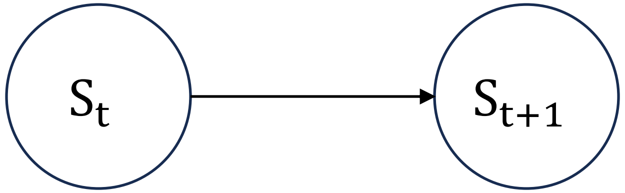
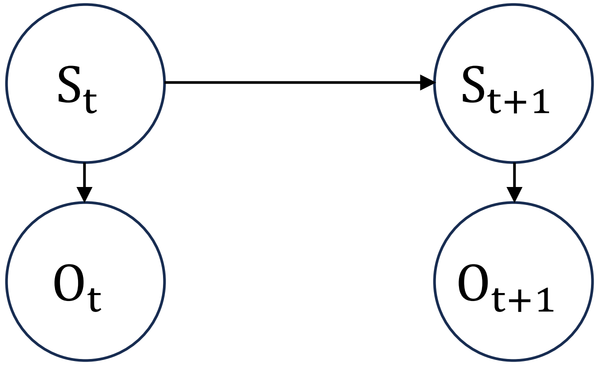
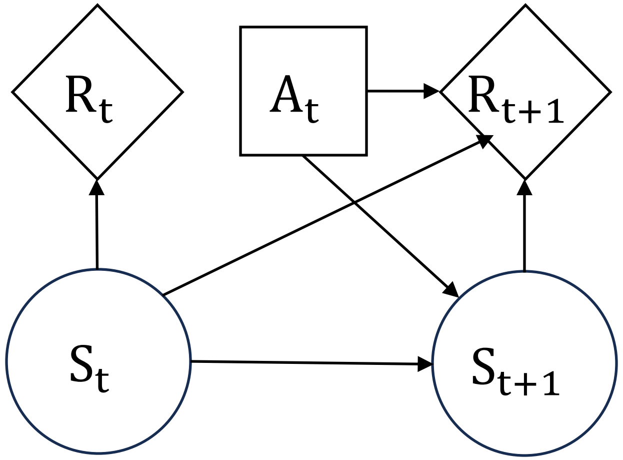
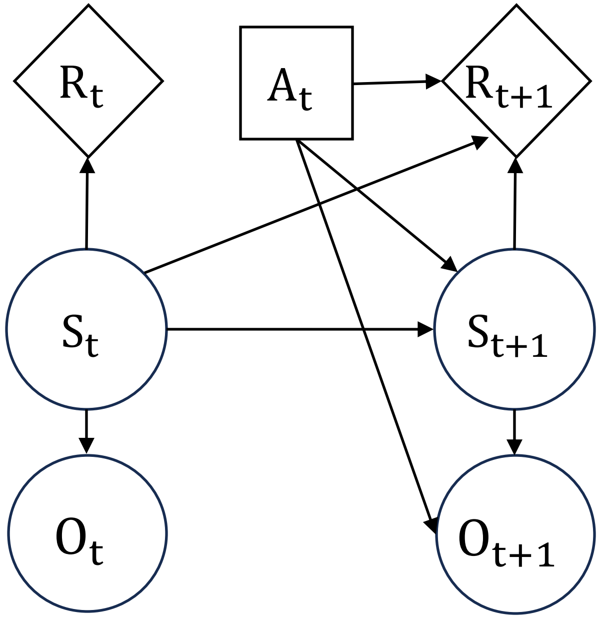
(First, let’s finish off MCTS…)

Planning with Partial Observability

Tom Silver
Robot Planning Meets Machine Learning
Princeton University
Fall 2025

Building Toward POMDPs

	Fully Observed	Partially Observed
Passive	 **Markov Chain**	 **Hidden Markov Model (HMM)**
Active	 **Markov Decision Process (MDP)**	 **Partially Observed Markov Decision Process (POMDP)**

Hidden Markov Model (HMM)

HMM: sequence of random variables $S_0, S_1, S_2, \ldots$, with domain $\mathcal{S}$ and random variables $O_0, O_1, O_2, \ldots,$ with domain $\mathcal{O}$ s.t.

$$P(S_0, S_1, S_2, \ldots, O_0, O_1, O_2, \ldots,) = P(S_0) \prod_t P(S_{t+1} | S_t) P(O_t | S_t)$$

Hidden Markov Model (HMM)

HMM: sequence of random variables $S_0, S_1, S_2, \ldots$, with domain $\mathcal{S}$ and random variables $O_0, O_1, O_2, \ldots,$ with domain $\mathcal{O}$ s.t.

$$P(S_0, S_1, S_2, \ldots, O_0, O_1, O_2, \ldots,) = P(S_0) \prod_t P(S_{t+1}|S_t) P(O_t|S_t)$$

Initial state distribution

Hidden Markov Model (HMM)

HMM: sequence of random variables $S_0, S_1, S_2, \dots,$ with domain $\mathcal{S}$ and random variables $O_0, O_1, O_2, \dots,$ with domain $\mathcal{O}$ s.t.

$$P(S_0, S_1, S_2, \dots, O_0, O_1, O_2, \dots,) = P(S_0) \prod_t P(S_{t+1} | S_t) P(O_t | S_t)$$

Initial state distribution

Transition model

Hidden Markov Model (HMM)

HMM: sequence of random variables $S_0, S_1, S_2, \ldots,$ with domain $\mathcal{S}$ and random variables $O_0, O_1, O_2, \ldots,$ with domain $\mathcal{O}$ s.t.

$$P(S_0, S_1, S_2, \ldots, O_0, O_1, O_2, \ldots,) = \underbrace{P(S_0)}_{\text{Initial state distribution}} \prod_t \underbrace{P(S_{t+1} | S_t)}_{\text{Transition model}} \underbrace{P(O_t | S_t)}_{\textbf{Observation model}}$$

Hidden Markov Model (HMM)

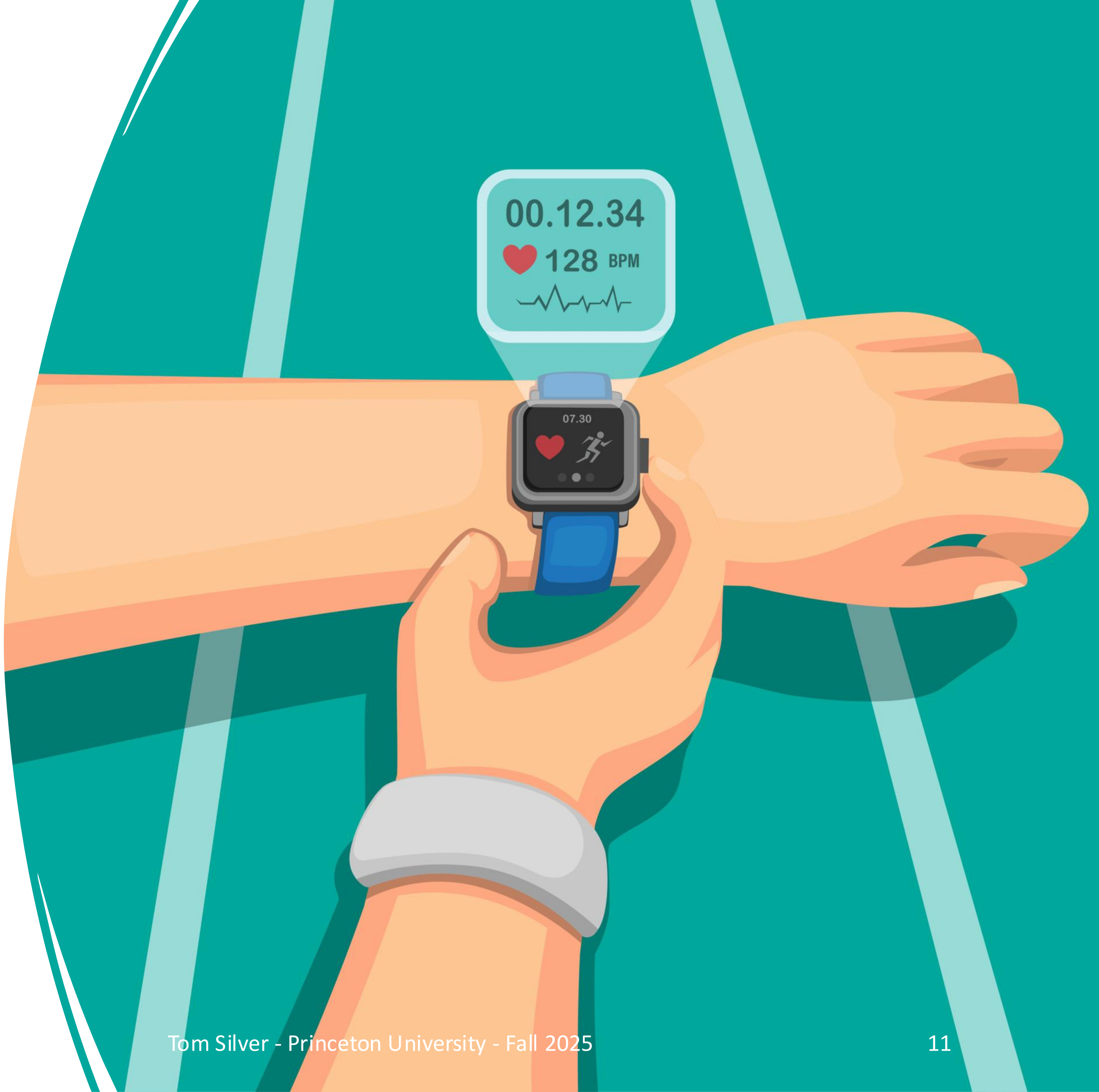
HMM: sequence of random variables $S_0, S_1, S_2, \dots$, with domain $\mathcal{S}$ and random variables $O_0, O_1, O_2, \dots$, with domain $\mathcal{O}$ s.t.

The **observation space**

$$P(S_0, S_1, S_2, \dots, O_0, O_1, O_2, \dots,) = \underbrace{P(S_0)}_{\text{Initial state distribution}} \prod_t \underbrace{P(S_{t+1} | S_t)}_{\text{Transition model}} \underbrace{P(O_t | S_t)}_{\textbf{Observation model}}$$

Example

How would we represent this scenario as a Hidden Markov Model?



HMM Inference: Filtering

As we receive observations, maintain **belief** about current state:

$$\mathrm{b}_0(\mathrm{s}) \triangleq P(S_0 = s \mid O_0)$$

$$\mathrm{b}_1(\mathrm{s}) \triangleq P(S_1 = s \mid O_0, O_1)$$

$$\mathrm{b}_2(\mathrm{s}) \triangleq P(S_2 = s \mid O_0, O_1, O_2)$$

…

HMM Inference: Filtering

As we receive observations, maintain **belief** about current state:

$\mathrm{b}_0(\mathrm{s}) \triangleq P(S_0 = s \mid O_0) \propto P(O_0 \mid S_0 = s) P(S_0 = s)$

Bayes' Theorem

$\mathrm{b}_1(\mathrm{s}) \triangleq P(S_1 = s \mid O_0, O_1)$

$\mathrm{b}_2(\mathrm{s}) \triangleq P(S_2 = s \mid O_0, O_1, O_2)$

…

HMM Inference: Filtering

As we receive observations, maintain **belief** about current state:

$\mathrm{b}_0(\mathrm{s}) \triangleq P(S_0 = s \mid O_0) \propto P(O_0 \mid S_0 = s) P(S_0 = s)$

Bayes' Theorem

$\mathrm{b}_1(\mathrm{s}) \triangleq P(S_1 = s \mid O_0, O_1)$

Can we compute b_{t+1} in terms of b_t?

