POMDPs:
Partially Observable Markov Decision Processes

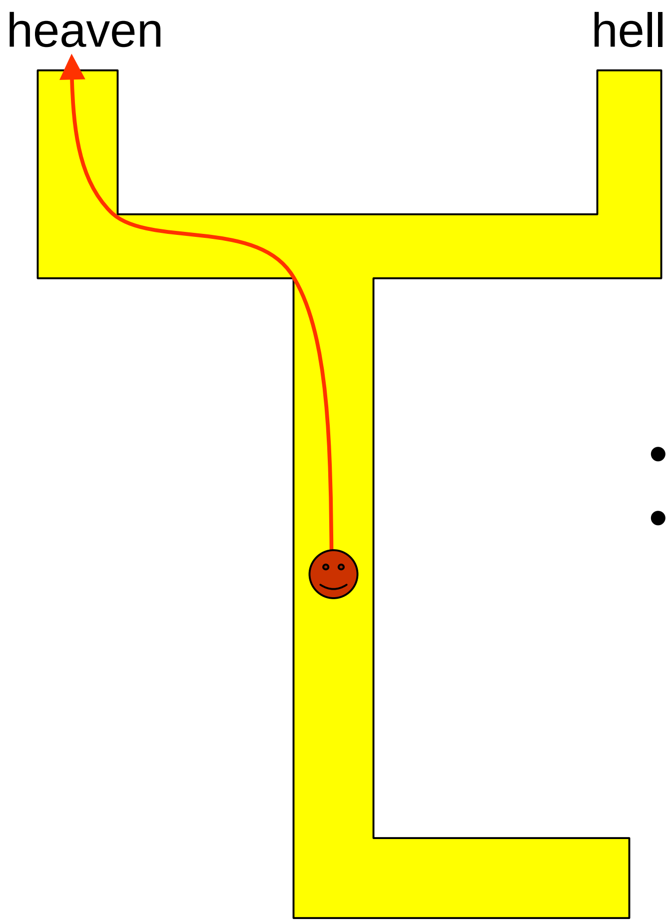
Advanced AI

Wolfram Burgard

Types of Planning Problems

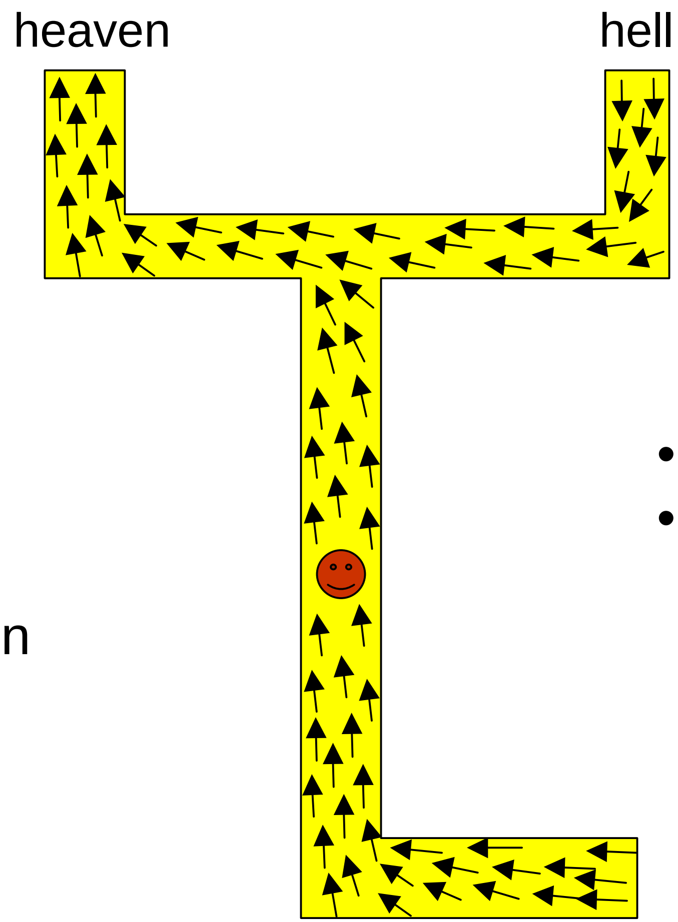
	State	Action Model
Classical Planning	observable	Deterministic, accurate
MDPs	observable	stochastic
POMDPs	partially observable	stochastic

