Partially Observable Markov Decision Processes (POMDP)

Content

- Introduction to the problem
- Convert to MDP: History Solutions
  - Window
  - UDM
  - Active Memorization
- Solve POMDPs
  - Belief States & Infinite-State MDP
  - Value Function of POMDP

Definition – MDP

- A Markov decision process is a tuple

$$< S, A, T, R >$$

- S – a finite set of states of the world
- A – a finite set of actions
- T: S×A $\rightarrow \Pi$(S) – state-transition function

$$T(s, a, s') = P(s_{t+1} = s' \mid s_t = s, a_t = a)$$

- R: S×A $\rightarrow \mathbb{R}$ – the reward function

Complete Observability

- Solution procedures for MDPs give values or policies for each state.
- Use of these solutions requires that the agent is able to detect the state it is currently in with complete reliability.

- Therefore, it is called CO-MDP (completely observable)

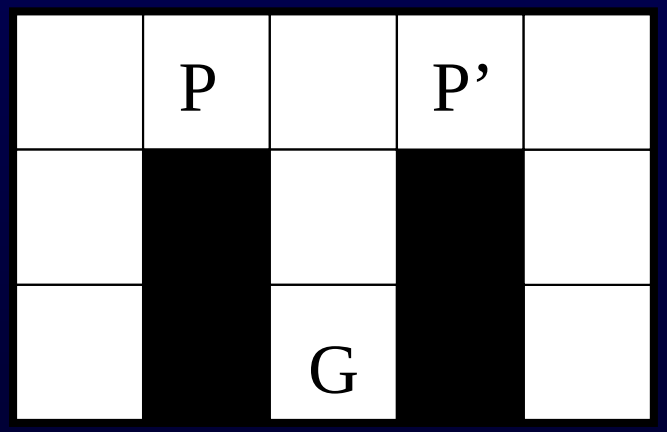
Complete perception

- In most cases perfect perception is not realistic
  - Agents with “local sensors” ...
  - ... in conjuntion with lack of memory
  - Noisy sensors
  - ...

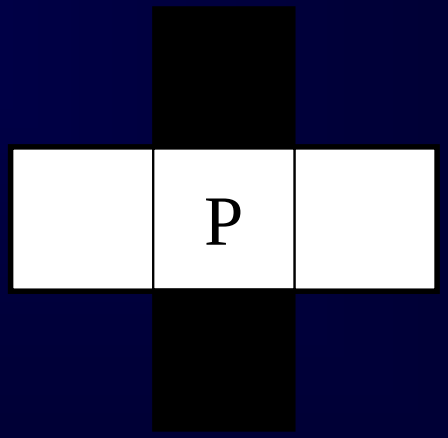
Perceptual aliasing problem

- The agent is aware only of partial information about the current state
- Typical example:

Complete information

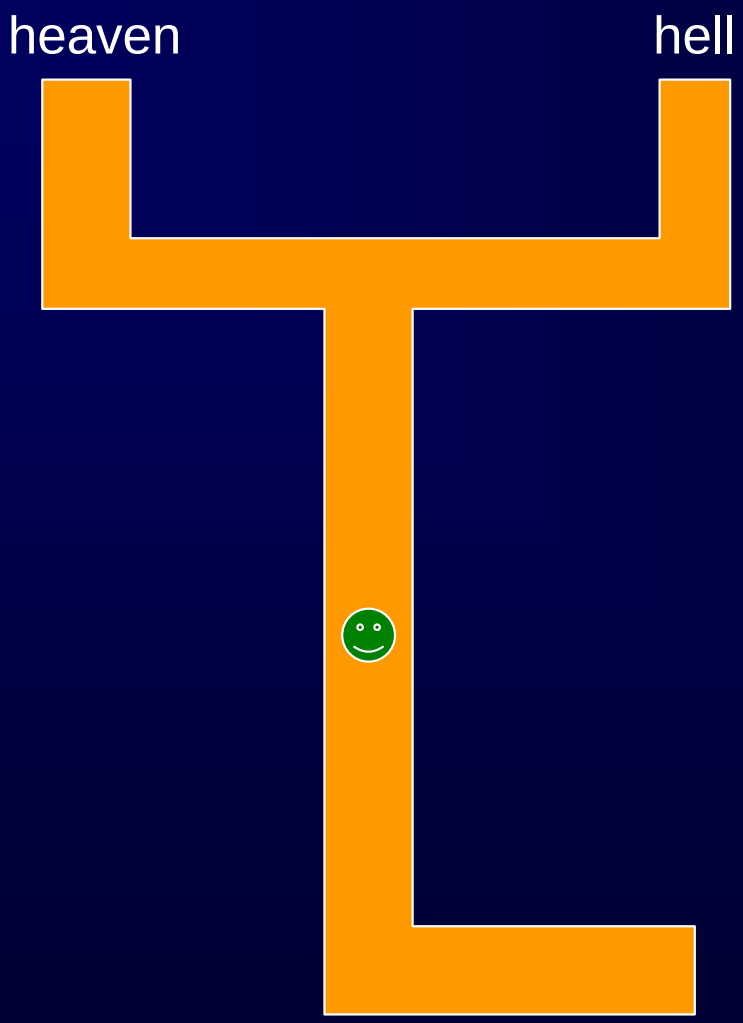


Partial information



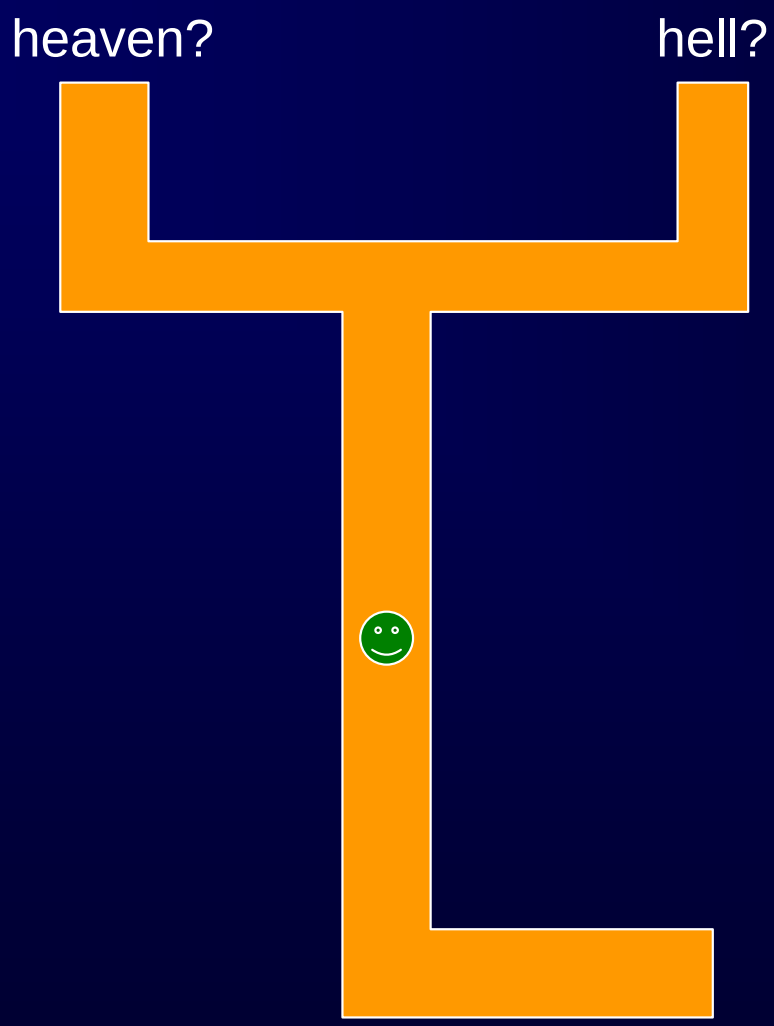
States P and P’ are aliased into a single perceptual state