$\mathrm{b}_2(\mathrm{s}) \triangleq P(S_2 = s \mid O_0, O_1, O_2)$

...

HMM Inference: Filtering

$$\mathrm{b}_{t+1}(\mathrm{s}') \propto P(O_{t+1} \mid S_{t+1} = s') \sum_{s} P(S_{t+1} = s' | S_t = s)\mathrm{b}_t(s)$$

Follows from definition of HMM

Forward algorithm or *Viterbi*

HMM Example: Moody Friend

- **States**: *mood* in {0, 1, 2}
- **Observations:** *face* in {smile, frown}
- **Transition distribution:**
  - Stay the same with 0.8 probability
  - Otherwise, move to adjacent mood with uniform probability
- **Observation model:**
  - P(smile | mood = 0) = 0.1
  - P(smile | mood = 1) = 0.5
  - P(smile | mood = 2) = 0.9

$$b_0(s') \propto P(O_0 = \text{🙂} \mid S_0 = s')P(S_0 = s')$$

		$s' = 0$	$s' = 1$	$s' = 2$
🙂	$b_0(s')$	0.067	0.333	0.6

$$b_0(0) \propto P(O_0 = \text{🙂} \mid S_0 = 0)P(S_0 = 0)$$
$$\propto 0.1(0.333\ldots)$$

$$b_0(1) \propto P(O_0 = \text{🙂} \mid S_0 = 1)P(S_0 = 1)$$
$$\propto 0.5(0.333\ldots)$$

$$b_0(2) \propto P(O_0 = \text{🙂} \mid S_0 = 2)P(S_0 = 2)$$
$$\propto 0.9(0.333\ldots)$$

Practice On Your Own: Compute b_{t+1} Given Next Observation: ☹️

		$s' = 0$	$s' = 1$	$s' = 2$
🙂	$b_0(s')$	0.067	0.333	0.6
☹️	$b_1(s')$			

$$b_{t+1}(s') \propto P(O_{t+1} \mid S_{t+1} = s') \sum_{s} P(S_{t+1} = s' \mid S_t = s) b_t(s)$$

		$s' = 0$	$s' = 1$	$s' = 2$
🙂	$b_0(s')$	0.067	0.333	0.6
☹️	$b_1(s')$			

$$b_1(s') \propto P\left(O_1 = \text{☹️} \,\middle|\, S_1 = s'\right) \sum_s P(S_1 = s' | S_0 = s) b_0(s)$$

$$b_1({\color{red}0}) \propto P\left(O_1 = \text{☹️} \,\middle|\, S_1 = {\color{red}0}\right) \sum_s P(S_1 = {\color{red}0} | S_0 = s) b_0(s)$$

$$\begin{aligned} \propto P\left(O_1 = \text{☹️} \,\middle|\, S_1 = {\color{red}0}\right) [&P(S_1 = {\color{red}0} | S_0 = {\color{red}0}) b_0({\color{red}0}) + \\ &P(S_1 = {\color{red}0} | S_0 = {\color{blue}1}) b_0({\color{blue}1}) + \\ &P(S_1 = {\color{red}0} | S_0 = {\color{green}2}) b_0({\color{green}2})] \end{aligned}$$

$$\propto {\color{red}0.9\ [0.8(0.067) + 0.1(0.333) + 0.0(0.6)] = 0.07821}$$

		$s' = 0$	$s' = 1$	$s' = 2$
🙂	$b_0(s')$	0.067	0.333	0.6
☹	$b_1(s')$			

$$b_1(s') \propto P\left(O_1 = \text{☹} \,\middle|\, S_1 = s'\right) \sum_s P(S_1 = s' | S_0 = s) b_0(s)$$

$$b_1(0) \propto 0.07821$$

$$b_1(1) \propto P\left(O_1 = \text{☹} \,\middle|\, S_1 = 1\right) \sum_s P(S_1 = 1 | S_0 = s) b_0(s)$$

$$\propto P\left(O_1 = \text{☹} \,\middle|\, S_1 = 1\right) [P(S_1 = 1 | S_0 = 0) b_0(0) +$$
$$P(S_1 = 1 | S_0 = 1) b_0(1) +$$
$$P(S_1 = 1 | S_0 = 2) b_0(2)]$$

$$\propto 0.5\,[0.1(0.067) + 0.8(0.333) + 0.1(0.6)] = 0.16655$$

	$s' = 0$	$s' = 1$	$s' = 2$
🙂 $b_0(s')$	0.067	0.333	0.6
☹️ $b_1(s')$			

$$b_1(0) \propto 0.07821$$
$$b_1(1) \propto 0.16655$$

$$b_1(2) \propto P(O_1 = \text{☹️} \mid S_1 = 2) \sum_s P(S_1 = 2 \mid S_0 = s) b_0(s)$$

$$\propto P(O_1 = \text{☹️} \mid S_1 = 2) \left[P(S_1 = 2 \mid S_0 = 0) b_0(0) + P(S_1 = 2 \mid S_0 = 1) b_0(1) + P(S_1 = 2 \mid S_0 = 2) b_0(2) \right]$$

$$\propto 0.1\ [0(0.067) + 0.1(0.333) + 0.8(0.6)] = 0.05133$$

	$s' = 0$	$s' = 1$	$s' = 2$
🙂 $b_0(s')$	0.067	0.333	0.6
☹️ $b_1(s')$	0.264	0.562	0.173

$$b_1(0) \propto 0.07821$$
$$b_1(1) \propto 0.16655$$
$$b_1(2) \propto 0.05133$$

$\xrightarrow{\text{Normalize}}$

Partially Observable MDP (POMDP)

POMDP: MDP + HMM.

Partially Observable MDP (POMDP)

POMDP: MDP + HMM.

- State space $\mathcal{S}$ No longer observed!
- Action space $\mathcal{A}$
- Reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$
- Transition distribution $P(S_{t+1} \mid A_t, S_t)$

Partially Observable MDP (POMDP)

POMDP: MDP + HMM.

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$
- Transition distribution $P(\, S_{t+1} \mid A_t, S_t \,)$
- **Observation space $\mathcal{O}$**
- **Observation model** $P(\, O_t \mid A_{t-1}, S_t \,)$

Note: can depend on previous action

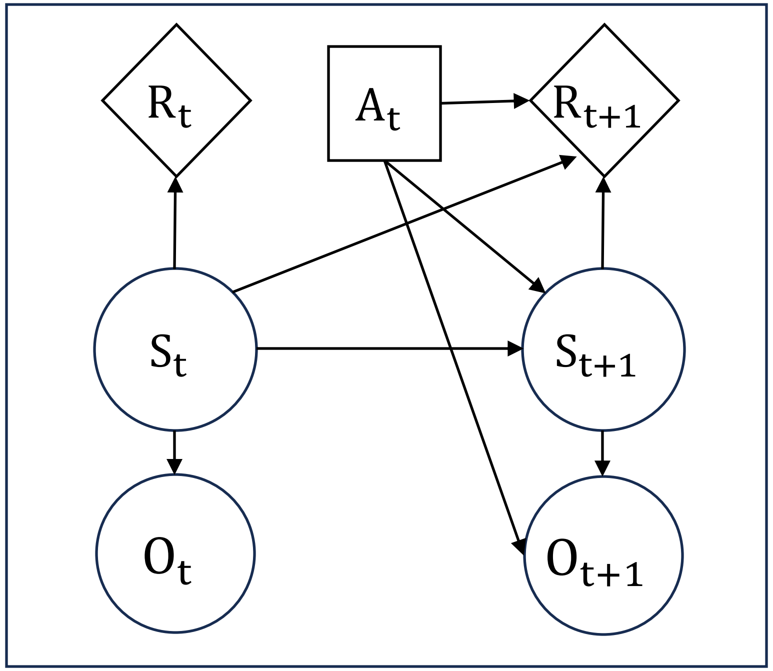
Partially Observable MDP (POMDP)

POMDP: MDP + HMM.

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$
- Transition distribution $P(S_{t+1} \mid A_t, S_t)$
- **Observation space $\mathcal{O}$**
- **Observation model** $P(O_t \mid A_{t-1}, S_t)$

Sometimes also define *initial* observation model $P(O_0 \mid S_0)$ since there is no previous action then

POMDP Influence Diagram



Things Carried Over from MDP Land

- Assume finite state, action, and now observation spaces
- Discrete time; possibly finite, infinite, or indefinite horizon
- Still want to maximize expected utility

POMDP State Estimation

As with HMMs, we will want to maintain **belief** about current state.

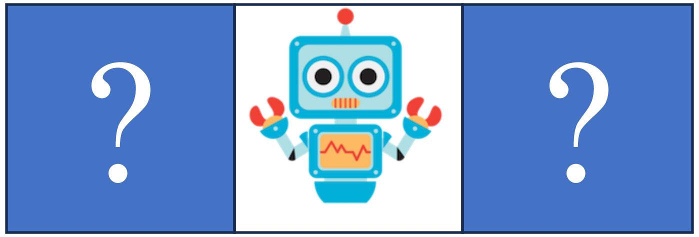
$$\mathrm{b}_{t+1}(\mathrm{s}') \propto P(O_{t+1} | S_{t+1} = s', A_t) \sum_{s} P(S_{t+1} = s' | S_t = s, A_t) \mathrm{b}_t(s)$$

Now conditioned on actions

Sometimes called *state estimation*

Example: Treasure Hunt

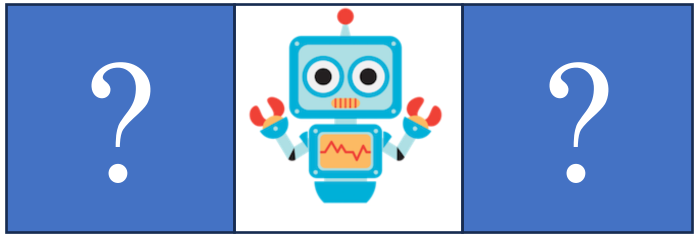
- **States**: (*robot-loc*, *treasure-loc*) in 3-cell grid
- **Actions:** *move-left*, *move-right*, *scan-left*, *scan-right*
- **Observations:** (*robot-loc*, *scan-response*)
  - Scan responses: *got-response*, *got-no-response*, *not-applicable*
- **Rewards:** 100 if robot at treasure, -1 for each step
- **Horizon:** indefinite (terminate when treasure obtained)
- **Transition distribution:**
  - Move actions: succeed with 0.95 probability. Otherwise, move in the other direction
  - Scan actions result in no state change
- **Observation model:**
  - Scan actions: get accurate response with 0.9 probability
  - Both actions: always observe correct robot location
  - Initial timestep: observe robot location only



Example: Treasure Hunt

Under what circumstances should the robot scan before moving?

