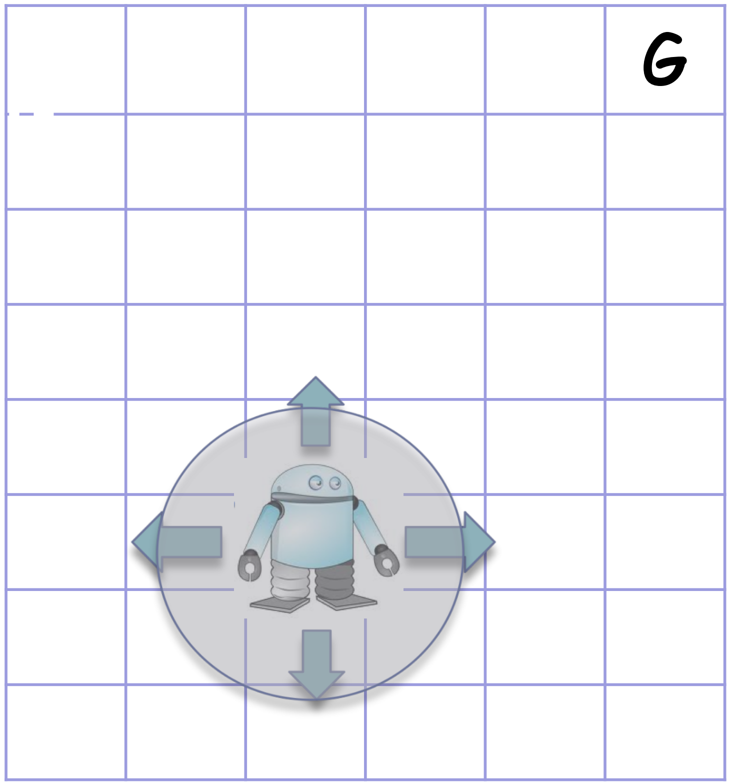
What is POMDPs?

✓Intuition

- Formal definitions
- Beliefs

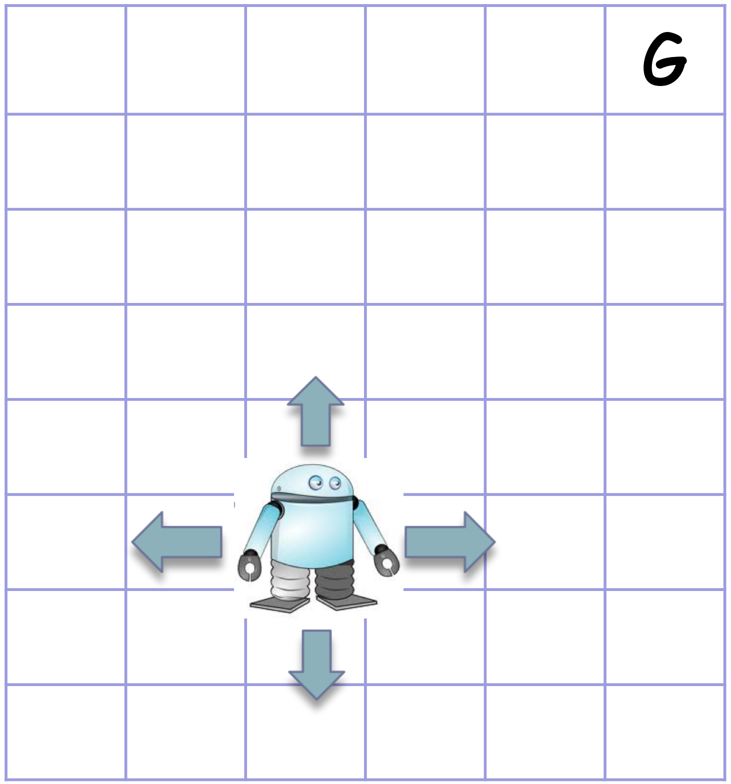
POMDP Model

- Main components:
  - State space (S)
  - Action space (A)
  - Observation space (O)