Stochastic, Partially Observable

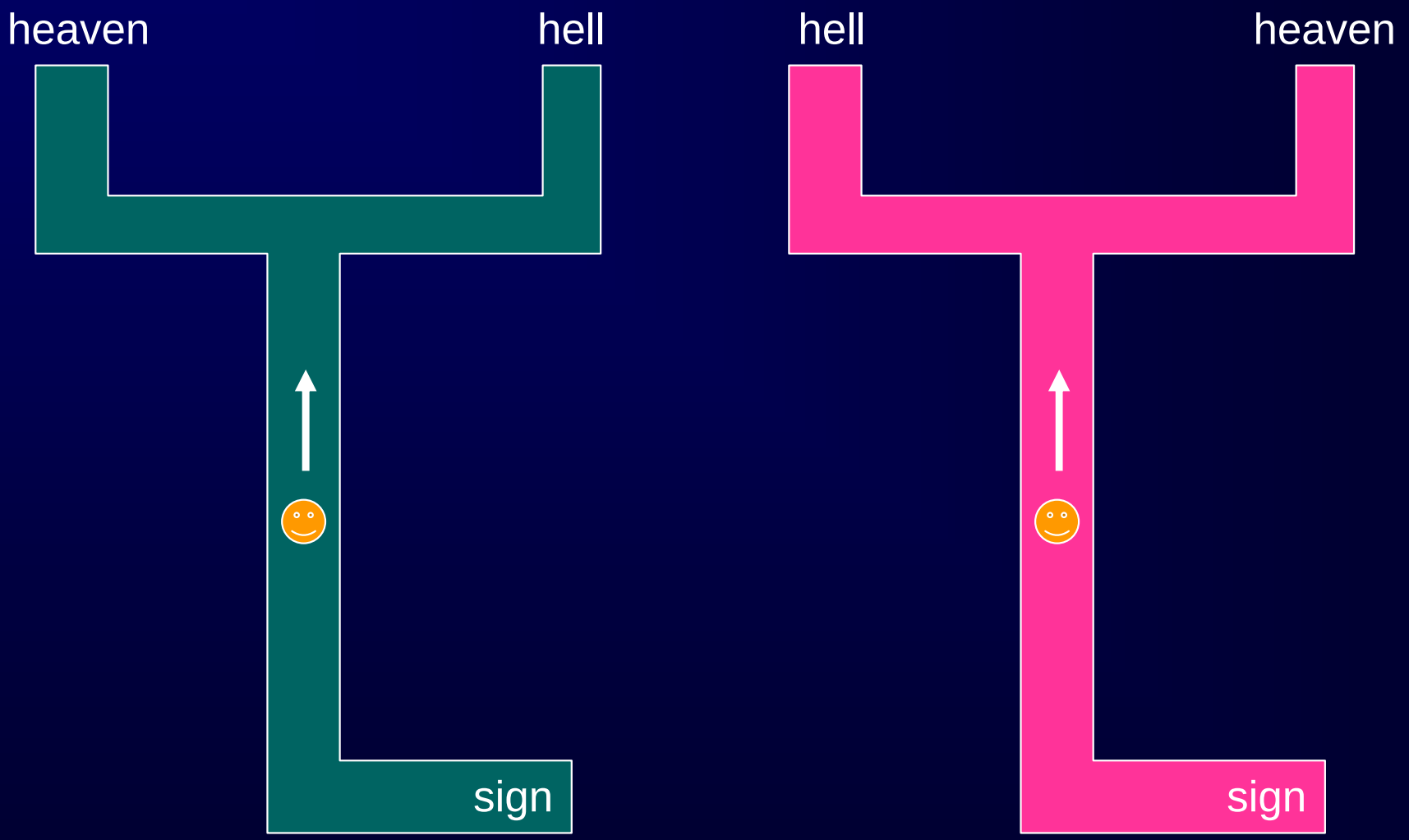


[Sondik 72] [Littman/Cassandra/Kaelbling 97]

Stochastic, Partially Observable

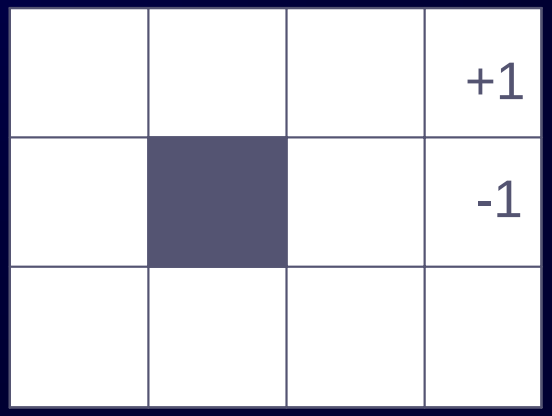


Stochastic, Partially Observable



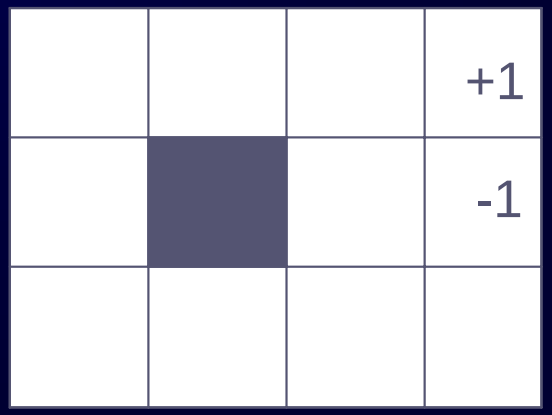
Example

- In our 4x3 environment:
- Agent don’t know where it is
  - Has no sensors
  - Being put in unknown state
- What should it do?
  - If has know it in (3,3) it would do Right action
  - But it doesn’t know