- **States**: (*robot-loc*, *treasure-loc*) in 3-cell grid
- **Actions:** *move-left*, *move-right*, *scan-left*, *scan-right*
- **Observations:** (*robot-loc*, *scan-response*)
  - Scan responses: *got-response*, *got-no-response*, *not-applicable*
- **Rewards:** 100 if robot at treasure, -1 for each step
- **Horizon:** indefinite (terminate when treasure obtained)
- **Transition distribution:**
  - Move actions: succeed with 0.95 probability. Otherwise, move in the other direction
  - Scan actions result in no state change
- **Observation model:**
  - Scan actions: get accurate response with 0.9 probability
  - Both actions: always observe correct robot location
  - Initial timestep: observe robot location only



Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

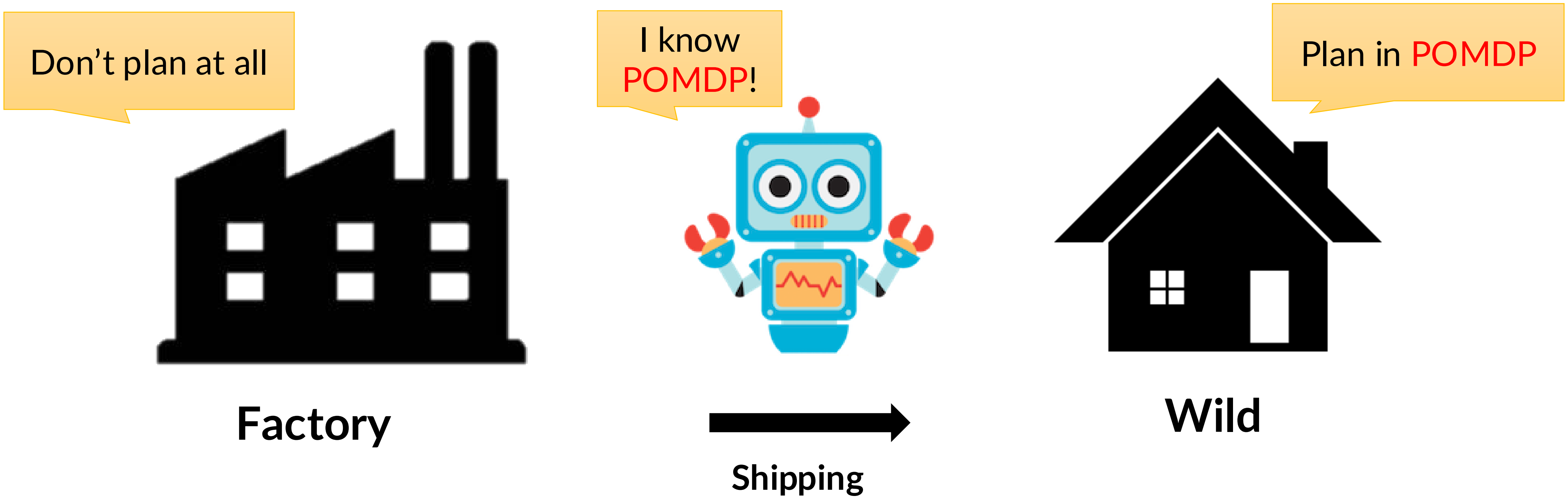
Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

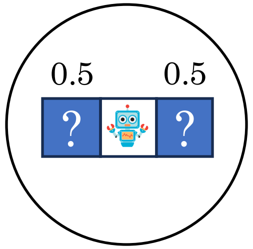
We'll start here this time

POMDP Planning Online (In the Wild)

