Planning under Uncertainty

Lecture 10 – Thursday December 8, 2016

Objectives

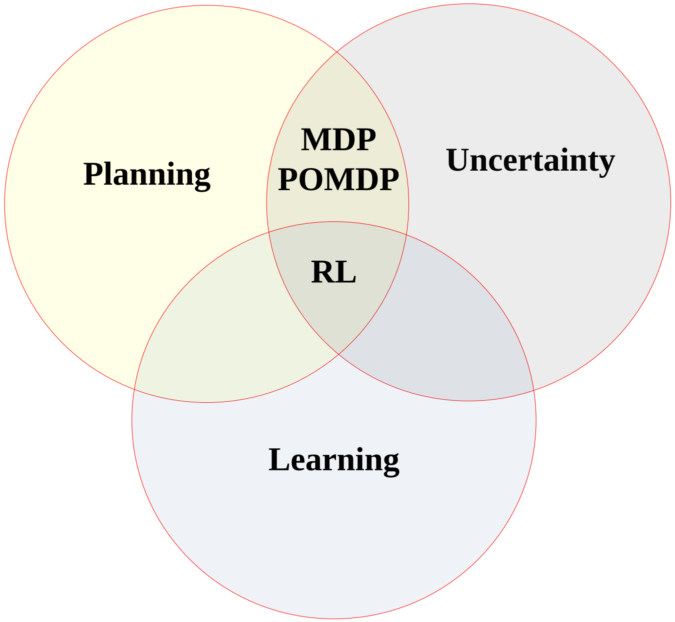
When you have finished this lecture you should be able to:

- Understand **MDP** and how to use this mathematical framework for **planning under uncertainty**.

Markov Decision Process (MDP)

Markov decision process (MDP) provides a mathematical framework for planning under uncertainty.

System	System state is fully observable	System state is partially observable
System is autonomous	Markov Chain (MC)	Hidden Markov Model (HMM)
System is controlled*	Markov Decision Process (MDP)	Partially Observable Markov Decision Process (POMDP)



MDP is used for modeling decision making in situations where outcomes are **partly random** and **partly under the control** of a decision maker.

* Controlled >> ability to change the current state by taking an action or ability to have control over state transitions.

Markov Decision Process (MDP)