Classical Planning



- World deterministic
- State observable

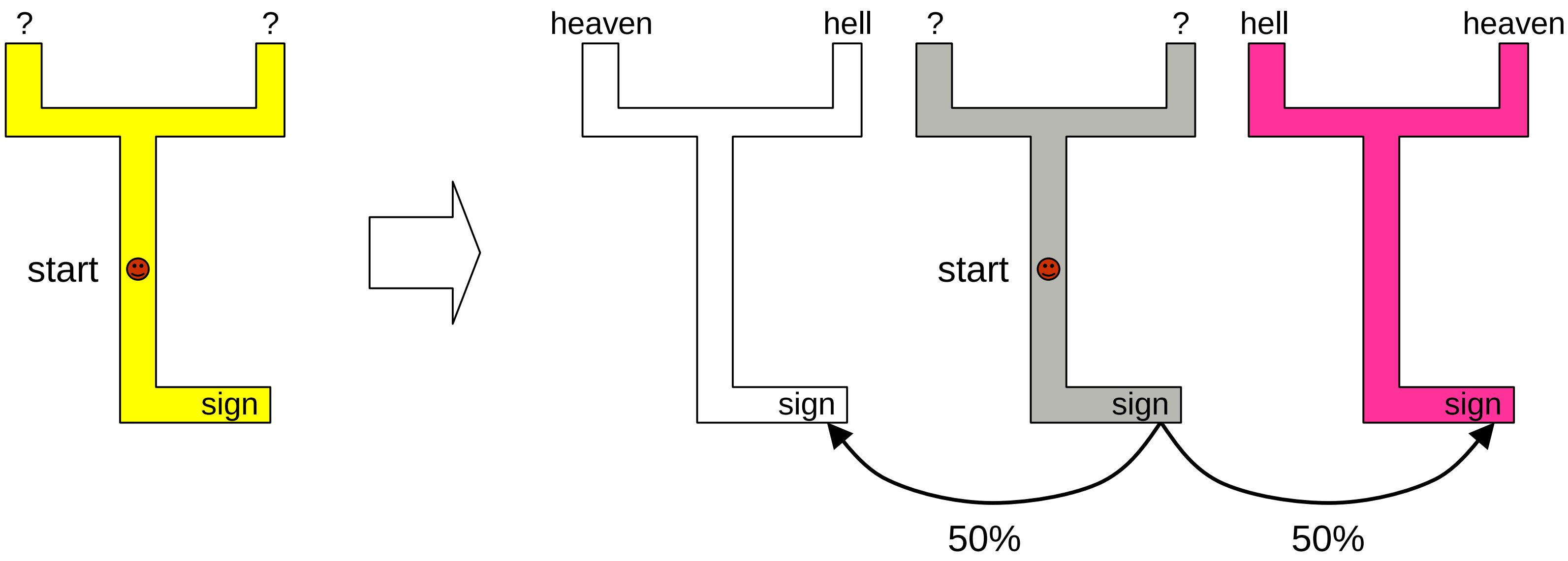
MDP-Style Planning



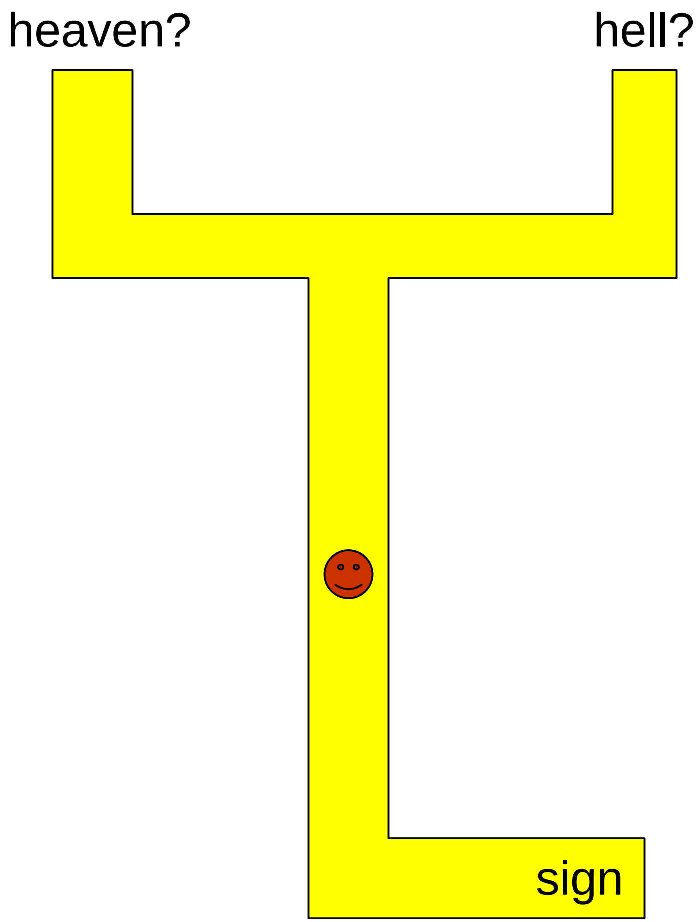
- Policy
- Universal Plan
- Navigation function