We ship the robot with the POMDP and have it plan **online.**

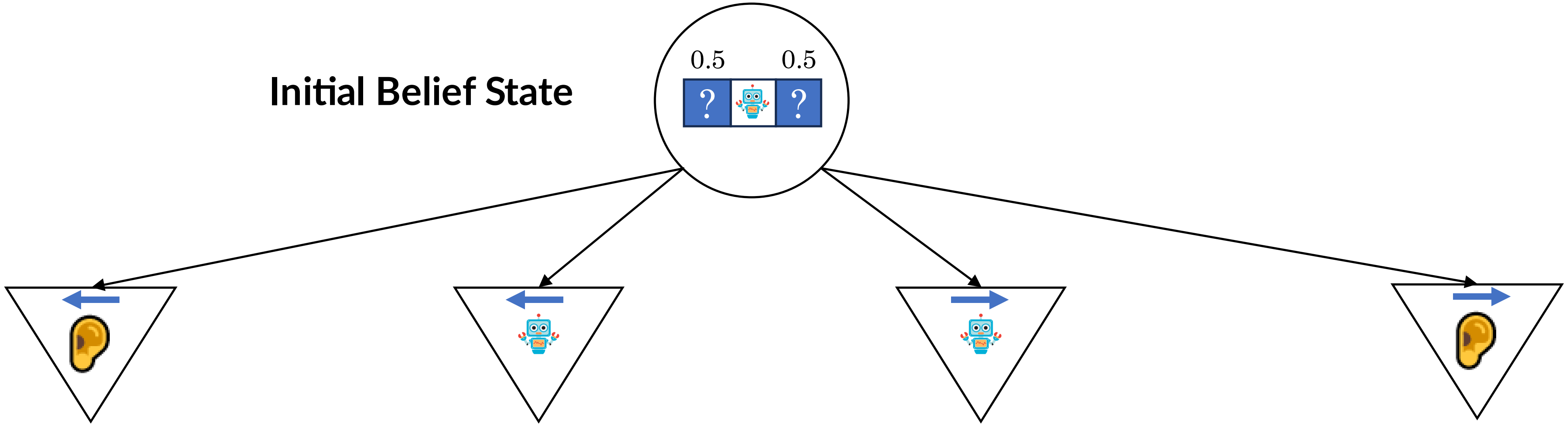


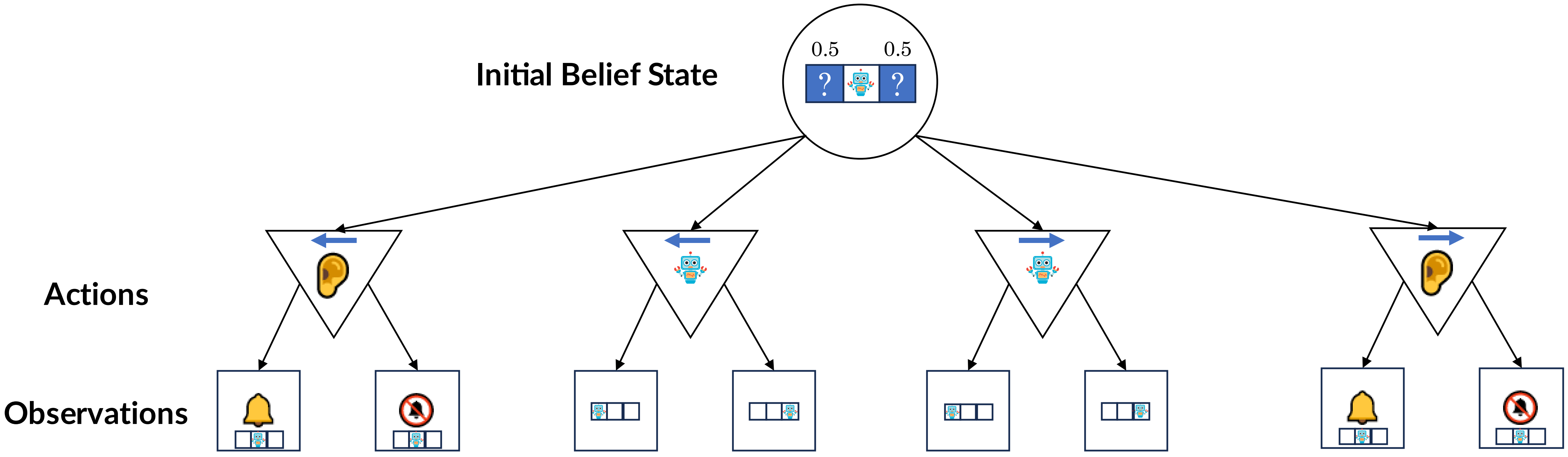
Initial Belief State

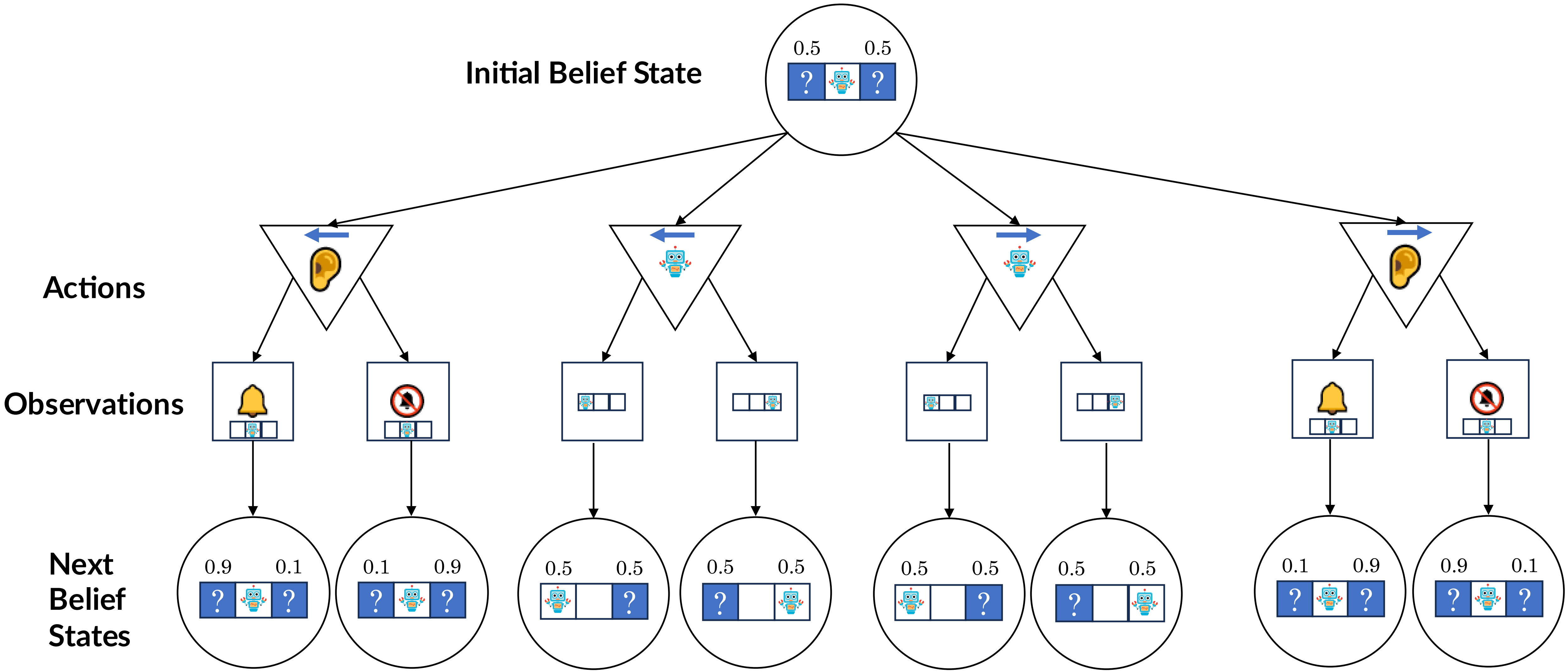


Initial Belief State

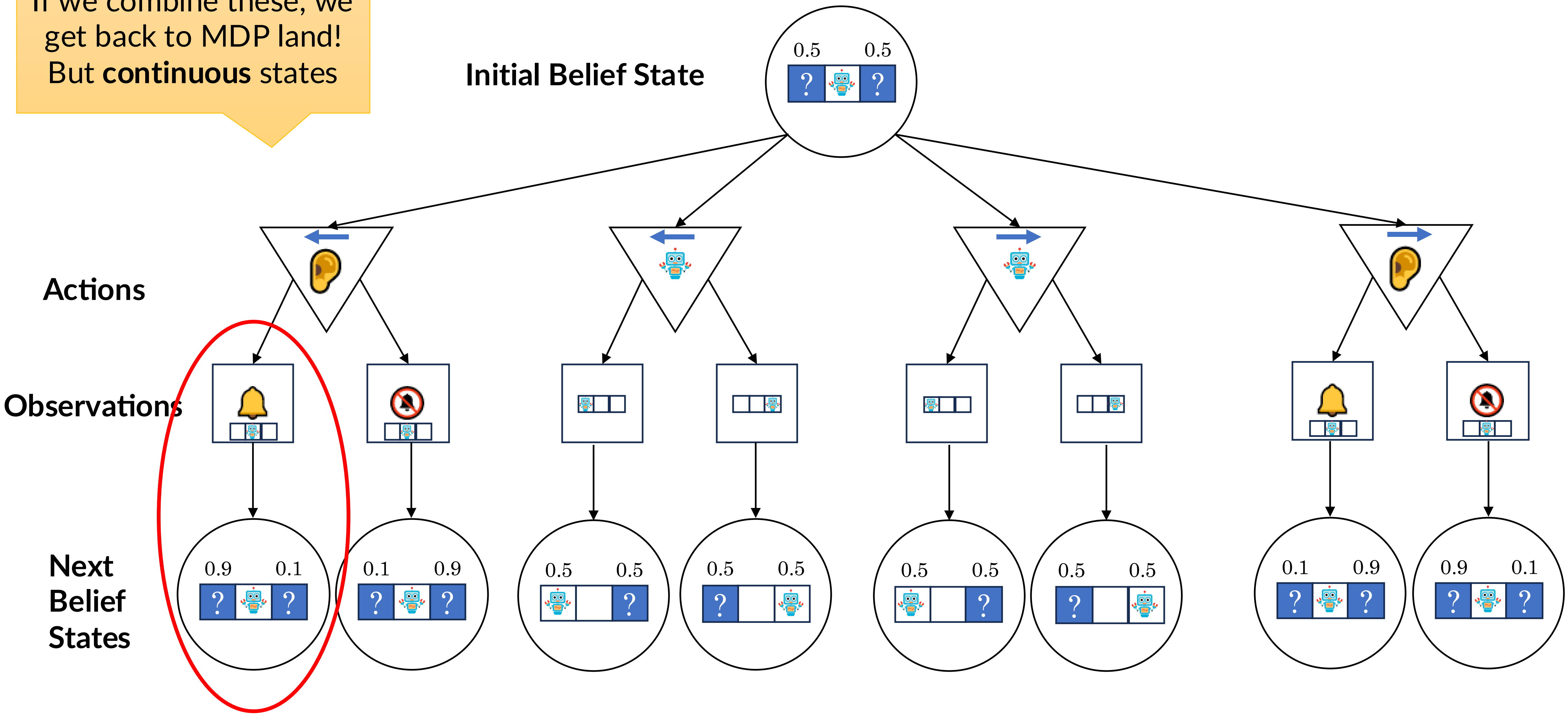
Actions




Initial Belief State
0.5
0.5
?
?
Actions
Observations

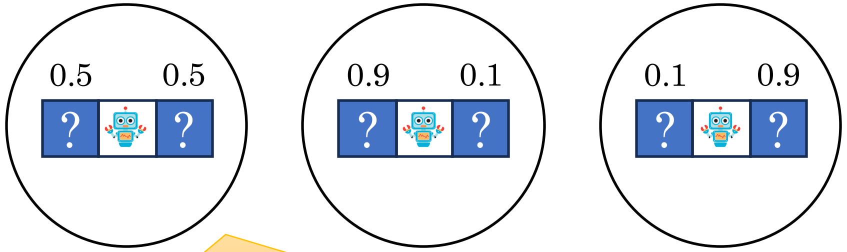

Initial Belief State
0.5
0.5
?
?
Actions
Observations
Next Belief States
0.9
0.1
0.1
0.9
0.5
0.5
0.5
0.5
0.5
0.5
0.5
0.5
0.1
0.9
0.9
0.1

If we combine these, we get back to MDP land! But **continuous** states



POMDP → Belief MDP

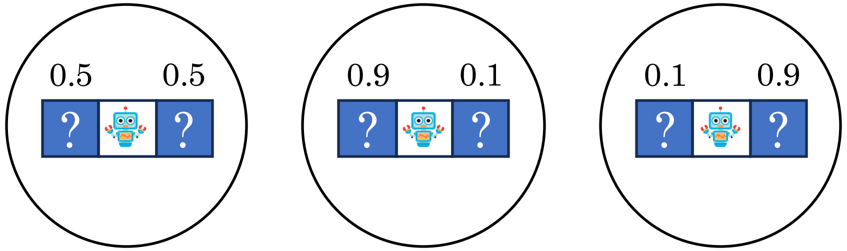
- **State space:** ***beliefs***



Three example states in the belief MDP

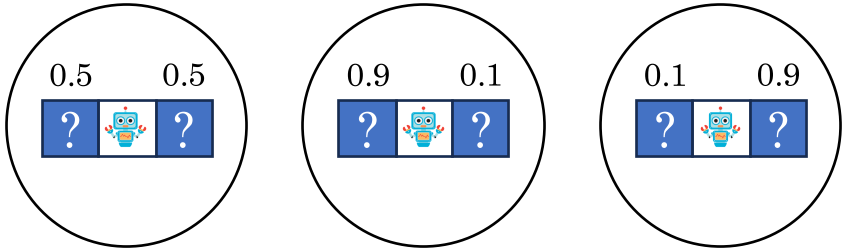
POMDP → Belief MDP

- **State space:** *beliefs*
- **Action space:** same



POMDP → Belief MDP

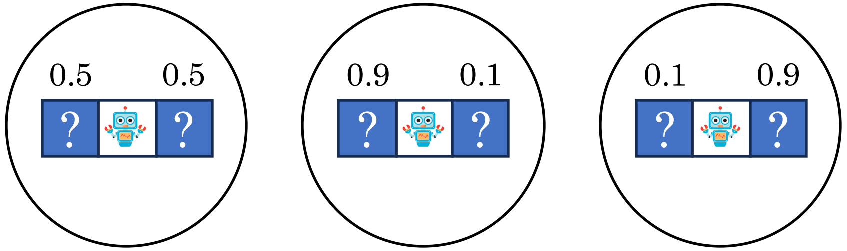
- **State space:** ***beliefs***
- **Action space:** same
- **Rewards:**



$$R(b_t, a_t) = \sum_{s_t} b_t(s_t) \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) R(s_t, a_t, s_{t+1})$$

POMDP → Belief MDP

- **State space:** *beliefs*
- **Action space:** same
- **Rewards:**



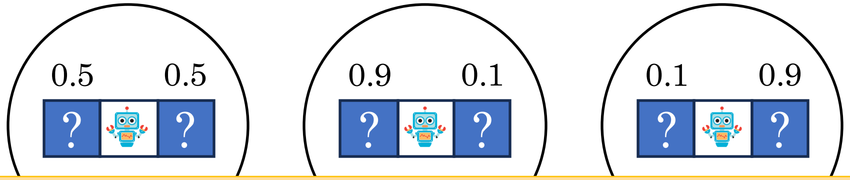
$$R(b_t, a_t) = \sum_{s_t} b_t(s_t) \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) R(s_t, a_t, s_{t+1})$$

- **Transitions:**

$$P(b_{t+1} \mid b_t, a_t) = \ \ldots$$

POMDP → Belief MDP

- **State space:** ***beliefs***



1. We're in state s_t with probability $b_t(s_t)$ and we take a_t.
2. Transition to s_{t+1} by sampling from $P(s_{t+1}|s_t, a_t)$.
3. We receive some o_{t+1} sampled from $P(o_{t+1}|a_t, s_{t+1})$.
4. We run state estimation to compute $b_{t+1}(s_{t+1})$.

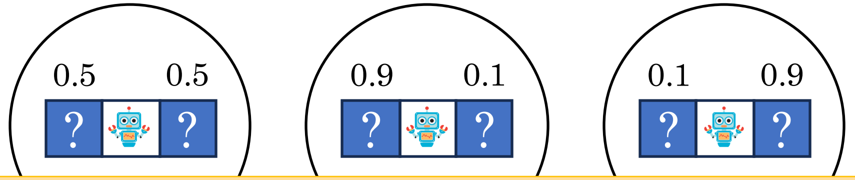
$a_t, s_{t+1})$

- **Transitions:**

$$P(b_{t+1} \mid b_t, a_t) = \ \ldots$$

POMDP → Belief MDP

- **State space:** ***beliefs***



1. We're in state s_t with probability $b_t(s_t)$ and we take a_t.
2. Transition to s_{t+1} by sampling from $P(s_{t+1}|s_t, a_t)$.
3. We receive some o_{t+1} sampled from $P(o_{t+1}|a_t, s_{t+1})$.
4. We run state estimation to compute $b_{t+1}(s_{t+1})$.