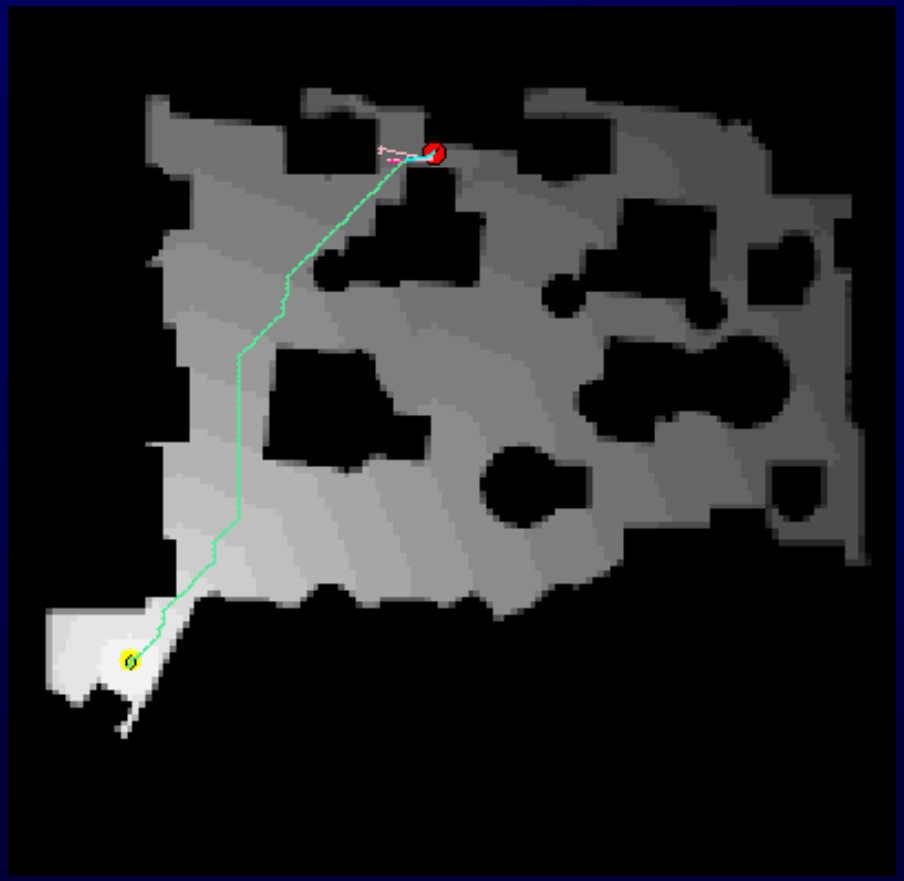


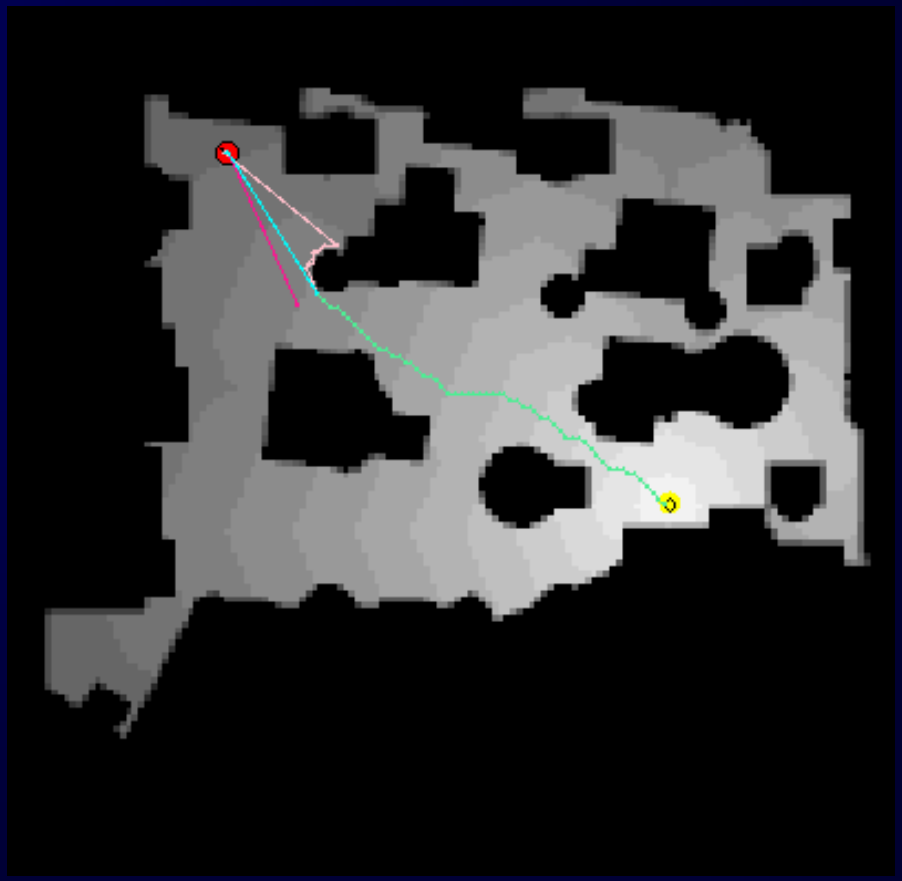
solutions

- First solution:
  - Step 1: reduces uncertainty
  - Step 2: try heading to the +1 exit
- Reduce uncertainty
  - move 5 times *Left* so quite likely to be at the left wall
  - Then move 5 times *Up* so quite likely to be at left top wall
  - Then move 5 time *Right* to goal state
  - Continue moving right increase chance to get to +1
- Quality of solution
  - Chance of 81.8% to get to +1
  - Expected utility of about 0.08
- Second solution: POMDP



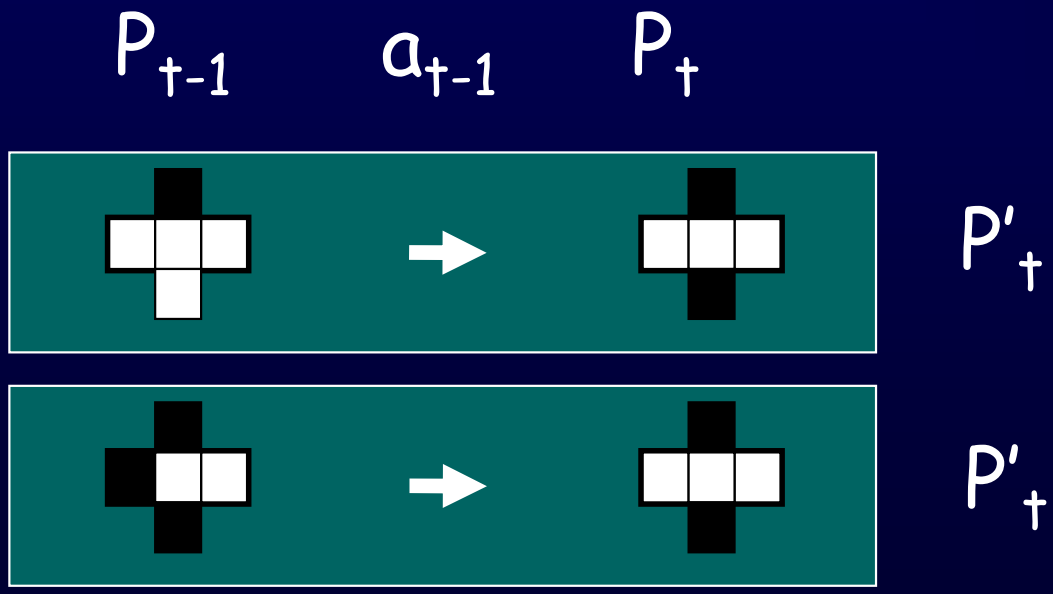
Robot Motion Planning

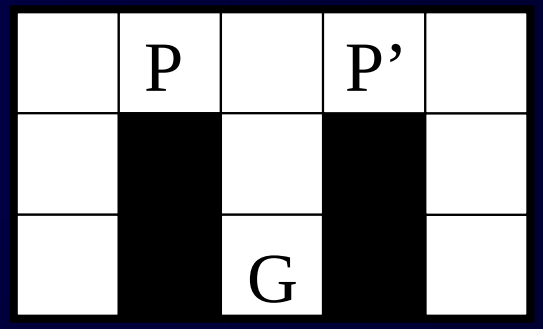




First solution

- Resolve current state by using memory of past perceptions and actions:





Window length 2 states, 1 action

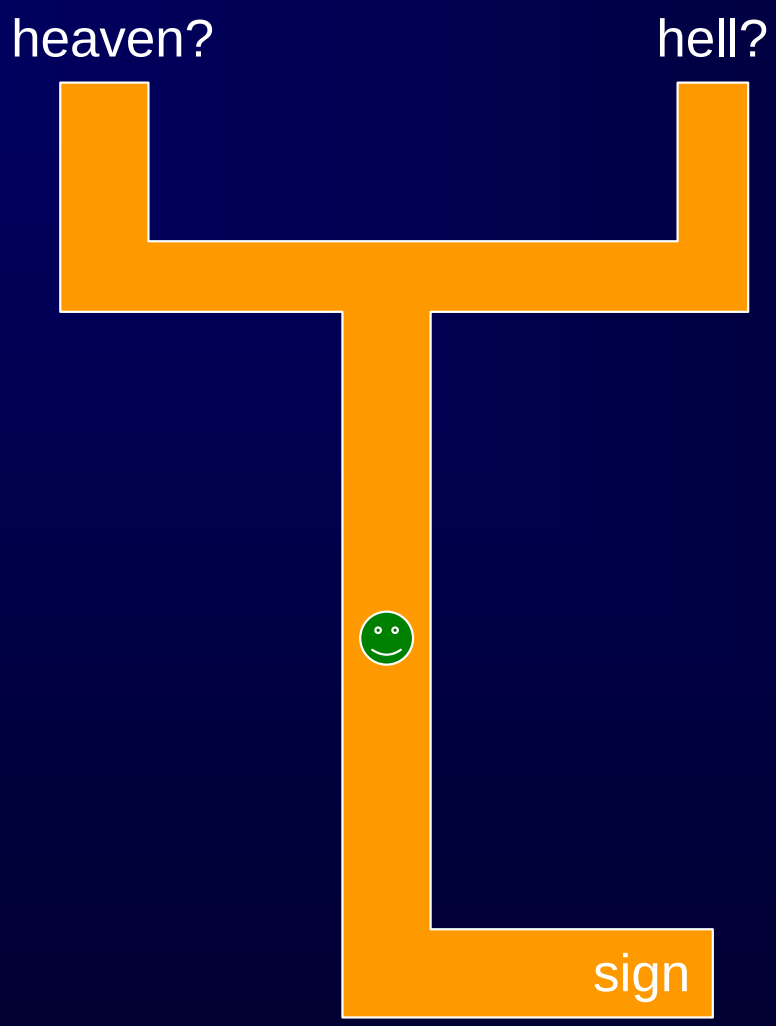
History solution

- Solution consists in converting the POMDP problem into a MDP problem by extending the information of the state
- Problems:
  - Length window is a new parameter to be estimated
  - Number of states (size of the table) grows exponentially with the length of the window
  - Some states past states are not necessary to disambiguate the state
  - Same action to be used in a state need to be learned separately

UDM approach

- Learn windows of diffenrent length for each situation to be disambiguated
- Saves in memory
- Expensive in number of steps to learn
- Continuous assumption still aplies

Window length in this case?



Active Memorization

- Concept:
  - In order to disambiguate the aliased perceptions, the agent's state is extended with a limited number of memory registers. The set of actions of the agent is also extended to allow the control of these memory registers.
  - The agent must learn to set adequately its internal state in order to solve the task.
- Littman (Littman 1994), ZCSM (Cliff, Ross 1994), (Martin 1998)

Active Memorization

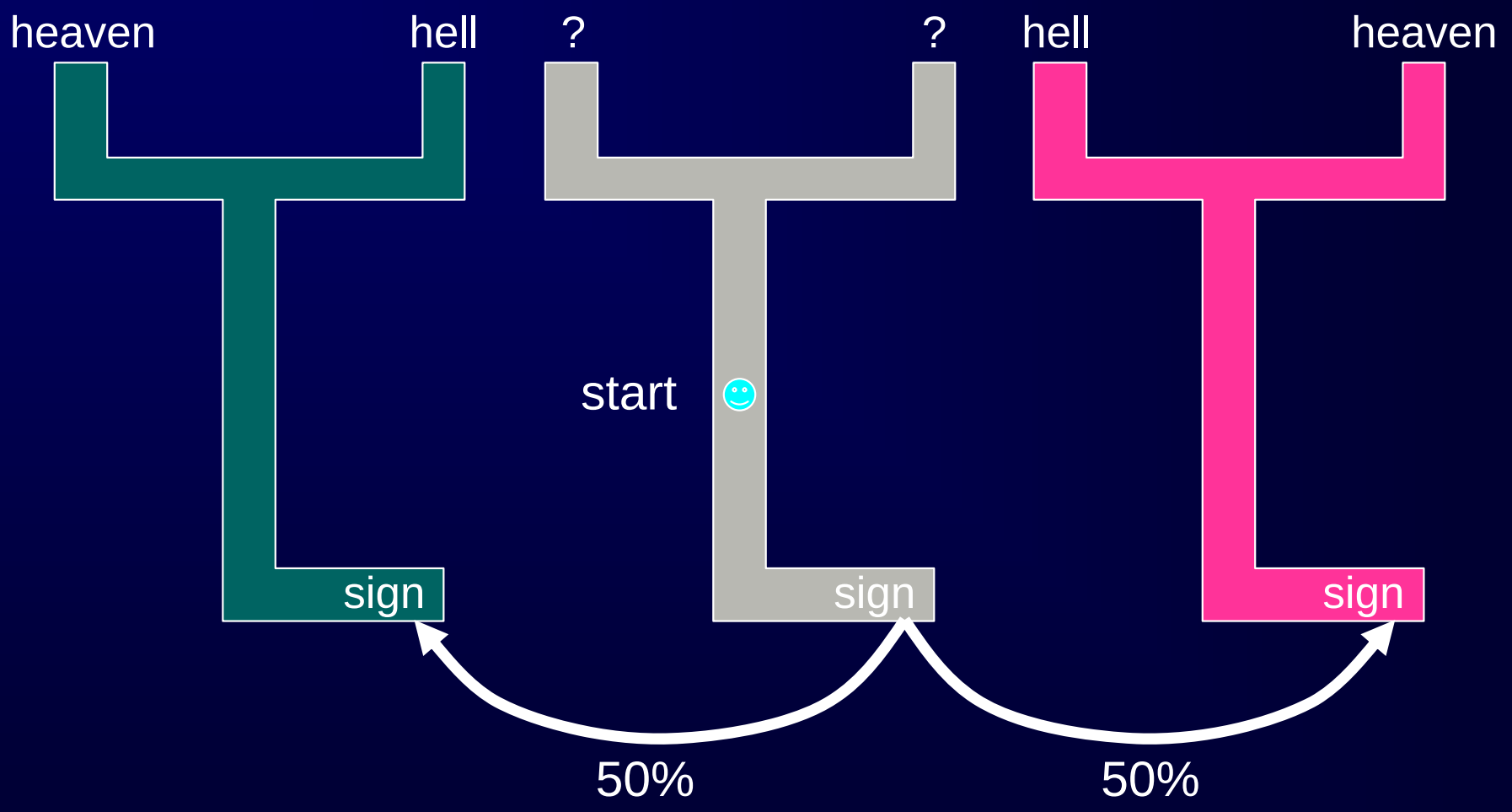
- A.M. separates memory actions and external actions.
- Littman and ZCS apps, 2^n actions (where n is the number of registers)
- Martin decreases the number of memory actions to only one per memory register that sets it to its current value plus one module the number of possible values of the memory bit.