- **Elements of MDP:**

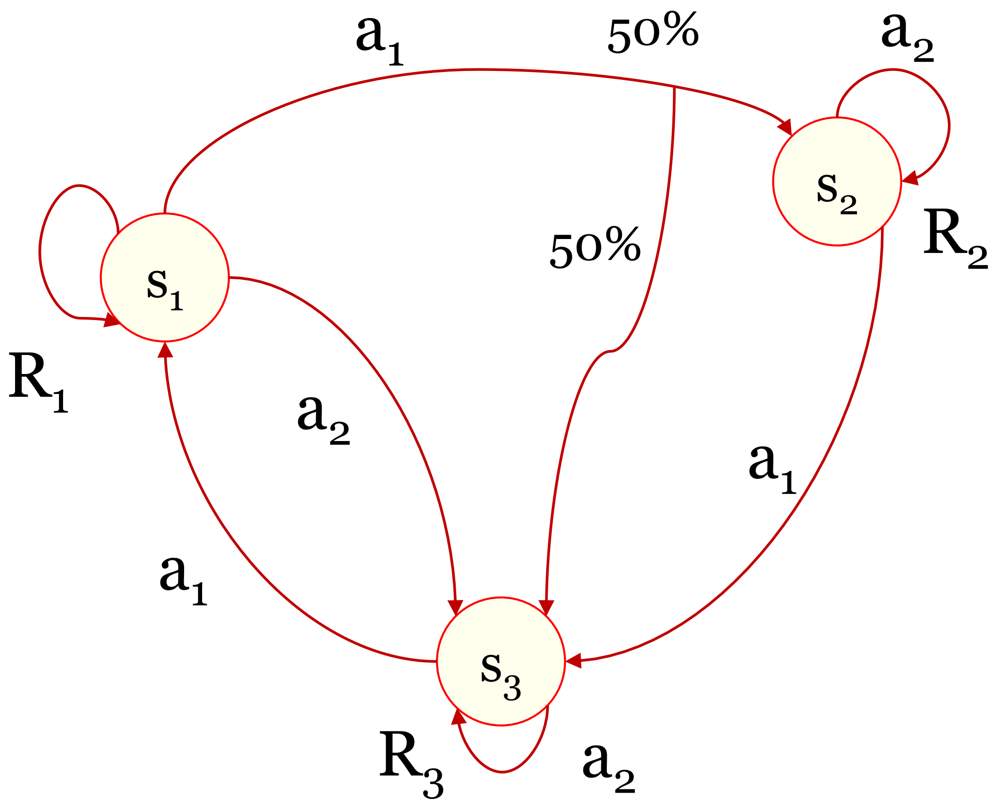
States: $s_1, s_2, \ldots s_n$

Actions: $a_1, a_2, \ldots a_n$

State transition matrix

$T(s,a,s') = p(s'|a,s)$, which is the probability of reaching state *s'* from state *s* after taking an action *a*.

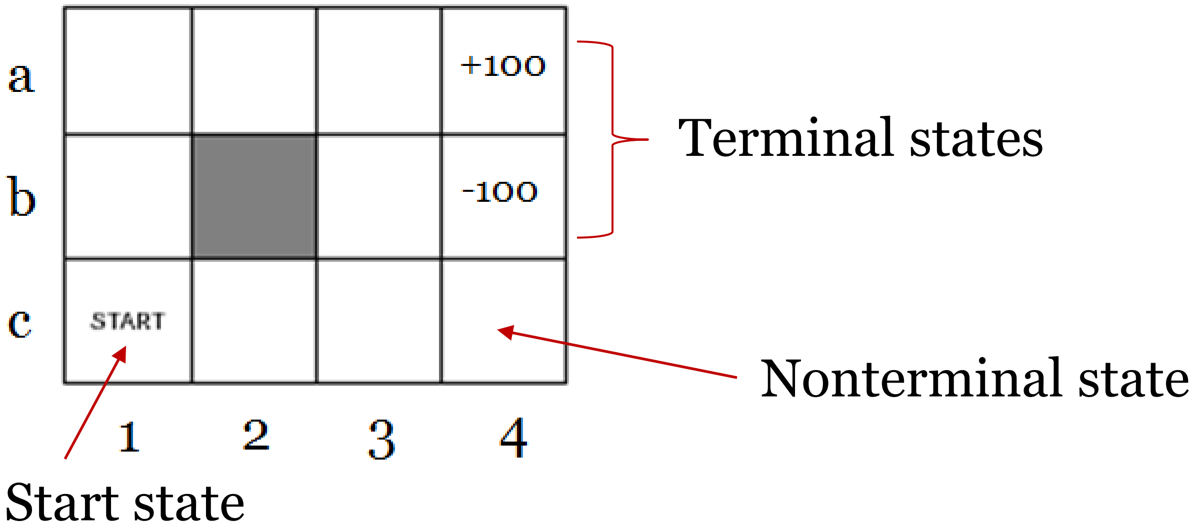
Reward function: $R(s)$



[1]

Markov Decision Process (MDP)

- **Grid World Example:**

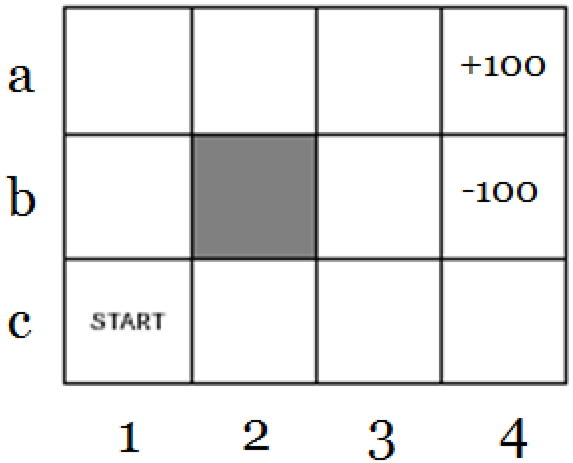


States: Eleven states ((a,4) and (b,4) are terminal states)

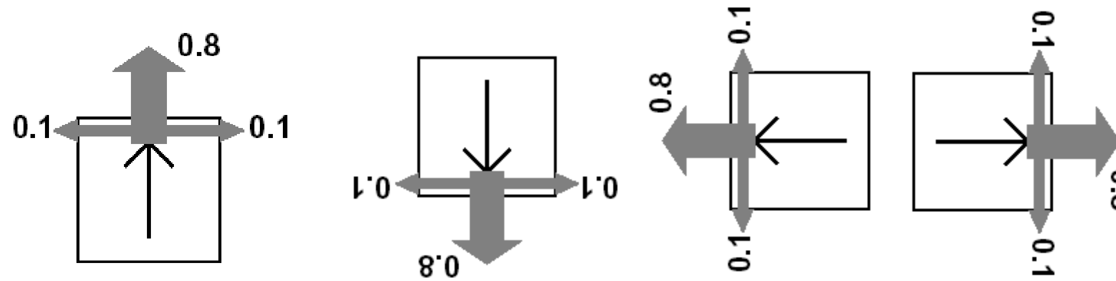
Environment is completely **observable**, i.e. observations give correct information about the state of the world.