For each possible observation o_{t+1}, there is one next belief b_{t+1}

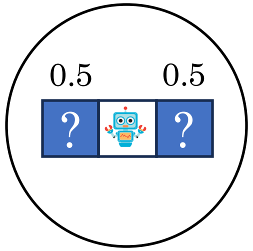
- **Transitions:**

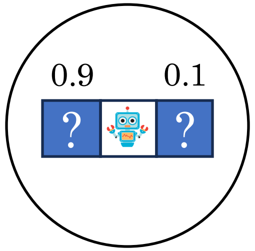
$$P(b_{t+1} \mid b_t, a_t) = \ \dots$$

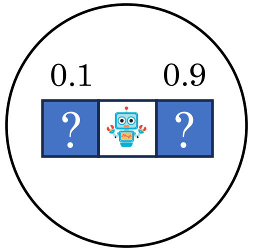
Notation:
$b_{t+1} = SE(b_t, a_t, o_{t+1})$

POMDP → Belief MDP

- **State space:** *beliefs*
- **Action space:** same







- **Rewards:**

$$R(b_t, a_t) = \sum_{s_t} b_t(s_t) \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) R(s_t, a_t, s_{t+1})$$

- **Transitions:**

What is this?

$$P(b_{t+1} \mid b_t, a_t) = \begin{cases} SE(b_t, a_t, o_{t+1}) & \text{w.p.} \;\; P(O_{t+1} = o_{t+1} | b_t, a_t) \\ SE(b_t, a_t, o'_{t+1}) & \text{w.p.} \;\; P(O_{t+1} = o'_{t+1} | b_t, a_t) \\ & \dots \end{cases}$$

POMDP → Belief MDP

Final equations to complete transition distribution:

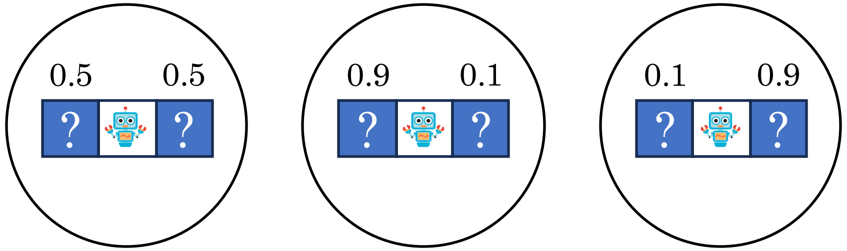
$$SE(b_t, a_t, o_{t+1})(s_{t+1}) \propto P(o_{t+1} | \, a_t, s_{t+1}) \sum_{s_t} b_t(s_t) P(s_{t+1} | s_t, a_t)$$

Same as “state estimation” slide

$$P(O_{t+1} = o_{t+1} | b_t, a_t) = \sum_{s_t} b_t(s_t) \sum_{s_{t+1}} P(o_{t+1} | \, a_t, s_{t+1}) \, P(s_{t+1} | s_t, a_t)$$

POMDP → Belief MDP

- **State space:** *beliefs*
- **Action space:** same



- **Rewards:**

$$R(b_t, a_t) = \sum_{s_t} b_t(s_t) \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) R(s_t, a_t, s_{t+1})$$

- **Transitions:**

$$P(b_{t+1} \mid b_t, a_t) = \begin{cases} SE(b_t, a_t, o_{t+1}) & \text{w.p.} \;\; P(O_{t+1} = o_{t+1} | b_t, a_t) \\ SE(b_t, a_t, o'_{t+1}) & \text{w.p.} \;\; P(O_{t+1} = o'_{t+1} | b_t, a_t) \\ & \dots \end{cases}$$

- **Horizon:** if finite or infinite, same. Indefinite: convert to infinite. (Why?)

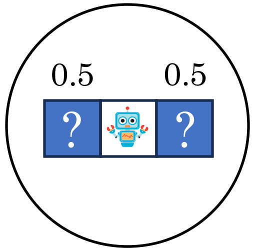
https://github.com/rpmml/rpmml-code/blob/main/scripts/treasure_hunt_pomdp_walkthrough.py

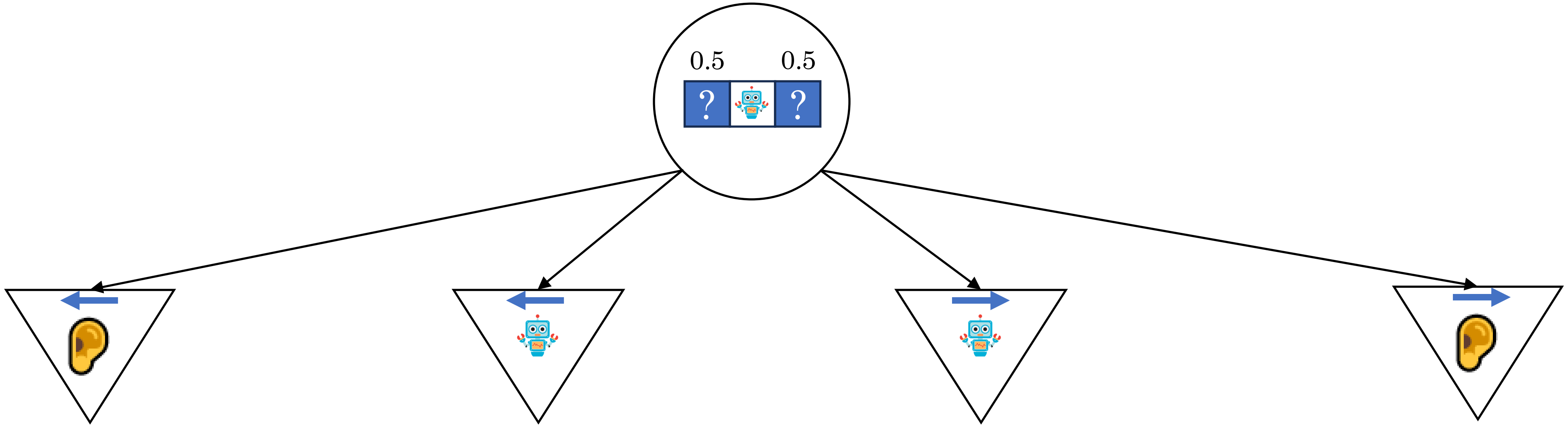
POMDP Expectimax Search

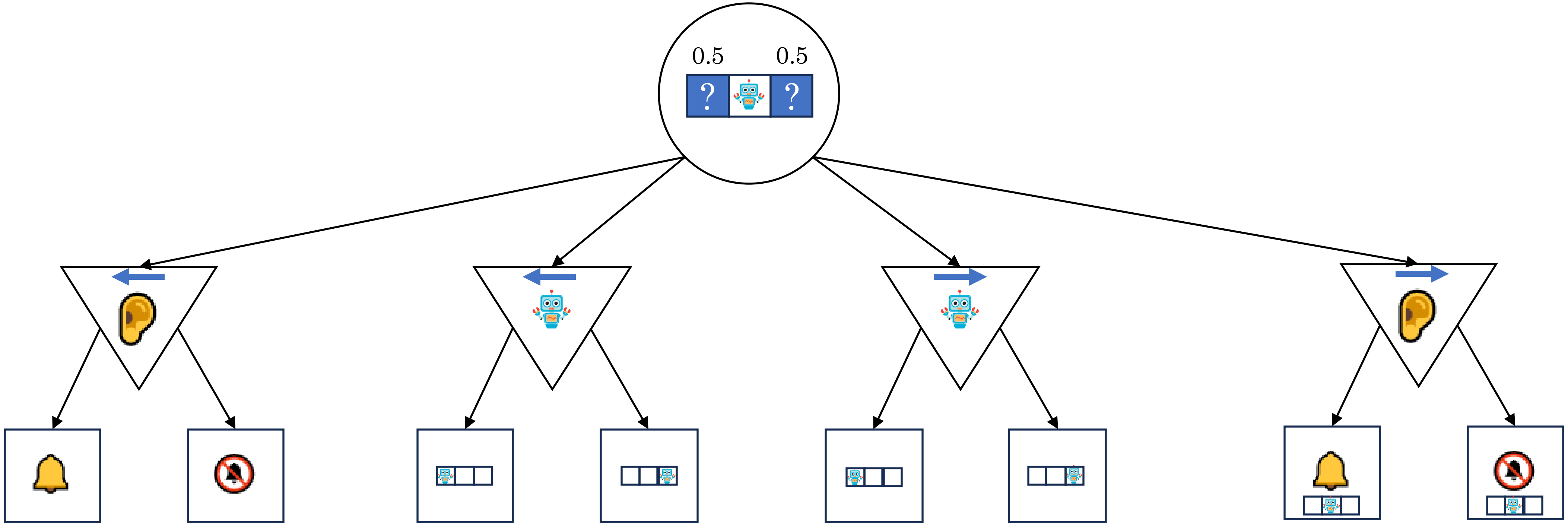
- Run expectimax search in the belief MDP.

  …

- That's pretty much it!

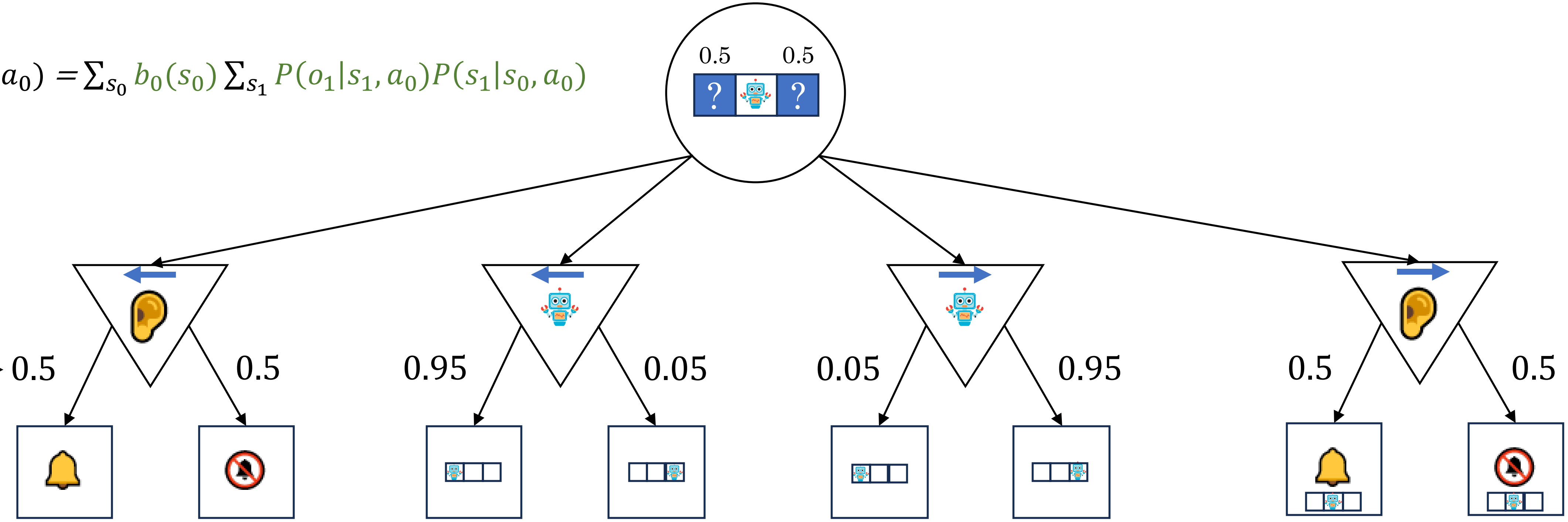

0.5
0.5
?
?