n memory bits $\rightarrow m+n$ actions (m is the number of external actions)

Evaluation of Active Memorization

- Advantage:
  - No need to maintain all previous n actions and perceptions
- Cost:
  - In some approaches exponential growth of actions
  - Power of the approach can decrease in some problems compared with the previous memory approaches (but usually, such power is not required)
  - Number of memory registers has to be estimated beforehand

Stochastic, Partially Observable



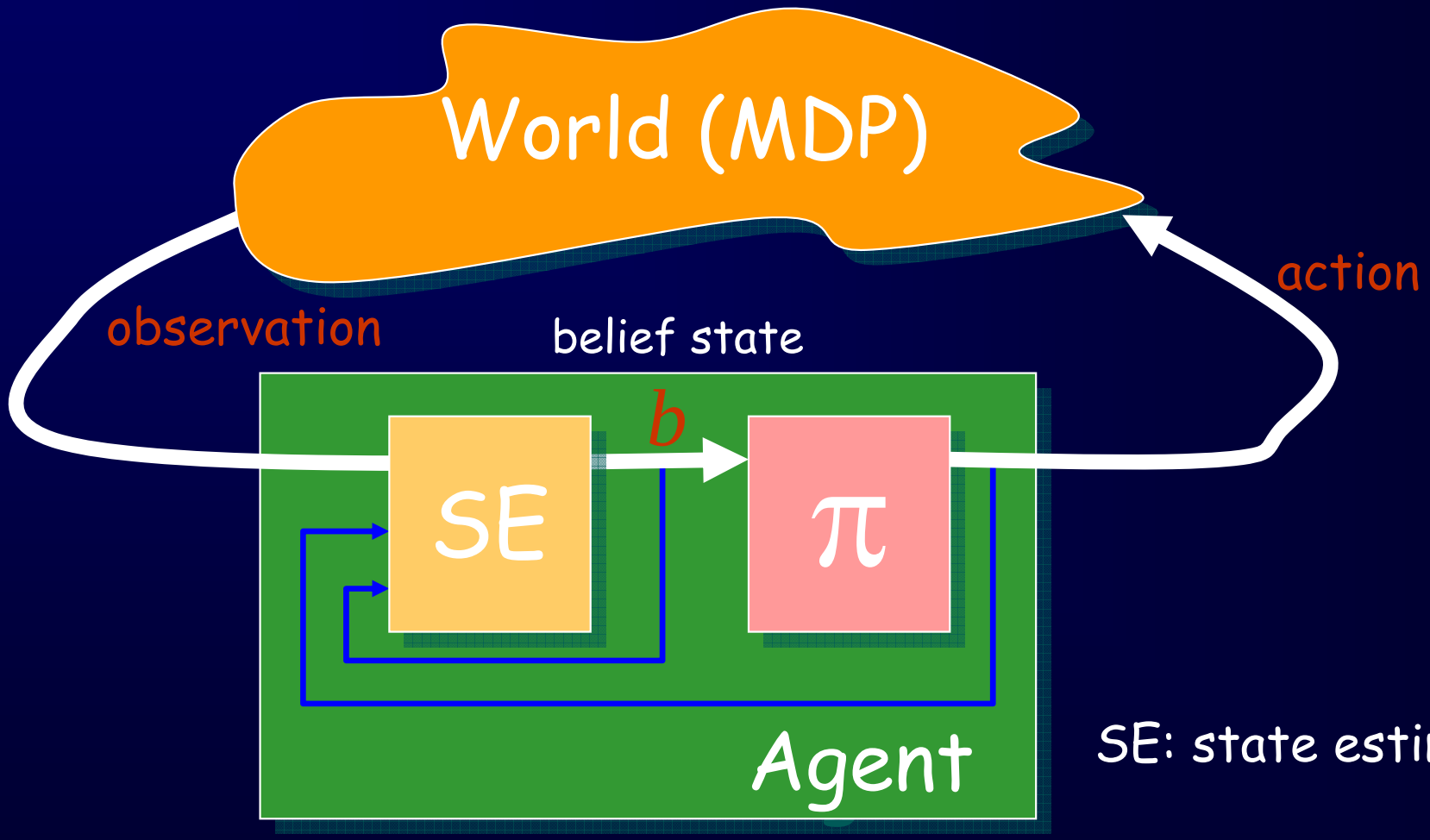
Content

- Introduction to the problem
- Convert to MDP: History Solutions
  - Window
  - UDM
  - Active Memorization
- Solve POMDPs
  - Belief States & Infinite-State MDP
  - Value Function of POMDP

Partial Observability

- Instead of directly measuring the current state, the agent makes an observation to get a hint about what state it is in.
- How to get hint (guess the state)?
  - To do an action and take an observation.
  - The observation can be probabilistic, i.e., it provides hint only.
  - The 'state' will be defined in probability sense.

POMDP Framework



SE: state estimator

Observation Model

Ω – a finite set of observations the agent can experience of its world.

$$O : S \times A \rightarrow \Pi(\Omega)$$

$$O(s', a, o) = P(o_{t+1} = o \mid s_{t+1} = s', a_t = a)$$

The probability of getting observation *o* given that the agent took action *a* and landed in state *s'*.

Definition – POMDP

A POMDP is a tuple $< S, A, T, R, \Omega, O >$

$< S, A, T, R >$ describes an MDP.

$O : S \times A \rightarrow \Pi(\Omega)$ is the observation function.