Markov Decision Process (MDP)

- **Grid World Example (cont'd):**



Nondeterministic effects of actions



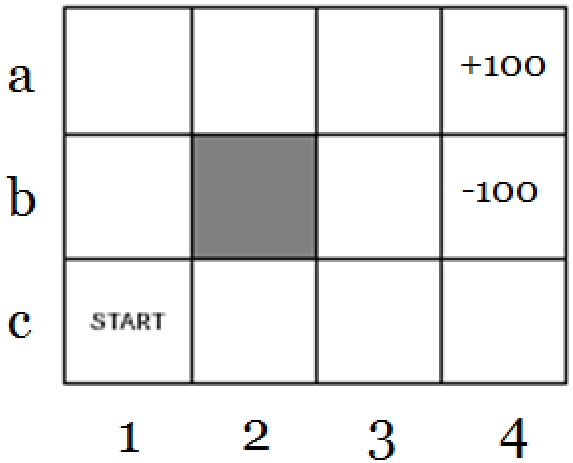
Note: If the actions are deterministic, probability the success of an action is 1

Actions: Agent moves in the above grid via stochastic actions Up, Down, Left, Right. Each action has:

- ◇ **0.8 probability** to reach its intended effect
- ◇ **0.1 probability** to move at right angles of the intended direction
- ◇ If the agents **bumps** into a wall, it says there.

Markov Decision Process (MDP)

- **Grid World Example (cont'd):**



Reward:

$$R(s)=\begin{cases} -3 & \text{(small penalty) for nonterminal states} \\ \pm 100 & \text{for terminal states} \end{cases}$$

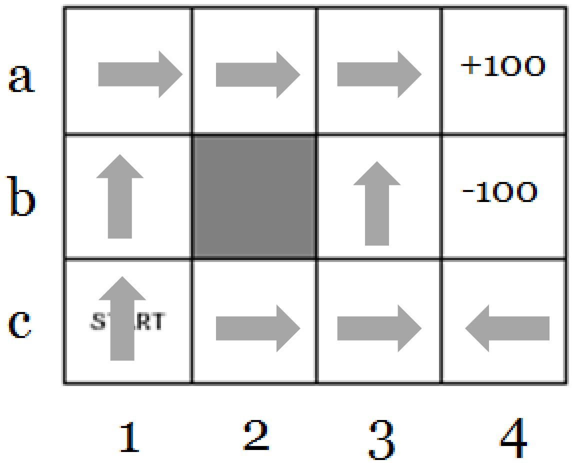
Total Reward: $E\left[\sum_{t=0}^{\infty} R_t\right]$

Discounted Rewards: $E\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$

where
γ is the **discount factor** that decays the **future rewards** relative to more **immediate reward**
γ=0.9

Markov Decision Process (MDP)

- **Grid World Example (cont'd):**



Policy:

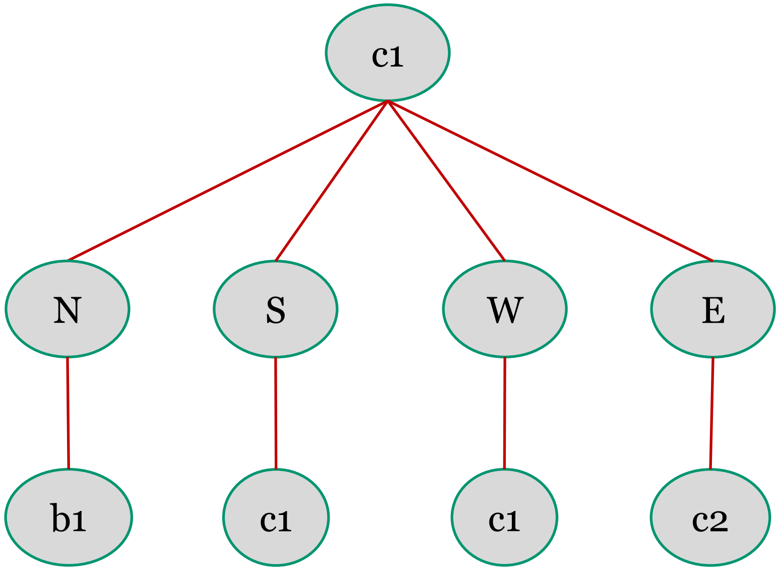
A policy for an MDP is a single decision function **π(s)** that specifies what the agent should do for each state **s**.