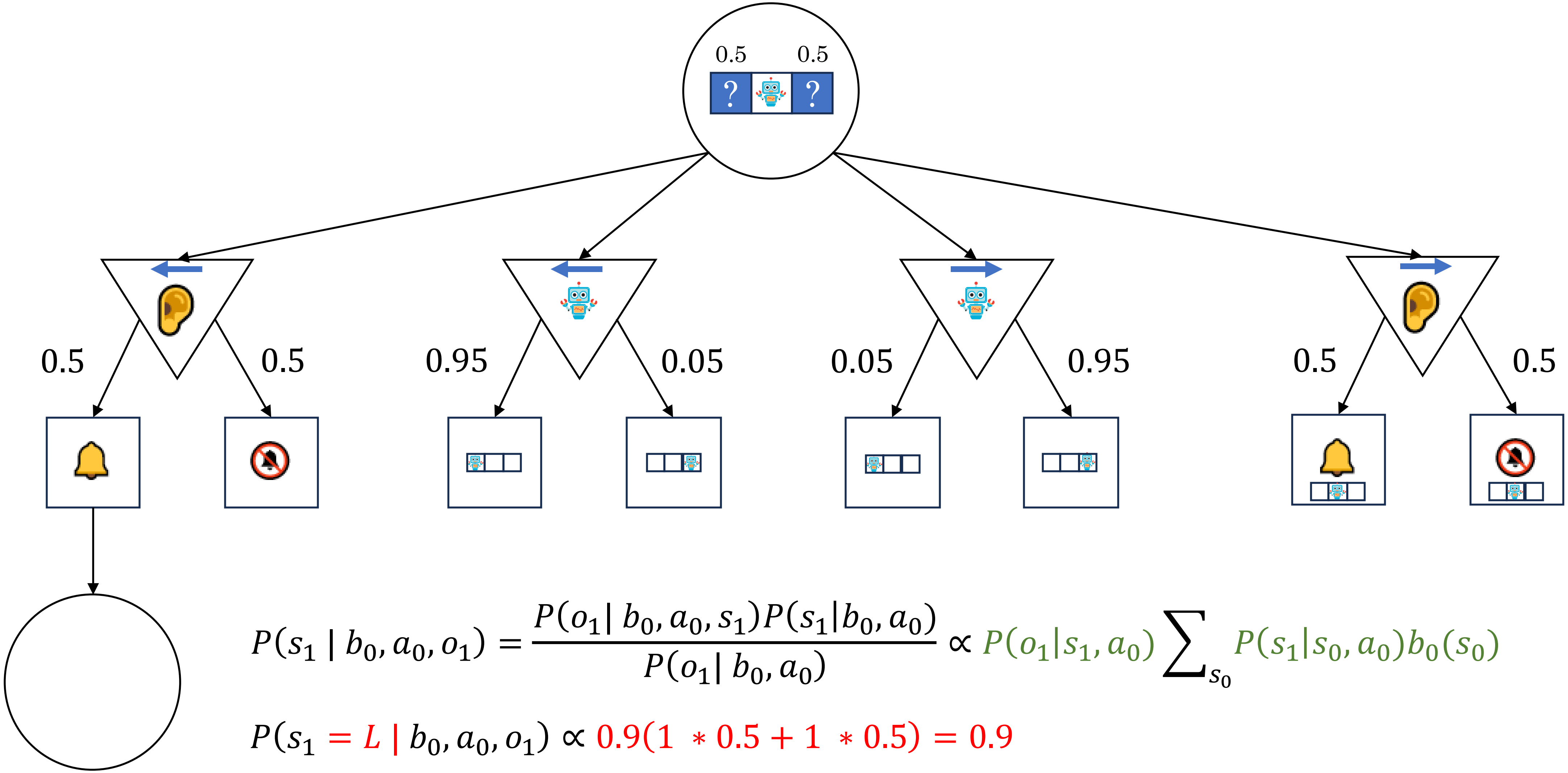

0.5
0.5
?
?



0.5
0.5
?
?

$P(o_1|b_0, a_0) = \sum_{s_0} b_0(s_0) \sum_{s_1} P(o_1|s_1, a_0)P(s_1|s_0, a_0)$

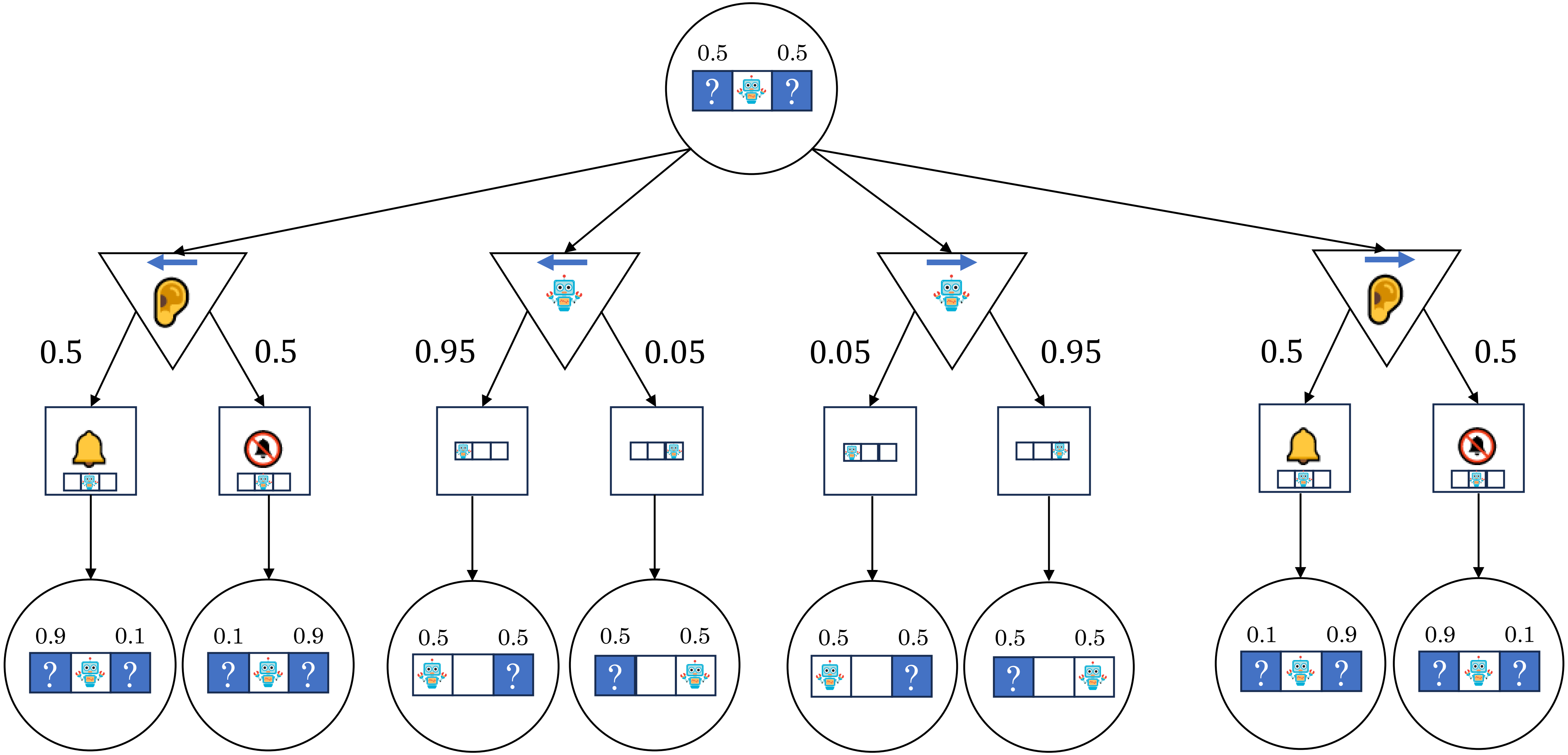




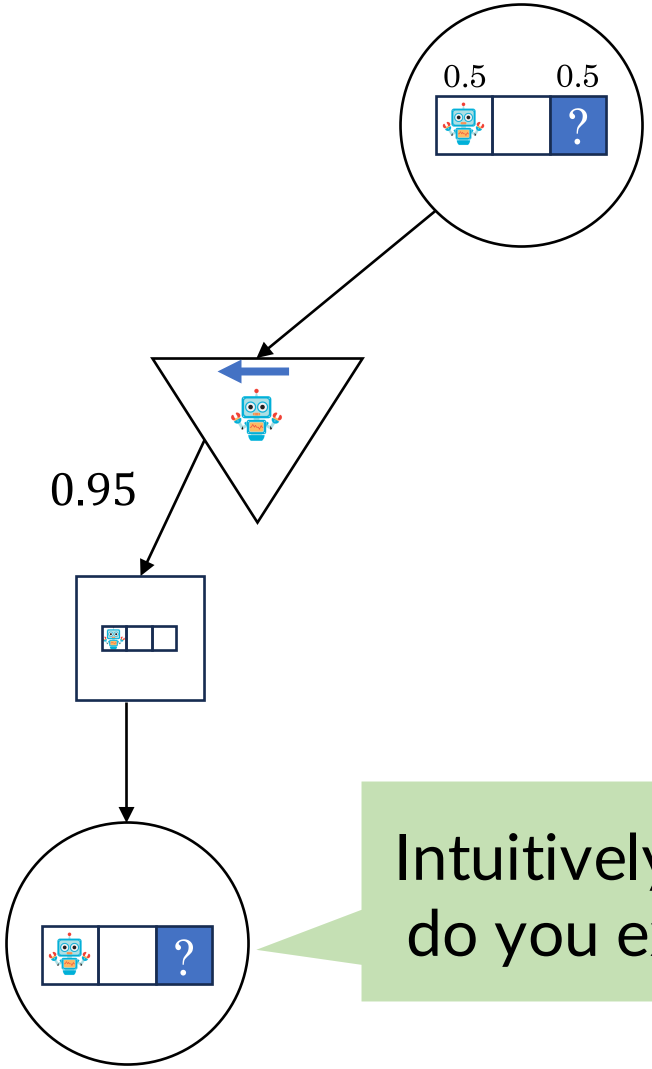
$$P(s_1 \mid b_0, a_0, o_1) = \frac{P(o_1 \mid b_0, a_0, s_1) P(s_1 \mid b_0, a_0)}{P(o_1 \mid b_0, a_0)} \propto P(o_1 \mid s_1, a_0) \sum_{s_0} P(s_1 \mid s_0, a_0) b_0(s_0)$$

$$P(s_1 = L \mid b_0, a_0, o_1) \propto 0.9(1 * 0.5 + 1 * 0.5) = 0.9$$

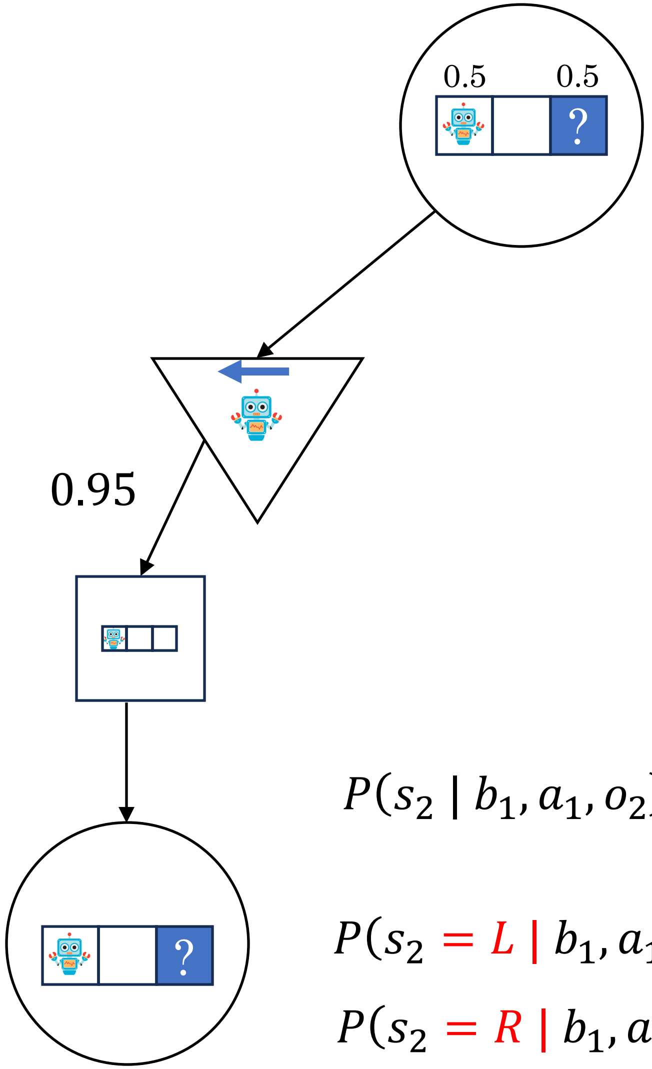
$$P(s_1 = R \mid b_0, a_0, o_1) \propto 0.1(1 * 0.5 + 1 * 0.5) = 0.1$$