- World stochastic
- State observable

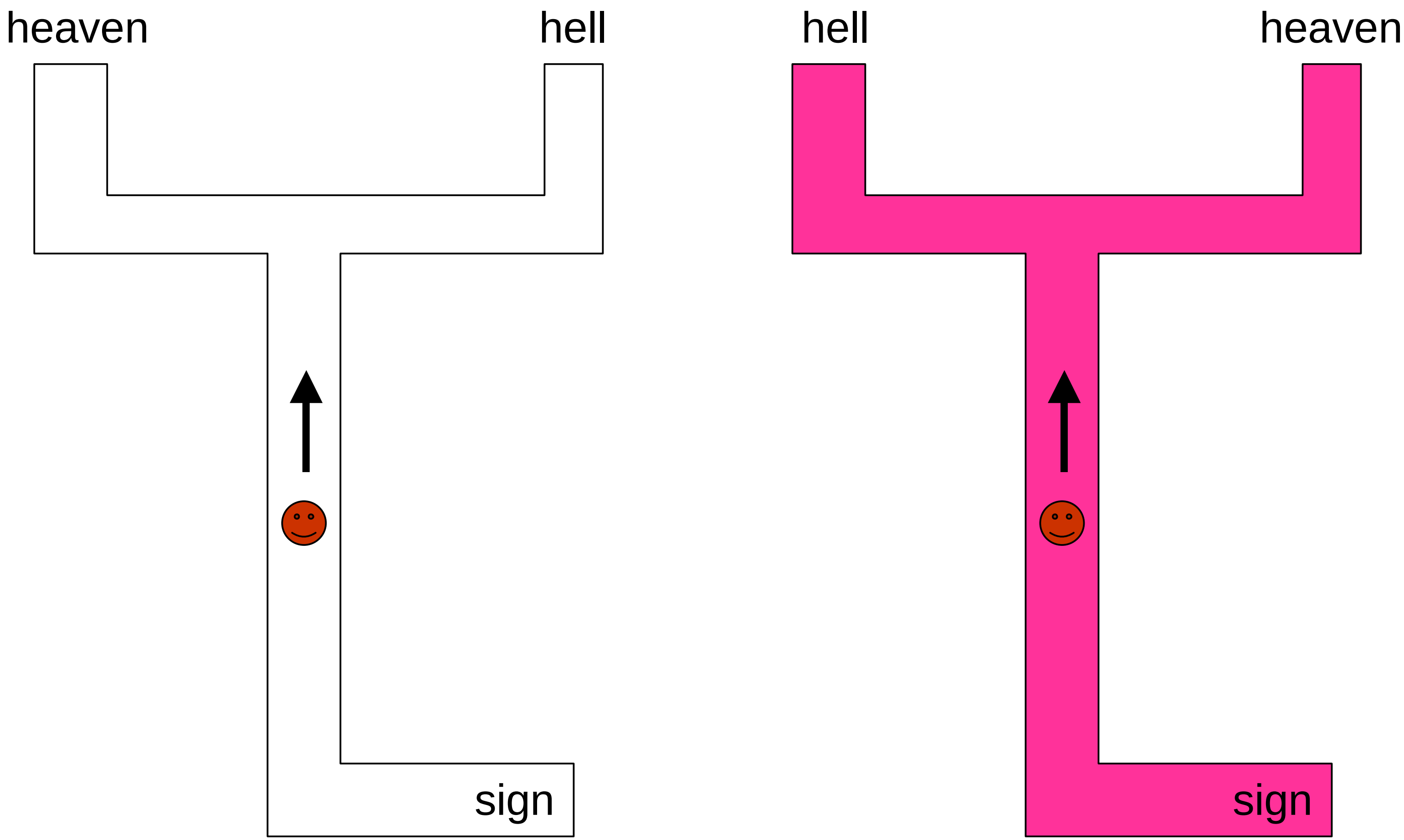
Stochastic, Partially Observable



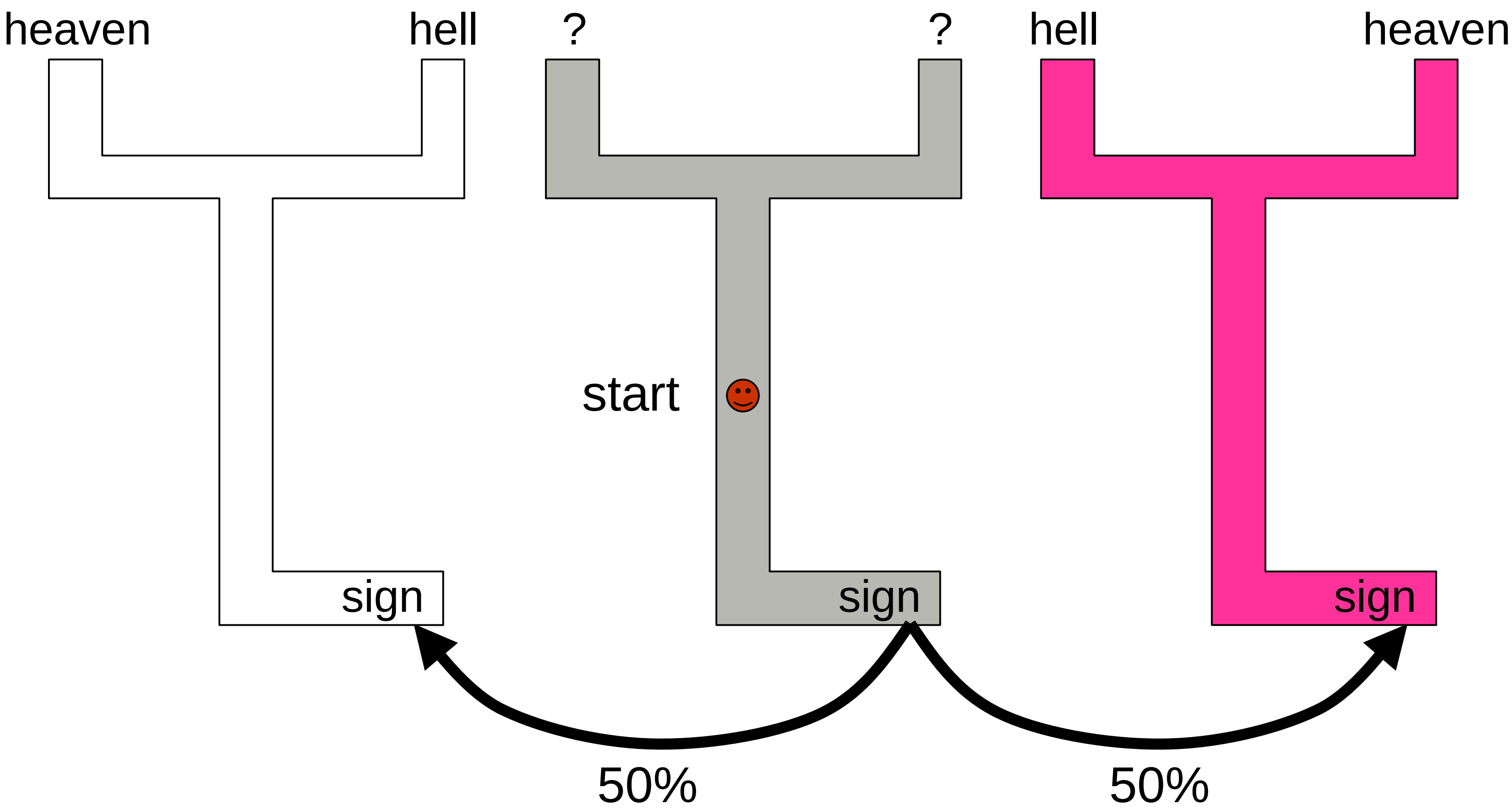
Stochastic, Partially Observable



Stochastic, Partially Observable



Stochastic, Partially Observable



Notation (1)

- Recall the Bellman optimality equation:

$$V^*(s) = \max_{a \in A(s)} \sum_{s'} P_{ss'}^a \left[R_{ss'}^a + \gamma V^*(s') \right]$$

- Throughout this section we assume

$$R_{ss'}^a = \frac{1}{\gamma} R_s^a = \frac{1}{\gamma} r(s,a)$$

is independent of s' so that the Bellman optimality equation turns into

$$V^*(s) = \gamma \max_{a \in A(s)} \left[R_s^a + \sum_{s'} V^*(s') P_{ss'}^a \right] = \gamma \max_{a \in A(s)} \left[r(s,a) + \sum_{s'} V^*(s') P_{ss'}^a \right]$$

Notation (2)

- In the remainder we will use a slightly different notation for this equation:

$$V(x) \quad = \quad \gamma \max_u \left[r(x,u) + \int V(x')\, p(x' \mid u, x)\, dx' \right]$$

- According to the previously used notation we would write

$$V^*(s) = \gamma \max_{a \in A(s)} \left[r(s,a) + \sum_{s'} V^*(s') P_{ss'}^{a} \right]$$

- We replaced s by x and a by u, and turned the sum into an integral.

Value Iteration

- Given this notation the value iteration formula is

$$V_T(x) \;=\; \gamma \max_u \left[r(x,u) + \int V_{T-1}(x')\, p(x' \mid u, x)\, dx' \right]$$

with

$$V_1(b) \;=\; \gamma \max_u r(x,u)$$

POMDPs

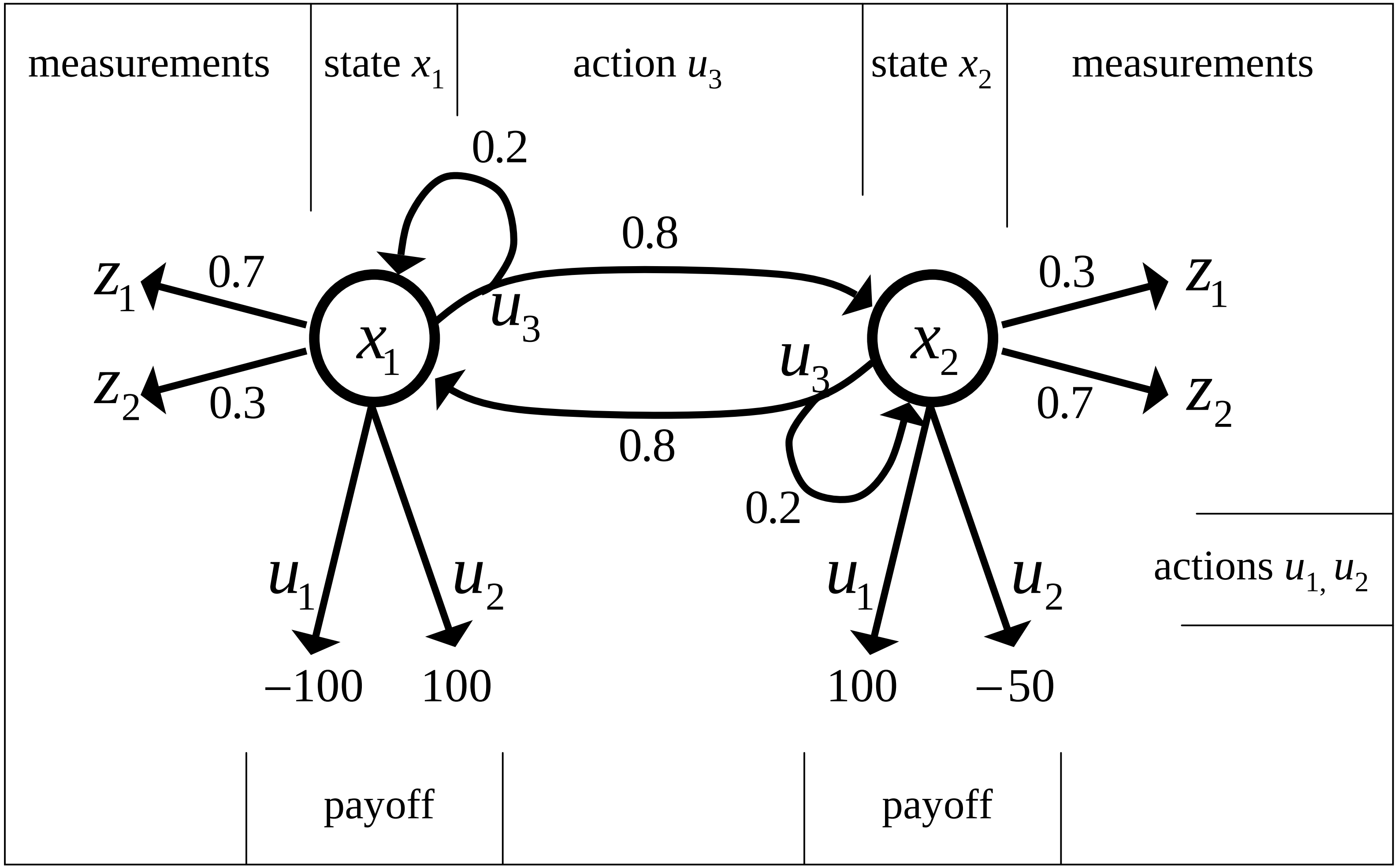
- In POMDPs we apply the very same idea as in MDPs.
- **Since the state is not observable**, the agent has to **make its decisions based on** the belief state which is a **posterior distribution over states**.
- Let b be the belief of the agent about the state under consideration.
- POMDPs compute a **value function over belief spaces**:

$$V_T(b) \;=\; \gamma \max_u \left[r(b,u) + \int V_{T-1}(b')\, p(b' \mid u, b)\, db' \right]$$

Problems

- Each belief is a probability distribution, thus, **each value in a POMDP is a function of an entire probability distribution**.
- **This is problematic, since probability distributions are continuous**.
- Additionally, we have to deal with the **huge complexity of belief spaces**.
- For **finite worlds** with finite state, action, and measurement spaces and finite horizons, however, we can **effectively represent the value functions by piecewise linear functions**.

An Illustrative Example



The Parameters of the Example

- The actions u_1 and u_2 are terminal actions.
- The action u_3 is a sensing action that potentially leads to a state transition.
- The horizon is finite and γ=1.

$$
\begin{aligned}
r(x_1, u_1) &= -100 & \quad r(x_2, u_1) &= +100 \\
r(x_1, u_2) &= +100 & \quad r(x_2, u_2) &= -50 \\
r(x_1, u_3) &= -1 & \quad r(x_2, u_3) &= -1
\end{aligned}
$$

$$
\begin{aligned}
p(x'_1|x_1, u_3) &= 0.2 & \quad p(x'_2|x_1, u_3) &= 0.8 \\
p(x'_1|x_2, u_3) &= 0.8 & \quad p(z'_2|x_2, u_3) &= 0.2
\end{aligned}
$$

$$
\begin{aligned}
p(z_1|x_1) &= 0.7 & \quad p(z_2|x_1) &= 0.3 \\
p(z_1|x_2) &= 0.3 & \quad p(z_2|x_2) &= 0.7
\end{aligned}
$$

Payoff in POMDPs

- In MDPs, the payoff (or return) depended on the state of the system.
- In POMDPs, however, the true state is not exactly known.
- Therefore, we compute the **expected payoff** by **integrating over all states**:

$$\begin{aligned} r(b,u) &= E_x[r(x,u)] \\ &= \int r(x,u)p(x)\ dx \\ &= p_1\ r(x_1,u) + p_2\ r(x_2,u) \end{aligned}$$

Payoffs in Our Example (1)

- If we are totally certain that we are in state x_1 and execute action u_1, we receive a reward of -100
- If, on the other hand, we definitely know that we are in x_2 and execute u_1, the reward is +100.
- In between it is the linear combination of the extreme values weighted by their probabilities

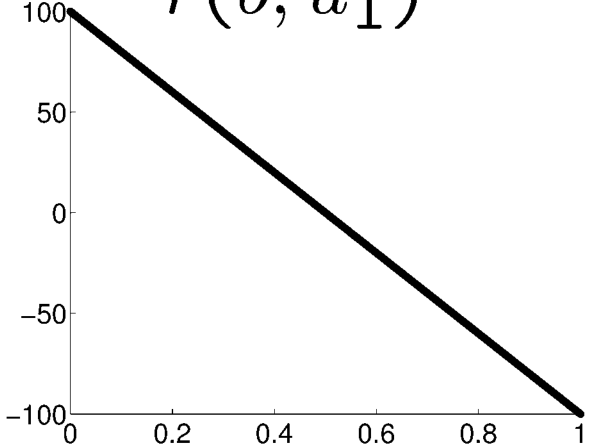
$$\begin{aligned} r(b,u_1) &= -100\,p_1 + 100\,p_2 \\ &= -100\,p_1 + 100\,(1-p_1) \end{aligned}$$