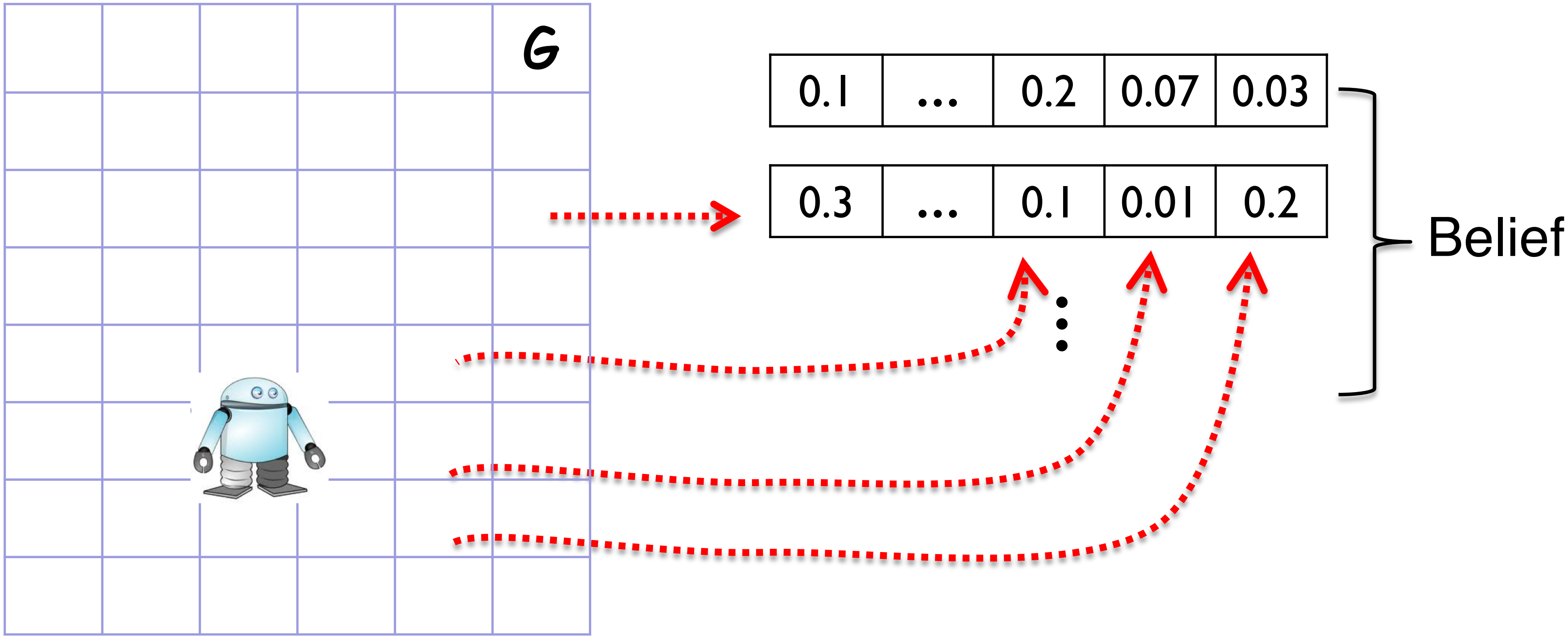
POMDP Model

- Main components:
  - State space (S) Not known
  - Action space (A)
  - Observation space (O)
  - Transition function (T)
    - $T = P(s' \mid s, a)$
  - Observation function (Z)
    - $Z = P(o \mid s', a)$
  - Reward function (R)
    - $R(s, a)$



$s, s' \in S, a \in A, o \in O$

POMDP Model



- Belief: distribution over the state space.
- Strategy/policy: mapping from beliefs to actions.