0.5 0.5
?
?
0.5
0.5
0.95
0.05
0.05
0.95
0.5
0.5
0.9 0.1
0.1 0.9
0.5 0.5
0.5 0.5
0.5 0.5
0.5 0.5
0.1 0.9
0.9 0.1



Intuitively, what do you expect?



$$P(s_2 \mid b_1, a_1, o_2) \propto P(o_2|s_2, a_1) \sum_{s_1} P(s_2|s_1, a_1) b_1(s_1)$$

$$P(s_2 = L \mid b_1, a_1, o_2) \propto 1.0 * (1.0 * 0.5) = 0.5$$

$$P(s_2 = R \mid b_1, a_1, o_2) \propto 1.0 * (0.95 * 0.5) = 0.475$$

Completing the Expectimax Agent

1. Receive initial observation

2. Initialize belief

3. Repeat:
    1. Run expectimax search in belief MDP
    2. Execute action and receive observation
    3. Run state estimation to update belief

Example: Home Inspection

Solve!

- **States**: *home-value* in {100, 300}
- **Actions:** *buy*, *do-not-buy*, *cheap-inspect*, *premium-inspect*
- **Observations:** *good-deal* or *bad-deal* or *none*
- **Rewards:**
  - +$100 if buy and home-value=300
  - -$100 if buy and home-value=100
  - -$10 to cheap-inspect
  - -$50 to premium-inspect
- **Horizon:** indefinite (terminate after buy or do-not-buy)
- **Transition distribution:**
  - There is a 5% chance that inspecting will make home-value 100
- **Observation model:**
  - Cheap-inspect is 75% accurate
  - Premium-inspect is 90% accurate

Planning in POMDPs

<u>Offline Planning</u>

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

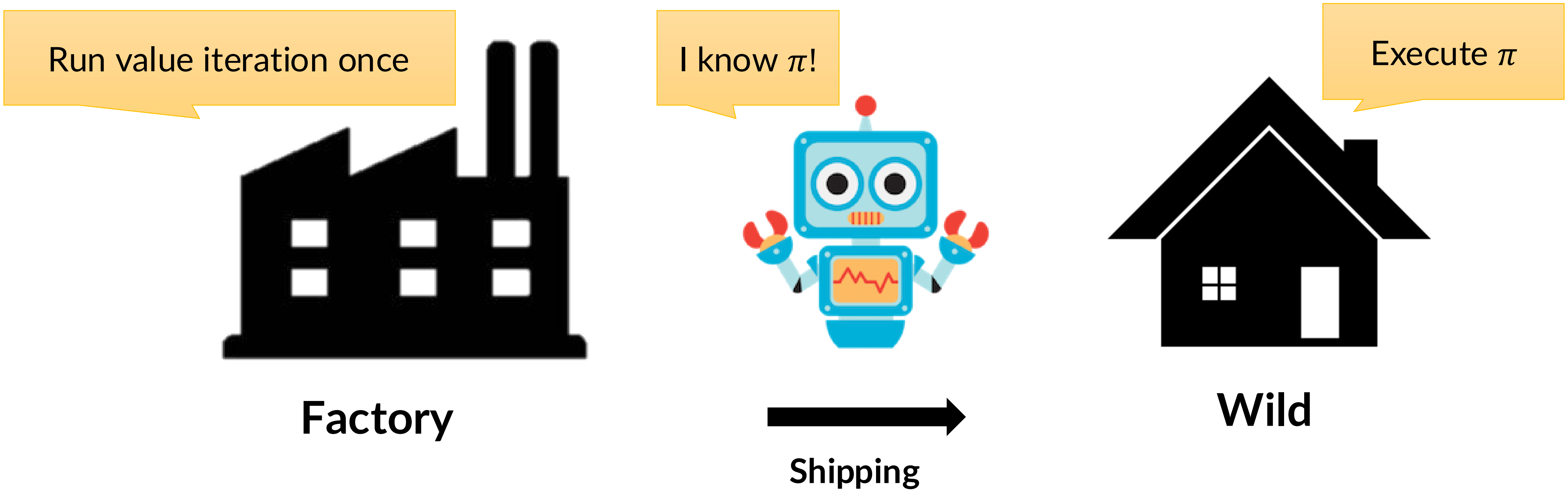
Now this

<u>Online Planning</u>

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

POMDP Planning Offline (In the Factory)

Run POMDP value iteration **offline** (in the factory) and compute π.



A Stupidest Possible Algorithm

$$\pi_{MLS}(b) \triangleq \pi^*_{\mathrm{MDP}}\left(\mathrm{argmax}_s\ b(s)\right)$$

where

- MLS = "Most Likely State" approximation
- π^*_{MDP} is an optimal policy for the underlying MDP

Why is this "stupid"?

Another Stupidest Possible Algorithm

- Enumerate candidate policies

- Evaluate each candidate and keep the best

Another Stupidest Possible Algorithm

- Enumerate candidate policies

> Wait, how? Aren't these continuous functions?

- Evaluate each candidate and keep the best

Policy Trees

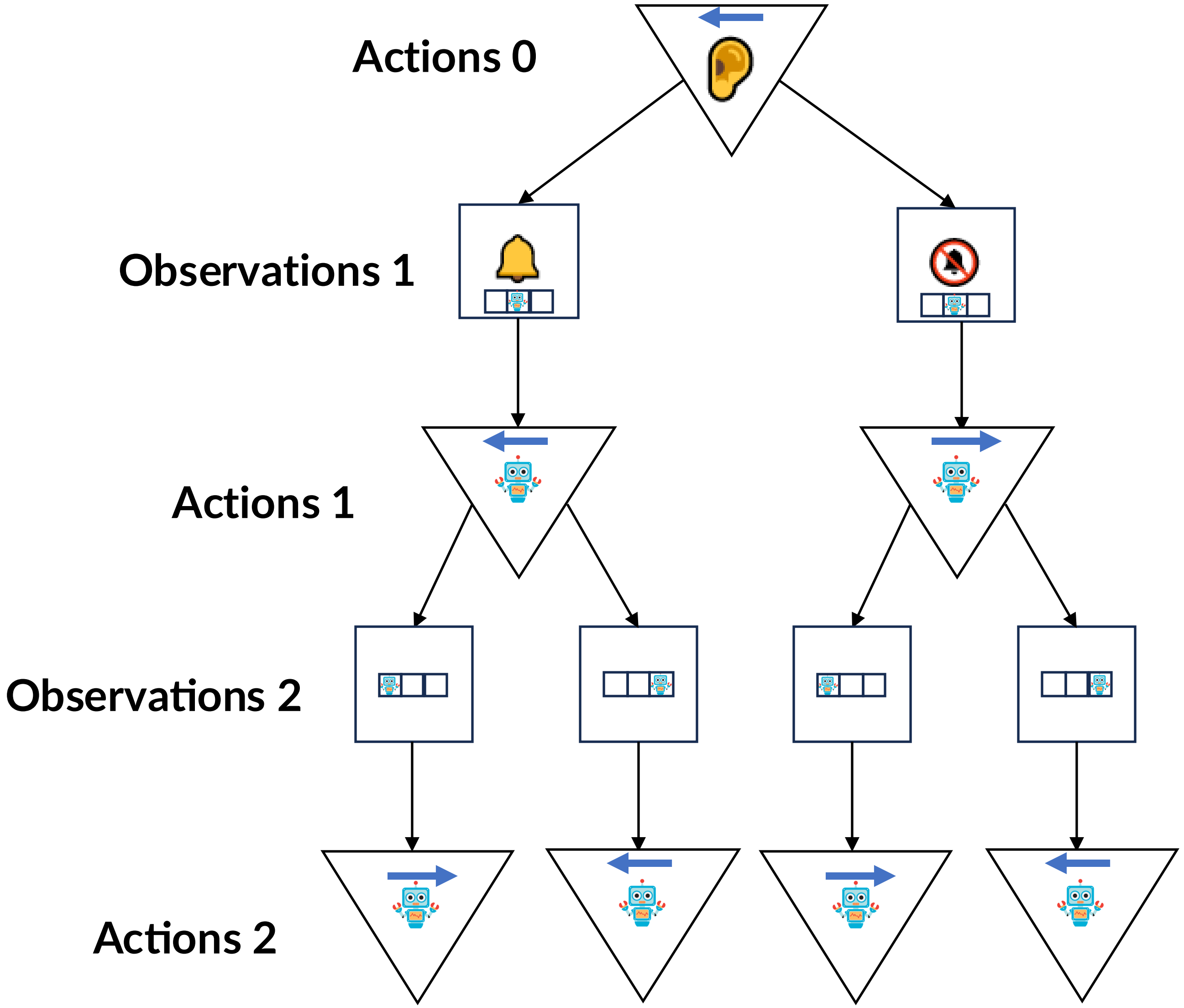
a.k.a. *conditional plans*

Repeat:

- Take root action
- Receive observation
- Child is new root

For finite horizon, there are finitely many policy trees

How many exactly?



Another Stupidest Possible Algorithm

- Enumerate candidate policies

Notation:
1. Γ is a policy tree
2. $[\Gamma]$ is the action at the root
3. $\Gamma[o]$ is the new policy tree after receiving observation o

- Evaluate each candidate and keep the best

Another Stupidest Possible Algorithm

- Enumerate candidate policies

Notation:
1. Γ is a policy tree
2. $[\Gamma]$ is the action at the root
3. $\Gamma[o]$ is the new policy tree after receiving observation o

- Evaluate each candidate and keep the best

How?

POMDP Policy Tree Evaluation

Finite Horizon

$$V_t^{\Gamma}(s) = \sum_{s'} P(s' \mid s, [\Gamma])[R(s, [\Gamma], s') + \sum_{o} O(o \mid s', [\Gamma])\, V_{t+1}^{\Gamma[o]}(s')]$$

$$V_H^{\Gamma}(s) = 0$$

$$V_t^{\Gamma}(b) = \sum_s b(s)\, V_t^{\Gamma}(s)$$

POMDP Policy Tree Evaluation

Finite Horizon

$$V_t^{\Gamma}(s) = \sum_{s'} P(s' \mid s, [\Gamma])[R(s, [\Gamma], s') + \sum_{o} O(o \mid s', [\Gamma]) \, V_{t+1}^{\Gamma[o]}(s')]$$

$$V_H^{\Gamma}(s) = 0$$

$$V_t^{\Gamma}(b) = \sum_s \; b(s) \, V_t^{\Gamma}(s)$$

This is a linear function of *b*!