$$r(b,u_2) = 100\,p_1 - 50\,(1-p_1)$$

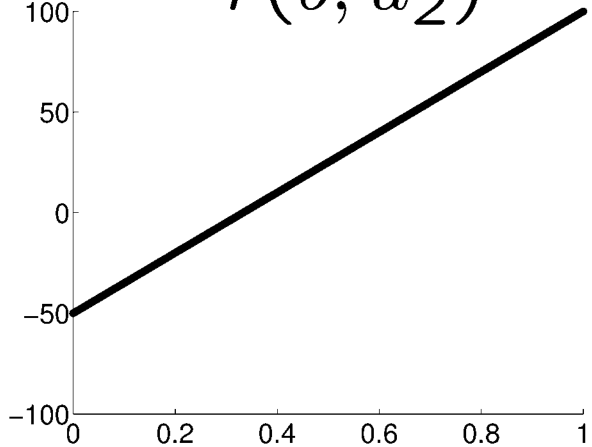
$$r(b,u_3) = -1$$

Payoffs in Our Example (2)

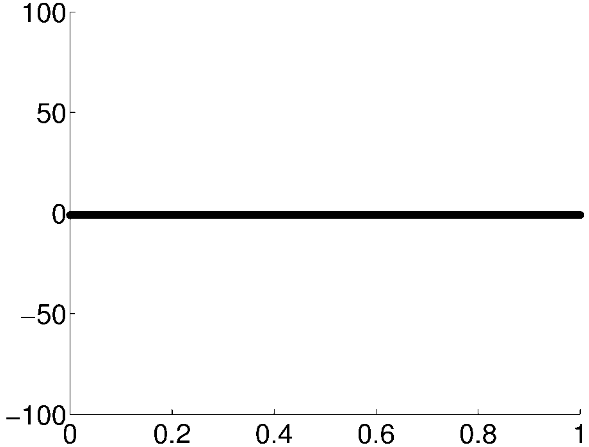
$r(b, u_1)$



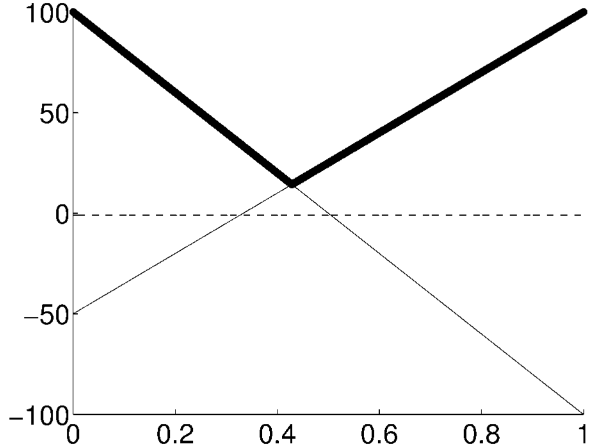
$r(b, u_2)$



$r(b, u_3)$



$V_1(b) = max_u r(b, u)$



The Resulting Policy for T=1

- Given we have a finite POMDP with T=1, we would use $V_1(b)$ to determine the optimal policy.
- In our example, the optimal policy for T=1 is

$$\pi_1(b) = \begin{cases} u_1 & \text{if } p_1 \le \frac{3}{7} \\ u_2 & \text{if } p_1 > \frac{3}{7} \end{cases}$$

- This is the upper thick graph in the diagram.

Piecewise Linearity, Convexity

- The resulting value function $V_1(b)$ is the maximum of the three functions at each point

$$
\begin{aligned}
V_1(b) &= \max_u r(b,u) \\
&= \max \left\{ \begin{array}{rr} -100\,p_1 & +100\,(1-p_1) \\ 100\,p_1 & -50\,(1-p_1) \\ -1 & \end{array} \right\}
\end{aligned}
$$

- It is piecewise linear and convex.

Pruning

- If we carefully consider $V_1(b)$, we see that only the first two components contribute.
- The third component can therefore safely be pruned away from $V_1(b)$.

$$V_1(b) \;=\; \max \left\{ \begin{array}{rr} -100\, p_1 & +100\,(1-p_1) \\ 100\, p_1 & -50\,(1-p_1) \end{array} \right\}$$

Increasing the Time Horizon

- If we go over to a time horizon of T=2, the agent can also consider the sensing action u_3.
- Suppose we perceive z_1 for which $p(z_1 \mid x_1)=0.7$ and $p(z_1 \mid x_2)=0.3$.
- Given the observation z_1 we update the belief using Bayes rule.
- Thus $V_1(b \mid z_1)$ is given by

$$
\begin{aligned}
V_1(b \mid z_1) &= \max \left\{ \begin{array}{cc} -100 \cdot \frac{0.7\, p_1}{p(z_1)} & +100 \cdot \frac{0.3\,(1-p_1)}{p(z_1)} \\ 100 \cdot \frac{0.7\, p_1}{p(z_1)} & -50 \cdot \frac{0.3\,(1-p_1)}{p(z_1)} \end{array} \right\} \\
&= \frac{1}{p(z_1)} \max \left\{ \begin{array}{cc} -70\, p_1 & +30\,(1-p_1) \\ 70\, p_1 & -15\,(1-p_1) \end{array} \right\}
\end{aligned}
$$