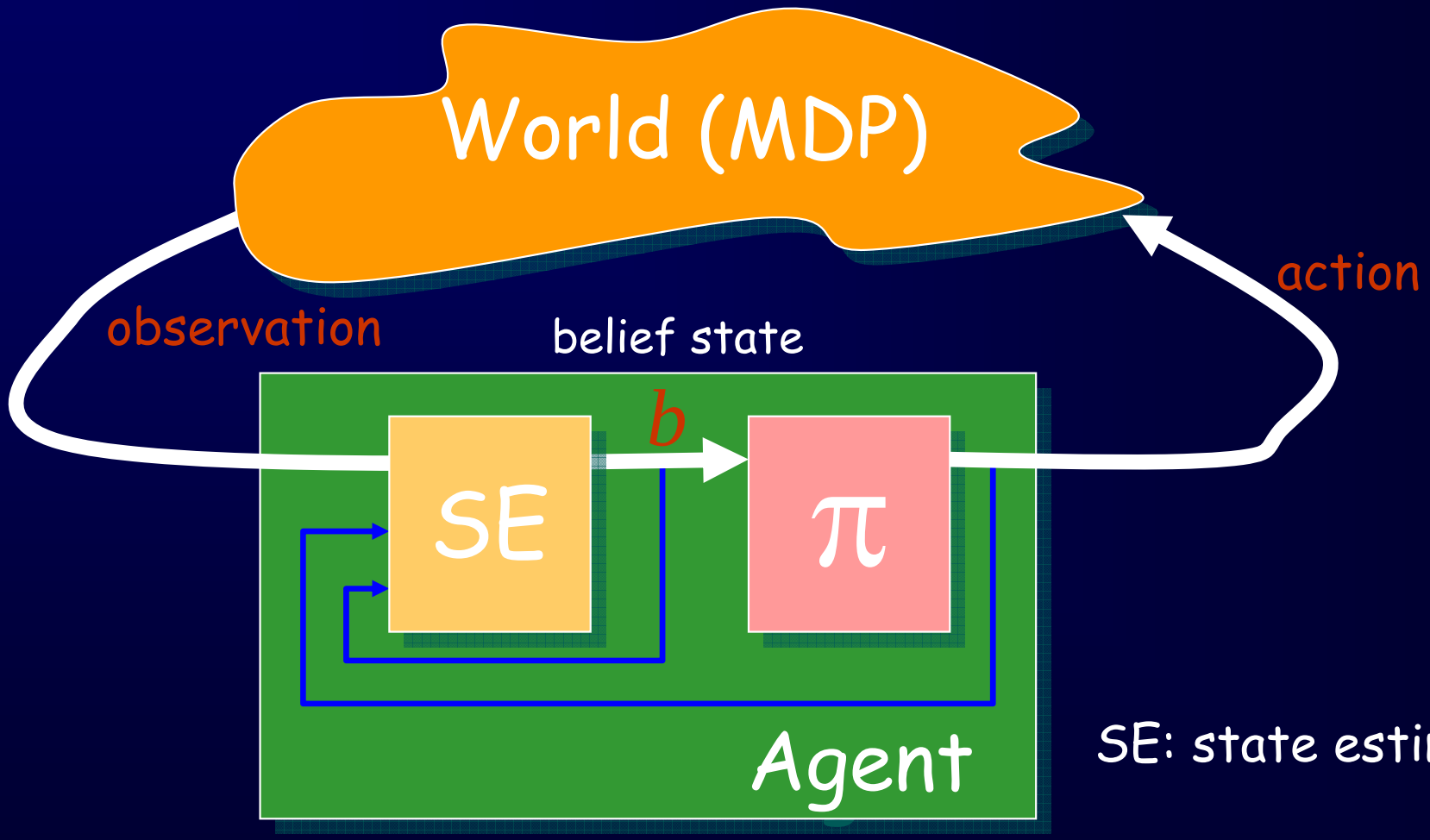
How to find optimal policy in such an environment?

Partial Observability

- Instead of directly measuring the current state, the agent makes an observation to get a hint about what state it is in.

- **How to get hint (guess the state)?**
  - **To do an action and take an observation.**
  - **The observation can be probabilistic, i.e., it provides hint only.**
  - **The 'state' will be defined in probability sense.**

POMDP Framework



SE: state estimator

Belief States

$$\mathbf{b} = (b(s_1), b(s_2), \ldots)^T, \quad s_i \in S, \; b(s_i) \geq 0$$

$$\sum_{s \in S} b(s) = 1$$

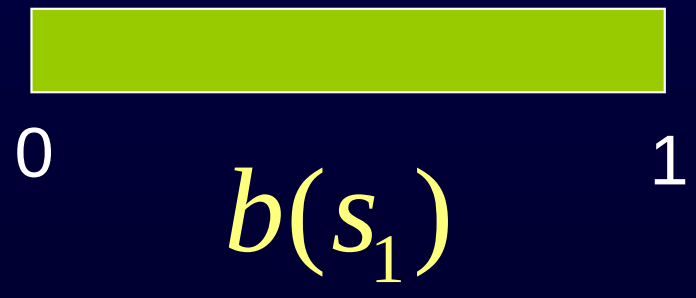
There are uncountably infinite number of belief states.

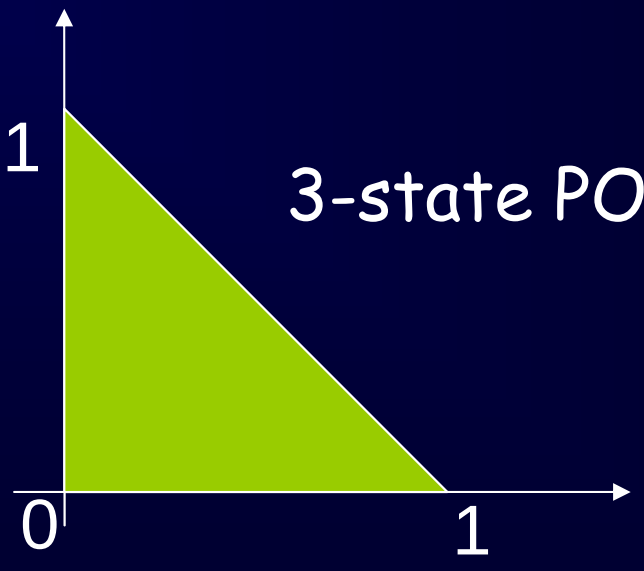
State Space

$$\mathbf{b} = (b(s_1), b(s_2), \ldots)^T, \quad s_i \in S, \; b(s_i) \geq 0$$

$$\sum_{s \in S} b(s) = 1$$

There are uncountably infinite number of belief states.





State Estimation

$$\mathbf{b} = (b(s_1), b(s_2), \ldots)^T, \quad s_i \in S, \; b(s_i) \geq 0$$

$$\sum_{s \in S} b(s) = 1$$

There are uncountably infinite number of belief states.

State estimation:

Given $\mathbf{b}_t$, a_t and o_{t+1}, $\mathbf{b}_{t+1}$=?

State Estimation

$$\mathbf{b}_t = (b_t(s_1), b_t(s_2), \ldots)^T$$
$$\mathbf{b}_{t+1} = (b_{t+1}(s_1), b_{t+1}(s_2), \ldots)^T$$

$$b_{t+1}(s') = P(s' \mid o, a, \mathbf{b}_t) = \frac{P(o \mid s', a, \mathbf{b}_t) P(s' \mid a, \mathbf{b}_t)}{P(o \mid a, \mathbf{b}_t)}$$

$$= \frac{P(o \mid s', a) \sum_{s \in S} P(s' \mid s, a, \mathbf{b}_t) P(s \mid a, \mathbf{b}_t)}{P(o \mid a, \mathbf{b}_t)}$$

$$= \frac{P(o \mid s', a) \sum_{s \in S} P(s' \mid s, a) b_t(s)}{P(o \mid a, \mathbf{b}_t)}$$

$$= \frac{O(s', a, o) \sum_{s \in S} T(s, a, s') b_t(s)}{P(o \mid a, \mathbf{b}_t)}$$