A policy assigns action to any state: $\pi(s): S \rightarrow A$

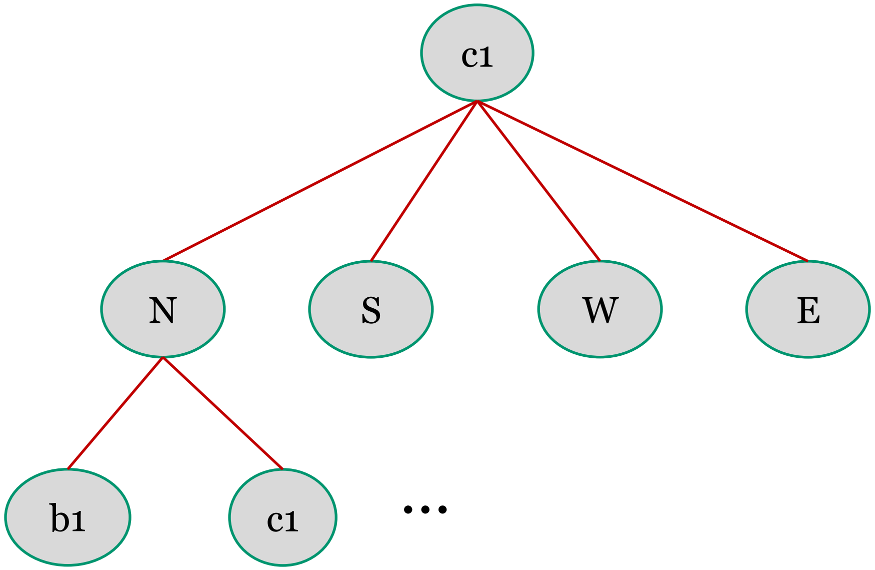
Planning problem is how to find an **optimal policy that maximizes the discounted rewards**

Markov Decision Process (MDP)

- **Grid World Example (cont'd):**



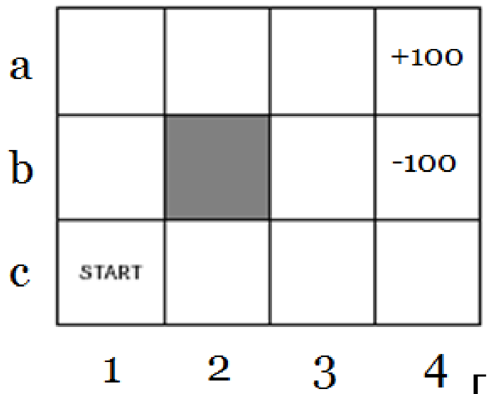
Deterministic Environment



Stochastic Environment

Why conventional planning such as A* cannot be used in stochastic environments?

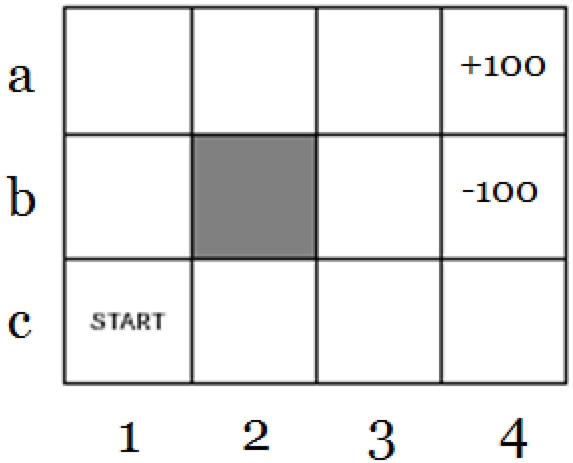
- ◇ Branching factor is large
- ◇ Tree is too deep
- ◇ Many states visited more than once



[1]

Markov Decision Process (MDP)