Belief can also be represented as parametric distributions

taken from *Statistical Inference* by Casella and Berger

Discrete Distrbutions

distribution	pmf	mean	variance	mgf/moment
Bernoulli(p)	$p^x(1-p)^{1-x};\ x=0,1;\ p\in(0,1)$	p	$p(1-p)$	$(1-p)+pe^t$
Beta-binomial(n,α,β)	$\binom{n}{x}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(x+\alpha)\Gamma(n-x+\beta)}{\Gamma(\alpha+\beta+n)}$	$\frac{n\alpha}{\alpha+\beta}$	$\frac{n\alpha\beta}{(\alpha+\beta)^2}$	
Notes: If $X\|P$ is binomial (n,P) and P is beta(α,β), then X is beta-binomial(n,α,β).				
Binomial(n,p)	$\binom{n}{x}p^x(1-p)^{n-x};\ x=1,\ldots,n$	np	$np(1-p)$	$[(1-p)+pe^t]^n$
Discrete Uniform(N)	$\frac{1}{N};\ x=1,\ldots,N$	$\frac{N+1}{2}$	$\frac{(N+1)(N-1)}{12}$	$\frac{1}{N}\sum_{i=1}^{N}e^{it}$
Geometric(p)	$p(1-p)^{x-1};\ p\in(0,1)$	$\frac{1}{p}$	$\frac{1-p}{p^2}$	$\frac{pe^t}{1-(1-p)e^t}$
Note: $Y=X-1$ is negative binomial$(1,p)$. The distribution is *memoryless*: $P(X>s\|X>t)=P(X>s-t)$.				
Hypergeometric(N,M,K)	$\frac{\binom{M}{x}\binom{N-M}{K-x}}{\binom{N}{K}};\ x=1,\ldots,K$	$\frac{KM}{N}$	$\frac{KM}{N}\frac{(N-M)(N-k)}{N(N-1)}$	?
	$M-(N-K)\le x\le M;\ N,M,K>0$			
Negative Binomial(r,p)	$\binom{r+x-1}{x}p^r(1-p)^x;\ p\in(0,1)$	$\frac{r(1-p)}{p}$	$\frac{r(1-p)}{p^2}$	$\left(\frac{p}{1-(1-p)e^t}\right)^r$
	$\binom{y-1}{r-1}p^r(1-p)^{y-r};\ Y=X+r$			
Poisson(λ)	$\frac{e^{-\lambda}\lambda^x}{x!};\ \lambda\ge 0$	λ	λ	$e^{\lambda(e^t-1)}$
Notes: If Y is gamma(α,β), X is Poisson$(\frac{x}{\beta})$, and α is an integer, then $P(X\ge\alpha)=P(Y\le y)$.				

“Best” policy

- Maps each belief to an action that satisfies the following objective function

$$V^*(b) = \max_{a \in A}\left(\sum_{s \in S} R(s,a)b(s) + \gamma \sum_{o \in O} P(o|b,a)V^*(b')\right)$$

Expected immediate reward

Expected total future reward

b': next belief after the system at belief b performs action a and observes o

γ: discount factor, (0,1)

Two notations from previous slides

- The next belief b'
- The function $P(o|b,a)$
  - Notice that the definition in the observation function is conditioned on the **resulting** state after the action is performed, while in this notation, b is the belief **from where** the action is performed

Formulation for b' and $P(o|b,a)$?

- Mathematically,
  - Use Bayes rule

$$
\begin{aligned}
b'(s') &= P(s'|o,a,b) \\
&= \frac{P(o|s',a,b)P(s'|a,b)\cancel{P(a,b)}}{P(o|a,b)\cancel{P(a,b)}} \\
&= \frac{P(o|s',a,)\sum_s P(s'|a,s)b(s)}{\sum_{s''}(P(o|a,s'')\sum_s P(s''|a,s)b(s))}
\end{aligned}
$$