State Estimation

$$\mathbf{b}_t = (b_t(s_1), b_t(s_2), \ldots)^T$$
$$\mathbf{b}_{t+1} = (b_{t+1}(s_1), b_{t+1}(s_2), \ldots)^T$$

$$b_{t+1}(s') = P(s' \mid o, a, \mathbf{b}_t) = \frac{P(o \mid s', a, \mathbf{b}_t) P(s' \mid a, \mathbf{b}_t)}{P(o \mid a, \mathbf{b}_t)}$$

$$= \frac{P(o \mid s', a) \sum_{s \in S} P(s' \mid s, a, \mathbf{b}_t) P(s \mid a, \mathbf{b}_t)}{P(o \mid a, \mathbf{b}_t)}$$

$$= \frac{P(o \mid s', a) \sum_{s \in S} P(s' \mid s, a) b_t(s)}{P(o \mid a, \mathbf{b}_t)}$$

$$= \frac{O(s', a, o) \sum_{s \in S} T(s, a, s') b_t(s)}{P(o \mid a, \mathbf{b}_t)}$$

$$\mathbf{b}_{t+1} = SE(\mathbf{b}_t, a, o)$$

State Estimation

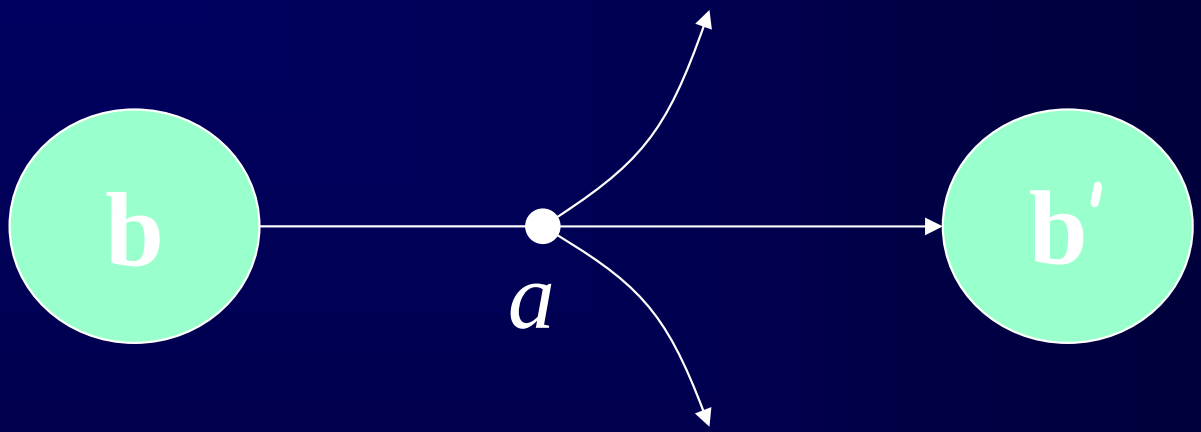
$$= \frac{O(s',a,o)\sum_{s} T(s,a,s')b_t(s)}{P(o \mid a,\mathbf{b}_t)}$$

$$P(o \mid a,b) = \sum_{s'} P(o \mid a,s',b)P(s' \mid a,b)$$

$$= \sum_{s'} O(s',o)P(s' \mid a,b) = \sum_{s'} O(s',o)\sum_{s} T(s,a,s')b(s)$$

$$\mathbf{b}' = SE(\mathbf{b}, a, o)$$

State Transition Function



$$\tau(\mathbf{b}, a, \mathbf{b}') = P(\mathbf{b}' \mid \mathbf{b}, a)$$

$$= \sum_{o \in \Omega} P(\mathbf{b}' \mid \mathbf{b}, a, o) P(o \mid \mathbf{b}, a)$$

POMDP = Infinite-State MDP

- A POMDP is an MDP with tuple $< \mathcal{B}, \mathcal{A}, \rho, \tau >$
- $\mathcal{B}$ – a set of Belief states
- $\mathcal{A}$ – the finite set of actions (the same as the original MDP)
- $\tau : \mathcal{B} \times \mathcal{A} \rightarrow \Pi(\mathcal{B})$ – state-transition function

$$\tau(\mathbf{b}, a, \mathbf{b}') = P(\mathbf{b}_{t+1} = \mathbf{b}' \mid \mathbf{b}_t = \mathbf{b}, a_t = a)$$

- $\rho : \mathcal{B} \times \mathcal{A} \rightarrow R$ – the reward function

Reward Function

$$\rho(\mathbf{b}, a) = \sum_{s \in S} b(s) \underbrace{R(s, a)}$$

The reward function of the original MDP

Solving POMDP

Best Strategy

- Value Iteration Algorithm:
  - Input: Actions, States, Reward Function, Probabilistic Transition Function.
  - Derive a mapping from states to “best” actions for a given horizon of time.
  - Starts with horizon length 1 and iteratively found the value function for the desired horizon.

→Optimal Policy

→Maps states to actions (S → A).

→It depends only on current state (Markov Property).

→**To apply this we must know the agent’s state.**

Partially Observable Markov Decision Processes

- Domains with partial information available about the current state (we can't observe the current state).
  - The observation can be probabilistic.
    - We need an observation function.
  - Uncertainly about current state.
- Non-Markovian process: required keeping track of the entire history.

Partially Observable Markov Decision Processes

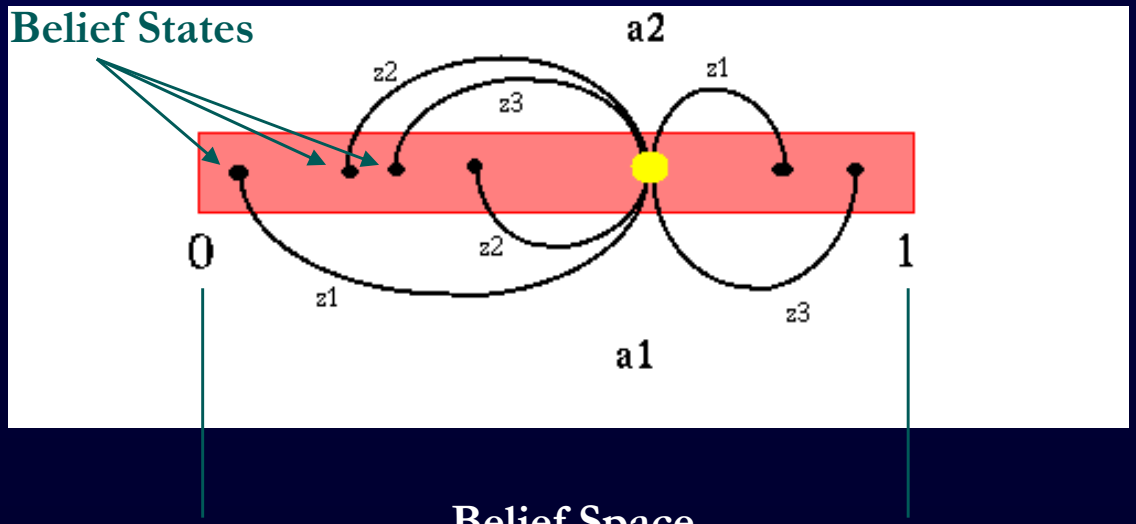
- In addition to MDP model we have:
  - Observation: A set of observation of the state.
    - Z = A set of observations.
  - Observation Function: Relation between the state and the observation.
    - O(s, a, z) = Pr(z |s, a).