POMDP Policy Tree Evaluation

Finite Horizon

$$V_t^{\Gamma}(s) = \sum_{s'} P(s' \mid s, [\Gamma])[R(s, [\Gamma], s') + \sum_{o} O(o \mid s', [\Gamma]) \, V_{t+1}^{\Gamma[o]}(s')]$$

$$V_H^{\Gamma}(s) = 0$$

$$V_t^{\Gamma}(b) = \sum_s \, b(s) \, V_t^{\Gamma}(s) \triangleq \boldsymbol{\alpha}_{\Gamma} \cdot \boldsymbol{b}$$

"Alpha vector"

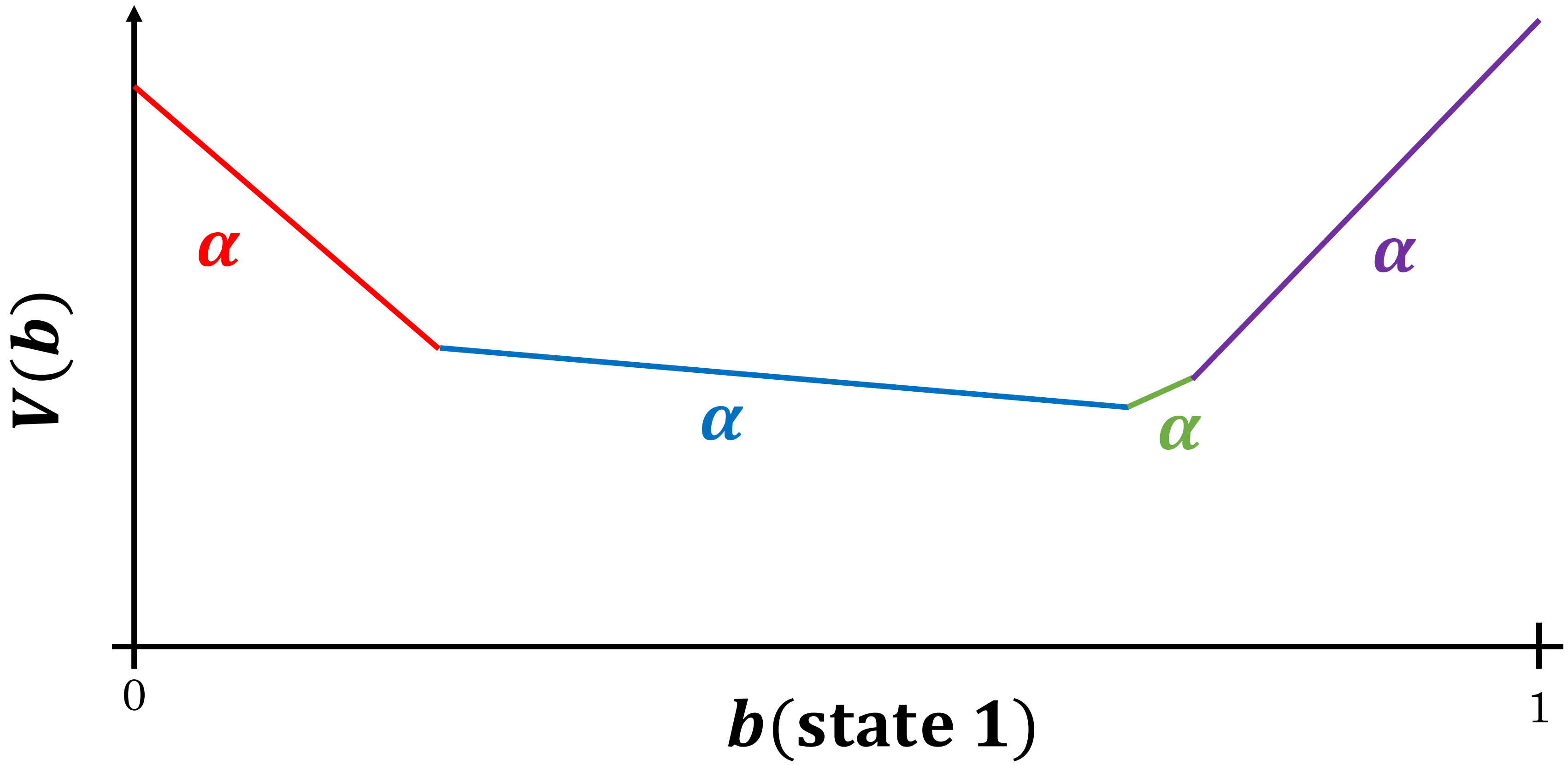
POMP Value Functions Have Special Structure

$$V^*(\boldsymbol{b}) = \max_{\Gamma} \boldsymbol{\alpha}_{\Gamma} \cdot \boldsymbol{b}$$

The optimal value function is *piecewise linear*

It's also *convex* (more certainty → higher value)

Example: 2-State POMDP

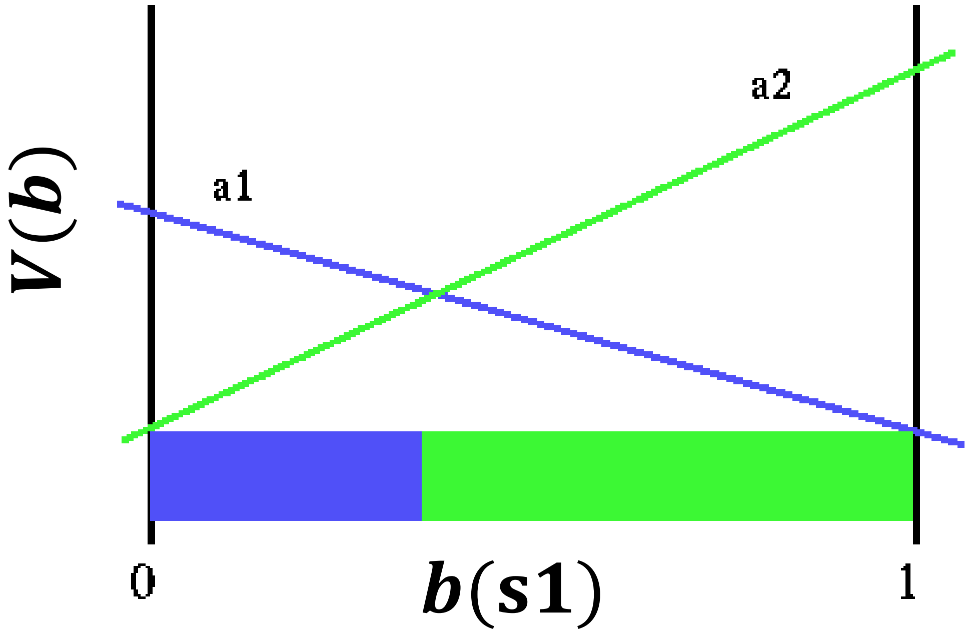


Example: Toy POMDP

- **States**: *s1*, *s2*
- **Actions:** *a1*, *a2*
- **Observations:** *z1*, *z2*, *z3*
- **Rewards:**
  - R(s1, a1) = 0
  - R(s1, a2) = 1.5
  - R(s2, a1) = 1
  - R(s2, a2) = 0

Horizon = 1

Suppose: $\mathrm{b}_0 = [0.75, 0.25]$



https://pomdp.org/tutorial/pomdp-vi-example.html

Recall: Value Iteration in MDPs

ValueIteration$(\mathcal{S}, \mathcal{A}, P, R, \gamma)$

// Represent values as dictionary $\texttt{V[s]} = V^*(s)$, initialized arbitrarily.
while not converged
    // Initialize new value function dictionary
    $\texttt{Vn} = \texttt{dict()}$
    for each $\texttt{s} \in \mathcal{S}$
        $\texttt{Vn[s]} = \text{BellmanBackup}(\texttt{s}, \texttt{V}, \mathcal{S}, \mathcal{A}, P, R, \gamma)$
    $\texttt{V} = \texttt{Vn}$
return V

Iteration 1: Horizon 1 (rewards only)
Iteration 2: Horizon 2 in terms of Horizon 1

Extending This Intuition To POMDPs

First, compute values for horizon = 1

- Policy trees are just actions
- Values are just rewards

Then, compute values for horizon = 2 using horizon = 1!

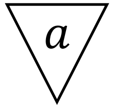
- Policy trees are depth 2
- Values are rewards + horizon 1 values

But how?

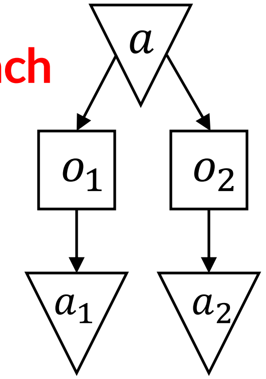
Etc…

Value Iteration Intuition

$$\alpha^0 = (0, \ldots, 0)$$

$$\alpha_a^1 = (R(s_1, a), \ldots, R(s_n, a))$$ **For each** 

Linear transform

$$\alpha_\Gamma^2 = \big(R(s_1, a), \ldots, R(s_n, a)\big) + M_{ao_1}\left(\alpha_{a_1}^1\right) + M_{ao_2}\left(\alpha_{a_2}^1\right)$$ **For each** 

Using previous step

…

Guaranteed to converge

Can prune *dominated* vectors after each step

More Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- **Policy iteration for POMDPs**
- See: PBVI, Witness Algorithm

Need to be careful about policy representation and evaluation

Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

More Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: **PBVI**, Witness Algorithm

Pineau, Gordon, Thrun (2003)

Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

More Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, **Witness Algorithm**

Littman et al. (1994)

Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

More Planning in POMDPs

<u>Offline Planning</u>

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

<u>Online Planning</u>

- Expectimax search for POMDPs
- **Sparse sampling for large transitions**
- Bandits / MCTS for smarter exploration
- See: POMCP, DESPOT

Key idea: need *sparse belief updates* too

More Planning in POMDPs

<u>Offline Planning</u>

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

<u>Online Planning</u>

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- **Bandits / MCTS for smarter exploration**
- See: POMCP, DESPOT

Key idea: store *histories* in nodes, rather than just *states*

More Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: **POMCP**, DESPOT

Silver & Veness (2010)

More Planning in POMDPs

Offline Planning

- Value iteration for POMDPs
- Policy iteration for POMDPs
- See: PBVI, Witness Algorithm

Online Planning

- Expectimax search for POMDPs
- Sparse sampling for large transitions
- Bandits / MCTS for smarter exploration
- See: POMCP, **DESPOT**

Somani et al. (2013)

POMDP Planning: **Takeaways**

1. POMDPs are **hard**

2. POMDPs → **continuous-state MDPs** with structure

3. Litmus test for candidate planners: **information-gathering**

For more, highly recommend: https://pomdp.org/