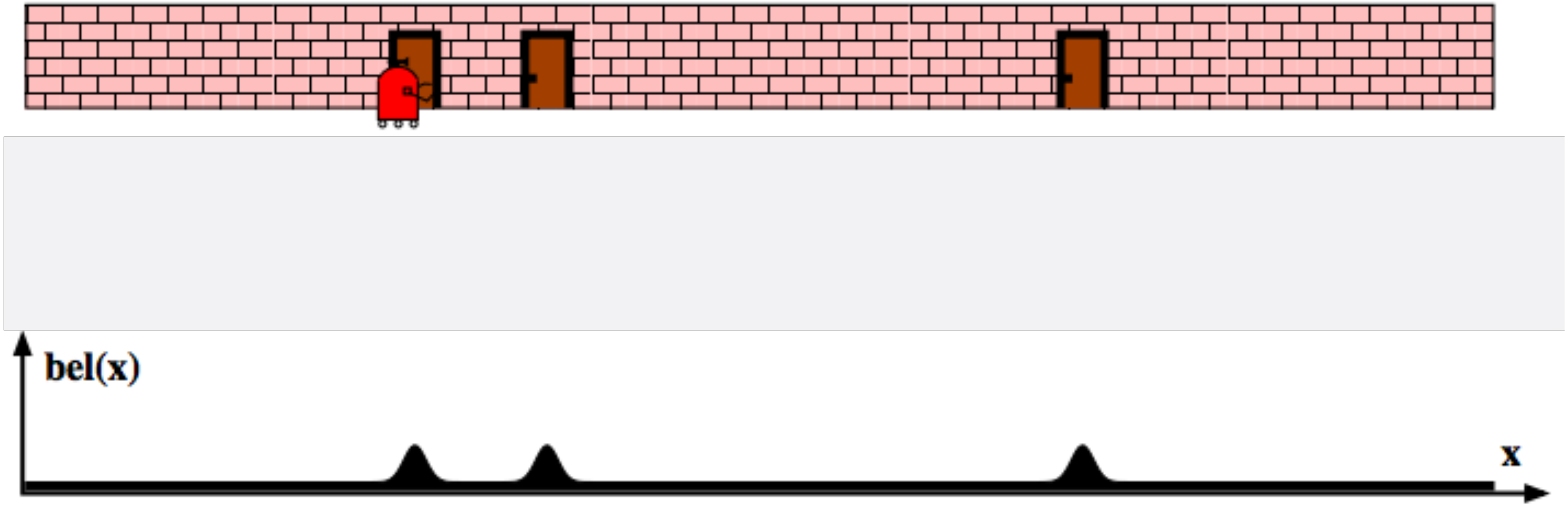
The denominator $P(o|a,b)$ is essentially the normalizing factor that makes b' over the state space sums to 1.

What is POMDPs?

- ✓ Intuition
- ✓ Formal definitions
- Beliefs

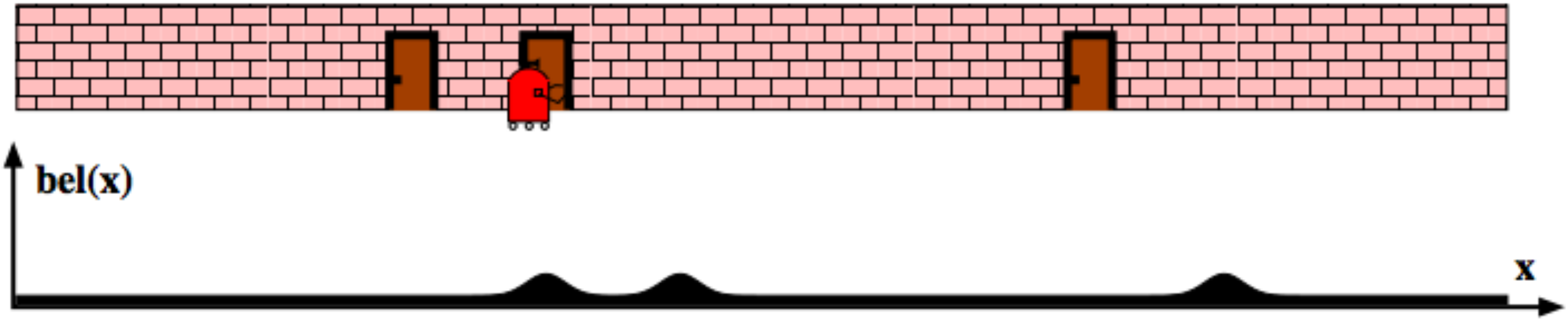
How to compute the next belief?