Background on Solving POMDPs

- We have to find a mapping from probability distribution over states to actions.
  - Belief State: the probability distribution over states.
  - Belief Space: the entire probability space.
- Assuming finite number of possible actions and observations, there are finite number of possible next beliefs states.
- Our next belief state is fully determined and it depends only on the current belief state (Markov Property).

Background on Solving POMDPs

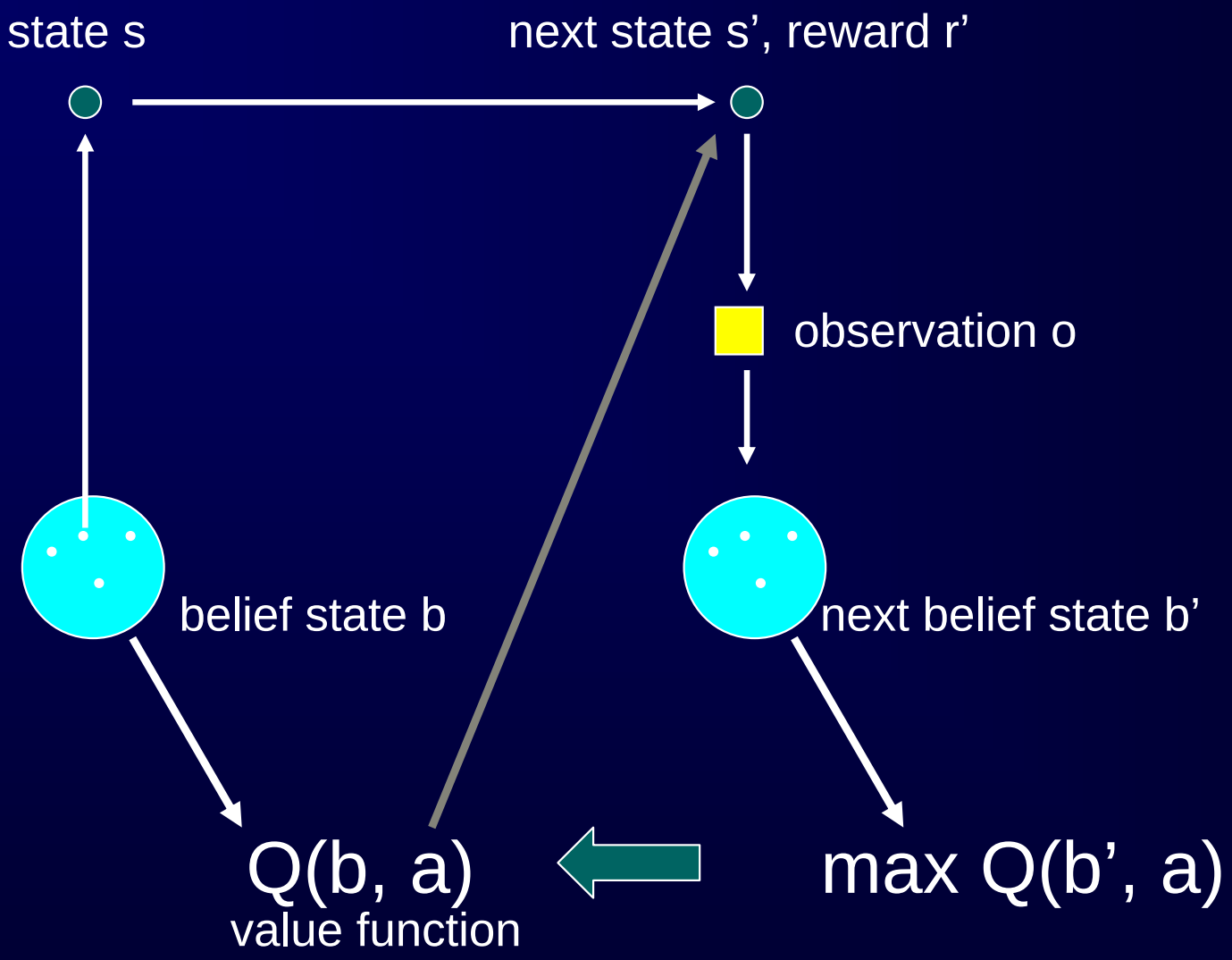
- Start from belief state b (Yellow Dot).
- Two states s1, s2.
- Two actions a1, a2.
- Tree observations z1, z2, z3.



Policies for POMDPs

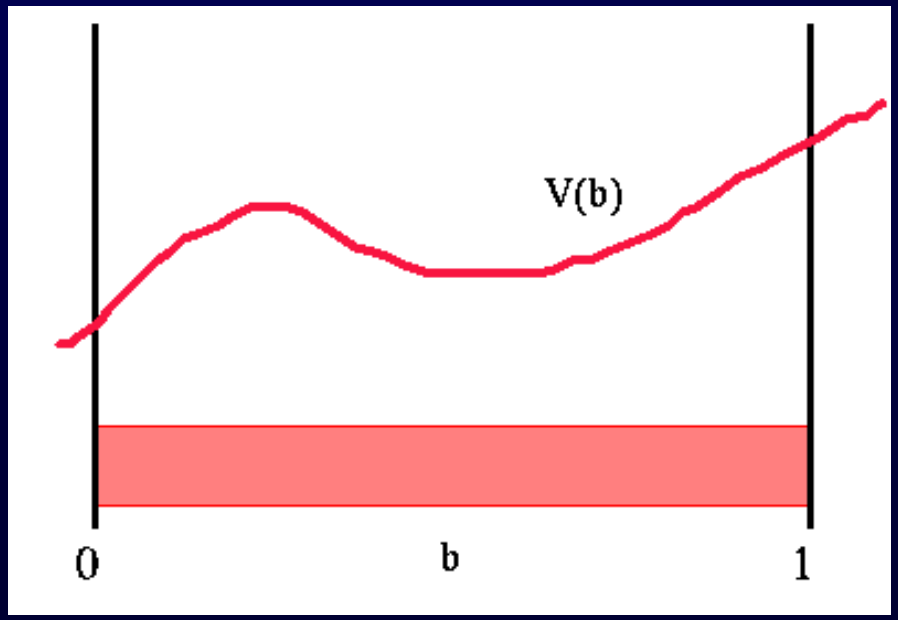
- An optimal POMDP policy maps belief states to actions.
- The way in which one would use a computed policy is to start with some a priori belief about where you are in the world. The continually:
  1. Use the policy to select action for current belief state;
  2. Execute the action;
  3. Receive an observation;
  4. Update the belief state using current belief, action and observation;
  5. Repeat.

Value Iteration in Belief Space



Value Function

- The Optimal Policy computation is based on Value Iteration.
- Main problem using the value iteration is that the space of all belief states is continuous.



Value Function

- For each belief state get a single expected value.
- Find the expected value of all belief states.
- Yield a value function defined over all belief space.

Algorithms to solve POMDPs

- Algorithms for POMDPs use a form of dynamic programming, called dynamic programming updates.
- One Value Function is translated into a another.
- Some of the algorithms using DPU:
  - One pass (Sondik 1971)
  - Exhaustive (Monahan 1982)
  - Linear support (Cheng 1988)
  - Witness (Littman, Cassandra & Kaelbling 1996)
  - Dynamic Pruning (Zhang & Liu 1996)