- **Grid World Example (cont'd): Value Iteration**



Value function given a policy π

$$V^{\pi}(s) = E_{\pi}\left[\sum_{t} \gamma^{t} R_{t} \mid s_{0} = s\right]$$

Value Iteration is used to recursively calculate the value function of each state

The expected value of following policy π in state *s*

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

Reward at state *s*

Reward value at *s'*

states reachable from *s* by doing a

Probability of getting to *s'* from s via *a*

Markov Decision Process (MDP)

- **Value Iteration Algorithm**

```
Procedure Value_Iteration(S,A,P,R,θ)
    Inputs
        S is the set of all states
        A is the set of all actions
        P is state transition function specifying p(s'|s,a)
        R is a reward function R(s,a,s')
        θ a threshold, θ>0
    Output
        π[S] approximately optimal policy
        V[S] value function
    Local
        real array V^π [S] is a sequence of value functions
        action array π[S]
```

[2]

Markov Decision Process (MDP)

- **Value Iteration Algorithm (cont'd)**

assign $V_0[S]$ arbitrarily

$k \leftarrow 0$

repeat

$k \leftarrow k+1$

for each state s do

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s,a) v(s') + R(s)$$

until $\forall s$ $\left| V^{\pi}(s) - v(s') \right| < \theta$

for each state s do

$$\pi(s) = \arg\max_{a} \sum_{s'} p(s' \mid s,a) V(s')$$

return π, V^{π}

[2]

Markov Decision Process (MDP)

• Grid World Example (cont'd): Value Iteration

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

a	0	0	0	+100
b	0		0	-100
c	START	0	0	-3
	1	2	3	4

Assume:

- ◇ Stochastic outcome of actions with 0.8 probability of success
- ◇ Initial expected values are zeros
- ◇ γ=1 for simplicity
- ◇ Rewards

$$R(s) = \begin{cases} -3 & \text{(small penalty) for nonterminal states} \\ \pm 100 & \text{for terminal states} \end{cases}$$

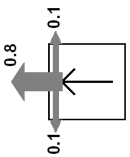
Markov Decision Process (MDP)

• Grid World Example (cont'd): Value Iteration

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s,a) v(s') + R(s)$$

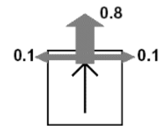
	1	2	3	4
a	0	0	0	+100
b	0		0	-100
c	START	0	0	-3

◇ **State a3: Moving West**



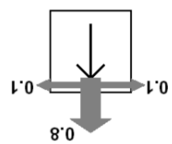
$$V^{W}(a3) = 0.8 \times 0 + 0.1 \times 0 + 0.1 \times 0 - 3 = -3$$

◇ **State a3: Moving North**



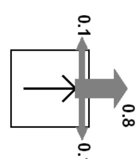
$$V^{N}(a3) = 0.8 \times 0 + 0.1 \times 0 + 0.1 \times 100 - 3 = 10 - 3 = 7$$

◇ **State a3: Moving South**



$$V^{S}(a3) = 0.8 \times 0 + 0.1 \times 0 + 0.1 \times 100 - 3 = 10 - 3 = 7$$

◇ **State a3: Moving East**



$$V^{E}(a3) = 0.8 \times 100 + 0.1 \times 0 + 0.1 \times 0 - 3 = 80 - 3 = 77$$

Markov Decision Process (MDP)

• Grid World Example (cont'd): Value Iteration

$$V^{\pi}(s) \leftarrow \max_a \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

a	0	0	77	+100
b	0		0	-100
c	START	0	0	-3
	1	2	3	4

◇ **State a3:**

$$V^{\pi}(a3) \leftrightarrow \max_a \left\{ V^{W}(a3), V^{N}(a3), V^{S}(a3), V^{E}(a3) \right\}$$

$$\leftarrow \max_a \left\{ -3, 7, 7, 77 \right\}$$

$$\leftarrow 77$$

Repeat for all the other states…

Quiz: calculate the values if R(s)=0 for all nonterminal states…

Markov Decision Process (MDP)

• Grid World Example (cont'd): Policy

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

$$\pi(s) = \arg\max_{a} \sum_{s'} p(s' \mid s, a) V(s')$$

Values after convergence

	1	2	3	4
a	85	89	93	+100
b	81		68	-100
c	77	72	70	47

	1	2	3	4
a	→	→	→	+100
b	↑		↑	-100
c	↑	→	→	←

[1]

Markov Decision Process (MDP)

- **Search and Rescue Example**
  - ◇ An **autonomous vehicle** in a search and rescue mission in an unknown environment that is affected by natural disaster or human-made disaster.
  - ◇ In every state, the robot must choose between **moving (M)** to collect more information from the environment or **staying (S)** to sample information remotely.