- Intuitively, divide into 2 steps
  - Compute the next belief after an action is performed
  - Adjust the belief based on the perceived observation



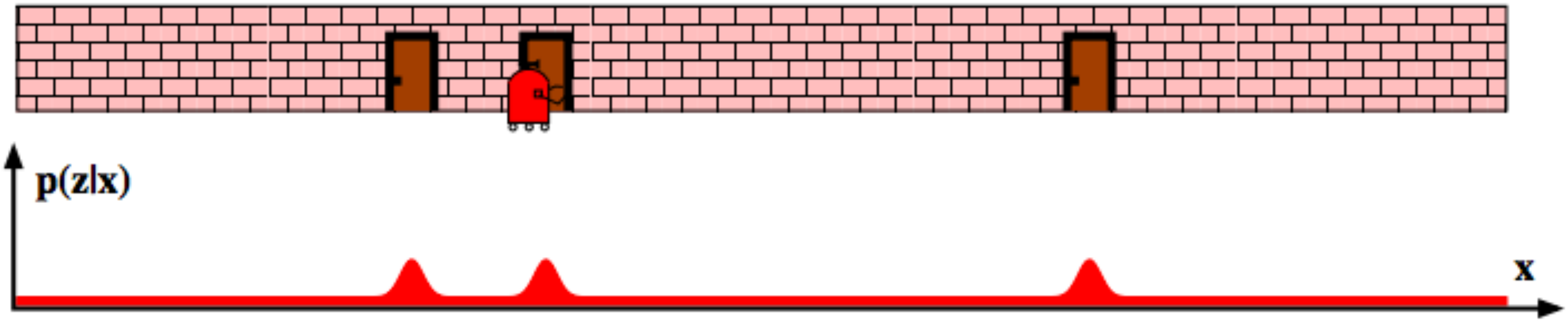
How to compute the next belief?

- Intuitively, divide into 2 steps
  - Compute the next belief after an action is performed
  - Adjust the belief based on the perceived observation



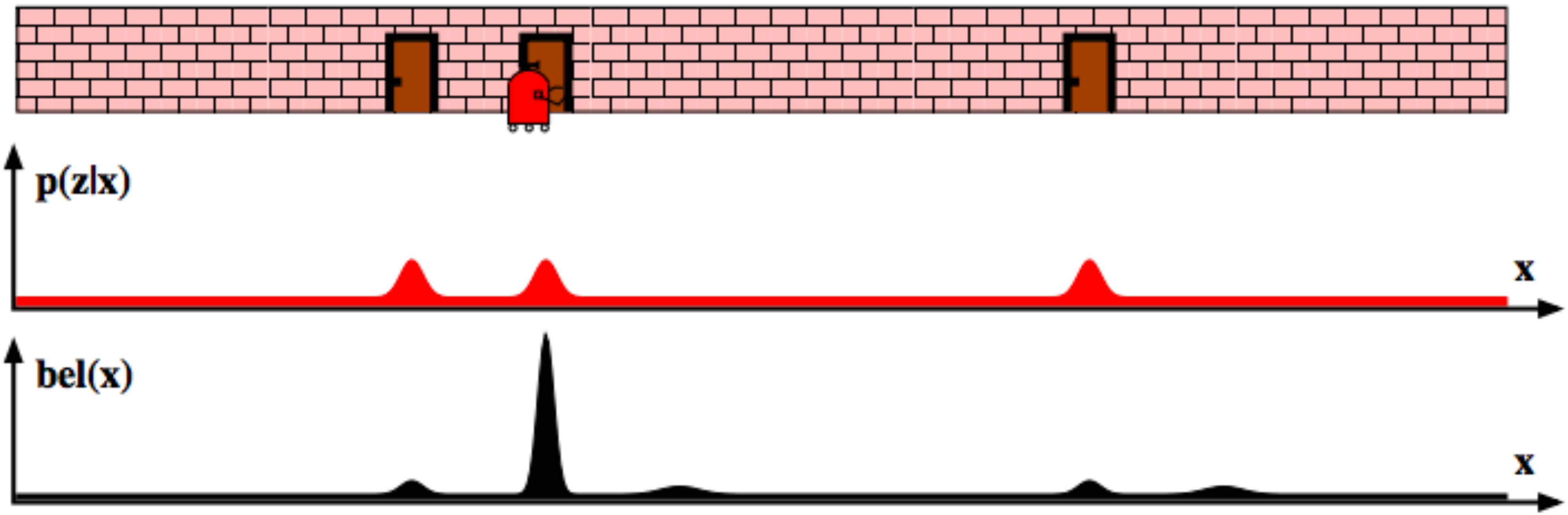
How to compute the next belief?