Expected Value after Measuring

- Since we do not know in advance what the next measurement will be, we have to compute the expected belief

$$
\begin{aligned}
\bar{V}_1(b) &= E_z[V_1(b \mid z)] \\
&= \sum_{i=1}^{2} p(z_i)\, V_1(b \mid z_i) \\
&= \max \left\{ \begin{array}{rr} -70\, p_1 & +30\,(1-p_1) \\ 70\, p_1 & -15\,(1-p_1) \end{array} \right\} \\
&\quad + \max \left\{ \begin{array}{rr} -30\, p_1 & +70\,(1-p_1) \\ 30\, p_1 & -35\,(1-p_1) \end{array} \right\}
\end{aligned}
$$

Resulting Value Function

- The four possible combinations yield the following function which again can be simplified and pruned.

$$\bar{V}_1(b) = \max \left\{ \begin{array}{llll} -70\, p_1 & +30\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ -70\, p_1 & +30\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \end{array} \right\}$$

$$= \max \left\{ \begin{array}{ll} -100\, p_1 & +100\,(1-p_1) \\ +40\, p_1 & +55\,(1-p_1) \\ +100\, p_1 & -50\,(1-p_1) \end{array} \right\}$$

State Transitions (Prediction)

- When the agent selects u_3 its state potentially changes.
- When computing the value function, we have to take these potential state changes into account.

$$
\begin{aligned}
p'_1 &= E_x[p(x_1 \mid x, u_3)] \\
&= \sum_{i=1}^{2} p(x_1 \mid x_i, u_3) p_i \\
&= 0.2 p_1 + 0.8(1 - p_1) \\
&= 0.8 - 0.6 p_1
\end{aligned}
$$

Resulting Value Function after executing u_3

- Taking also the state transitions into account, we finally obtain.

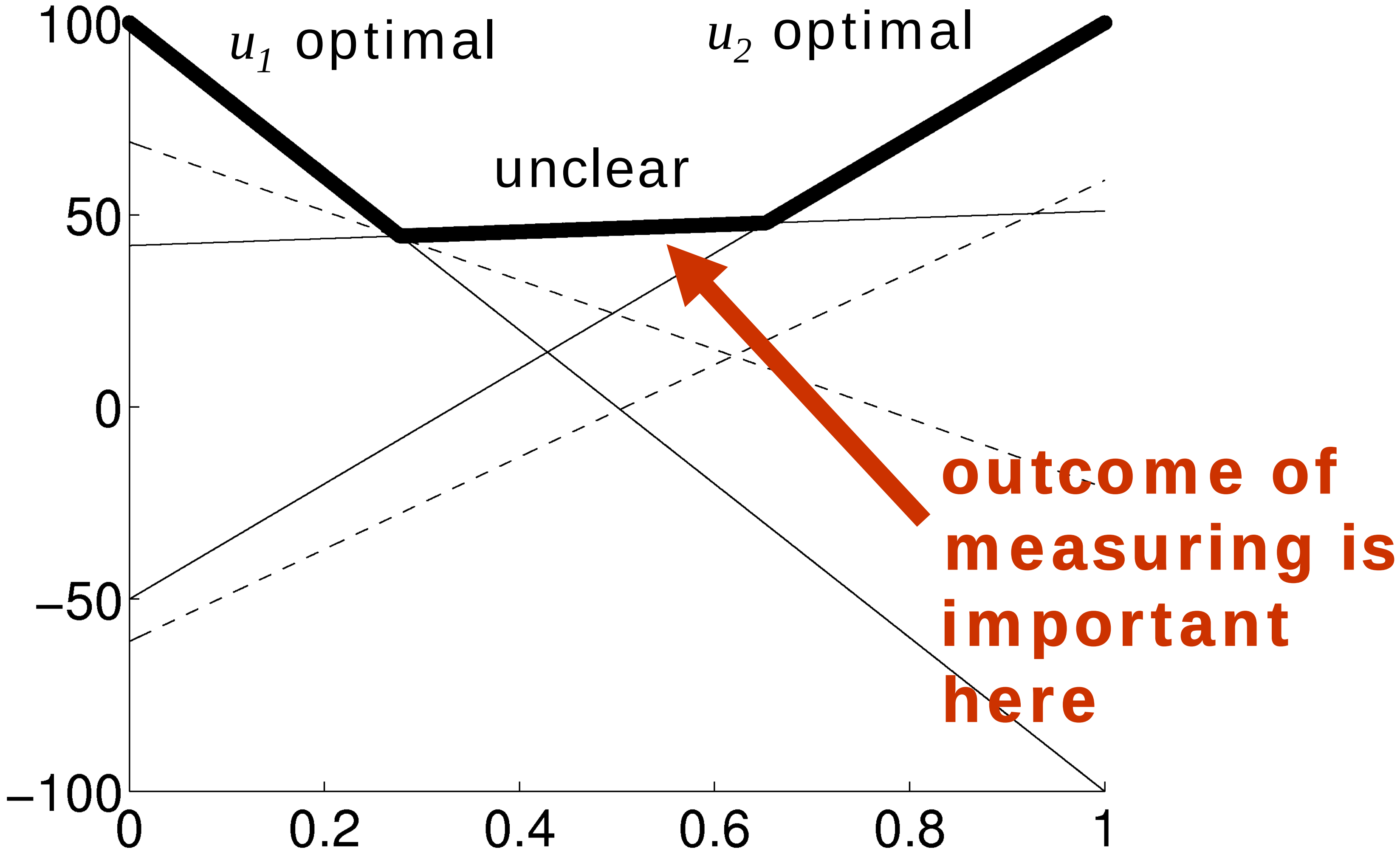
$$\bar{V}_1(b \mid u_3) \;=\; \max \left\{ \begin{array}{rr} 60\,p_1 & -60\,(1-p_1) \\ 52\,p_1 & +43\,(1-p_1) \\ -20\,p_1 & +70\,(1-p_1) \end{array} \right\}$$