- Intuitively, divide into 2 steps
  - Compute the next belief after an action is performed
  - Adjust the belief based on the perceived observation



How to compute the next belief?

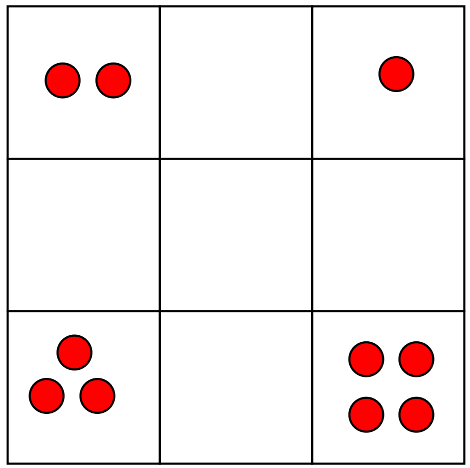
- Intuitively, divide into 2 steps
  - Compute the next belief after an action is performed
  - Adjust the belief based on the perceived observation



How to implement? Particle Filter

- Represent distributions as particles.
  - Basically as a set of single states.
    - Can be weigthed or unweighted
  - The number of sets at a certain region, represents the probability.

0.2	0.0	0.1
0.0	0.0	0.0
0.3	0.0	0.4

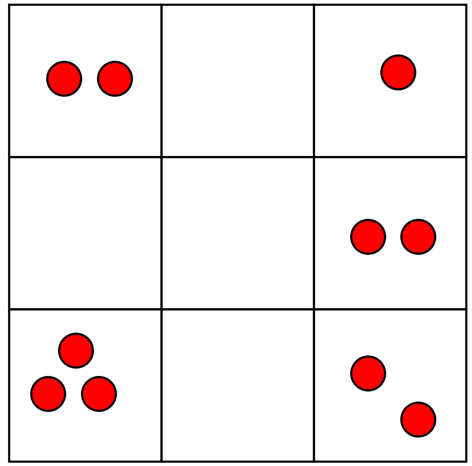


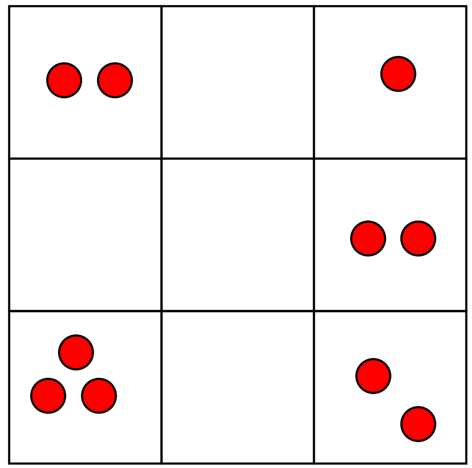
How to implement? Particle Filter

- Recall 2 steps:
  - Effect of action a: For each particle s, compute next particle by sampling from transition function T (i.e., P(s' l s, a))
  - Effect of observation o:
    - Assign weight to the samples, based on Bayes rule on Z (i.e., P(o l s, a)P(s))
    - Resample based on the weight

How to implement? Particle Filter

- Two steps:
  - Predict next state, based on dynamics. For each particle s, compute next particle by sampling from the transition function

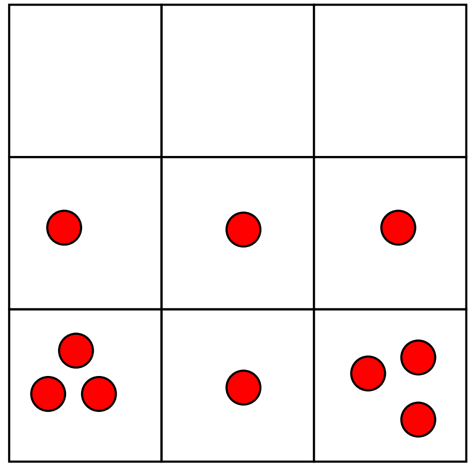


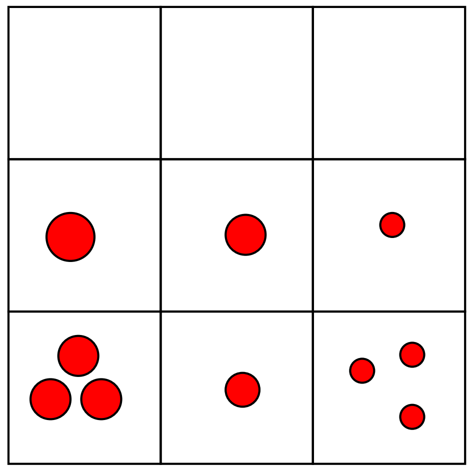


How to implement? Particle Filter

- Two steps:
  - Predict next state, based on dynamics. For each particle s, compute next particle by sampling from the transition function
  - Update based on data/measurement
    - Assign weight to the samples based on the measurement (o) perceived:

$$w(s) = P(o \mid s)\, P(s)$$



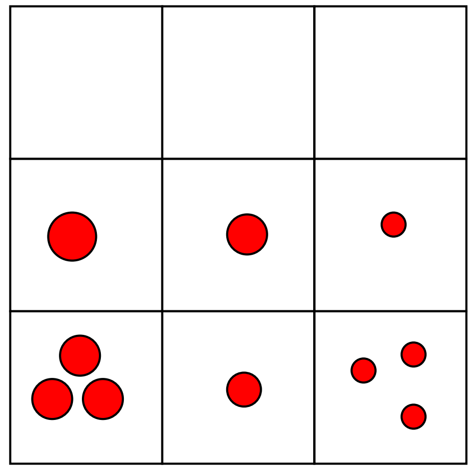


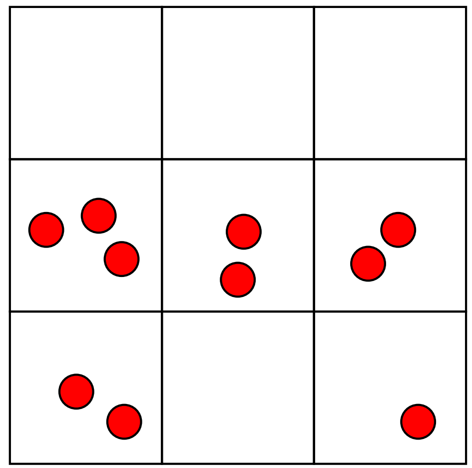
How to implement? Particle Filter

- Two steps:
  - Predict next state, based on dynamics. For each particle s, compute next particle by sampling from the transition function
  - Update based on data/measurement
    - Assign weight to the samples based on the measurement (o) perceived:

      $$w(s) = P(o \mid s)\, P(s)$$

    - Resample based on the weight to get a list of (unweighted) particles that represent the new distribution.





What is POMDPs?

- ✓ Intuition
- ✓ Formal definitions
- ✓ Beliefs

Next: Solving POMDPs