Value Function for T=2

- Taking into account that the agent can either directly perform u_1 or u_2, or first u_3 and then u_1 or u_2, we obtain (after pruning)

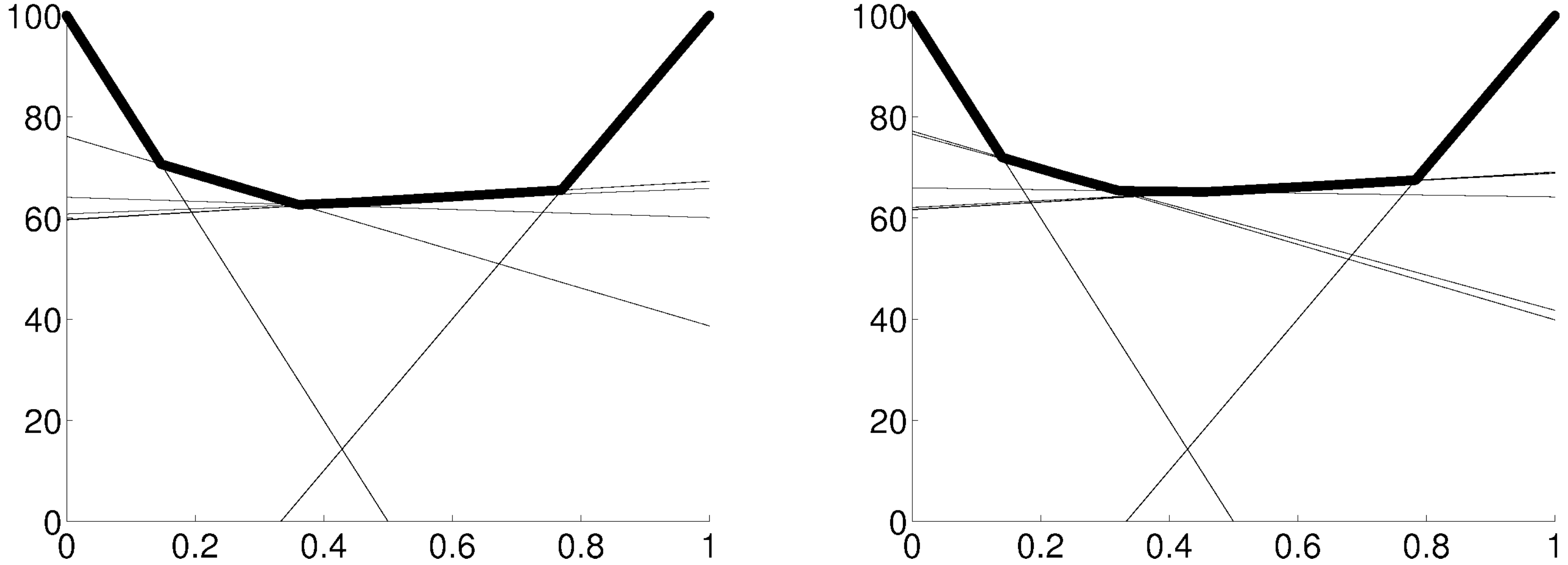
$$\bar{V}_2(b) = \max \left\{ \begin{array}{rr} -100\,p_1 & +100\,(1-p_1) \\ 100\,p_1 & -50\,(1-p_1) \\ 51\,p_1 & +42\,(1-p_1) \end{array} \right\}$$

Graphical Representation of $V_2(b)$



Deep Horizons and Pruning

- We have now completed a full backup in belief space.
- This process can be applied recursively.
- The value functions for T=10 and T=20 are



Why Pruning is Essential

- Each **update introduces additional linear components** to V.
- Each **measurement squares the number of linear components**.
- Thus, an unpruned value function for T=20 includes more than $10^{547,864}$ linear functions.
- At T=30 we have $10^{561,012,337}$ linear functions.
- The pruned value functions at T=20, in comparison, contains only 12 linear components.
- The combinatorial explosion of linear components in the value function are the major reason why **POMDPs are impractical for most applications**.

A Summary on POMDPs

- POMDPs compute the optimal action in partially observable, stochastic domains.
- For finite horizon problems, the resulting value functions are piecewise linear and convex.
- In each iteration the number of linear constraints grows exponentially.
- POMDPs so far have only been applied successfully to very small state spaces with small numbers of possible observations and actions.