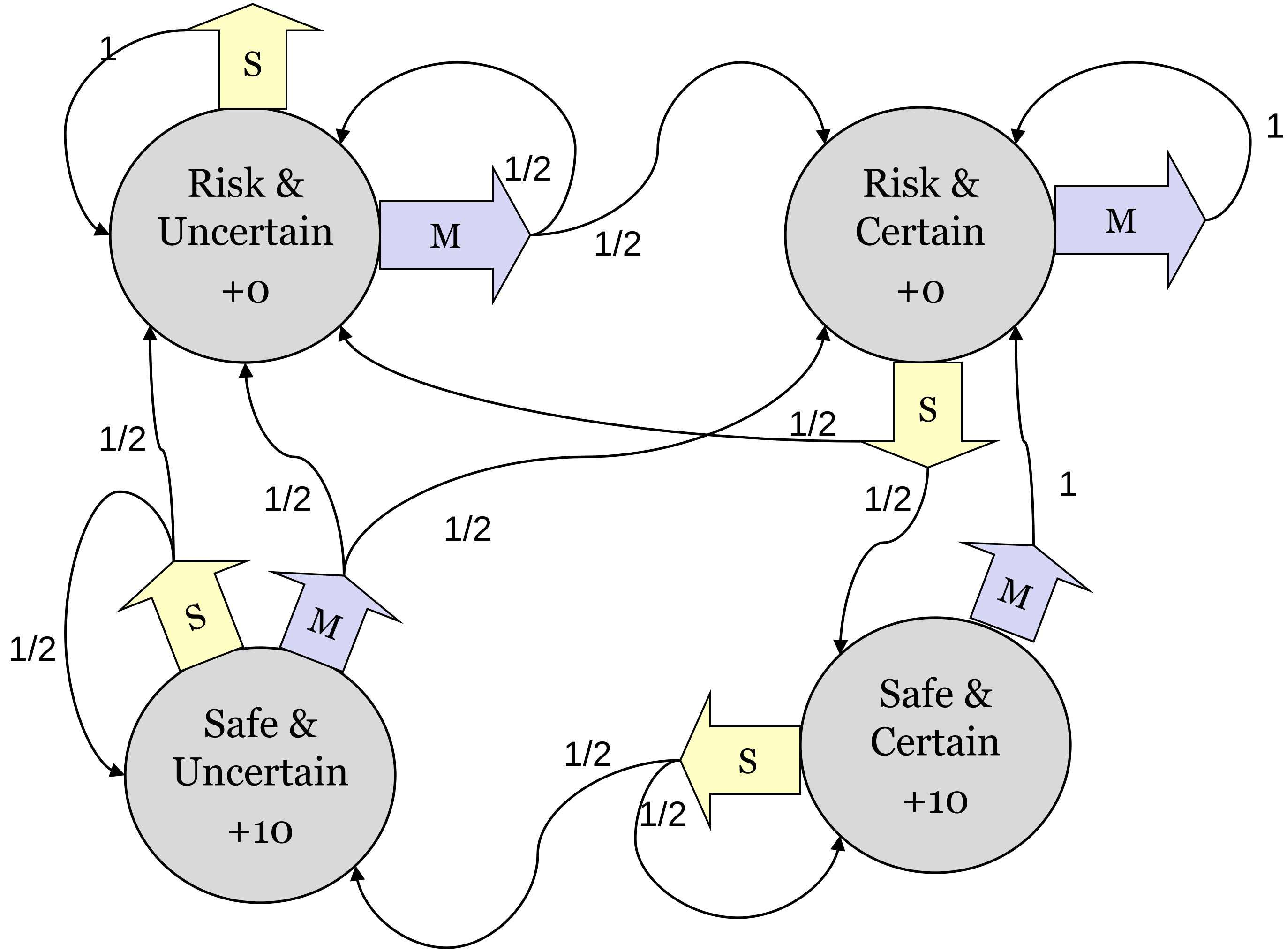
  - ◇ Moving may result in a **risk** on the robot but makes the robot more **certain**.
  - ◇ Staying makes the robot more **safe** but **uncertain** about the environment.

Markov Decision Process (MDP)

- **Search and Rescue Example**

  - ◇ Assume that

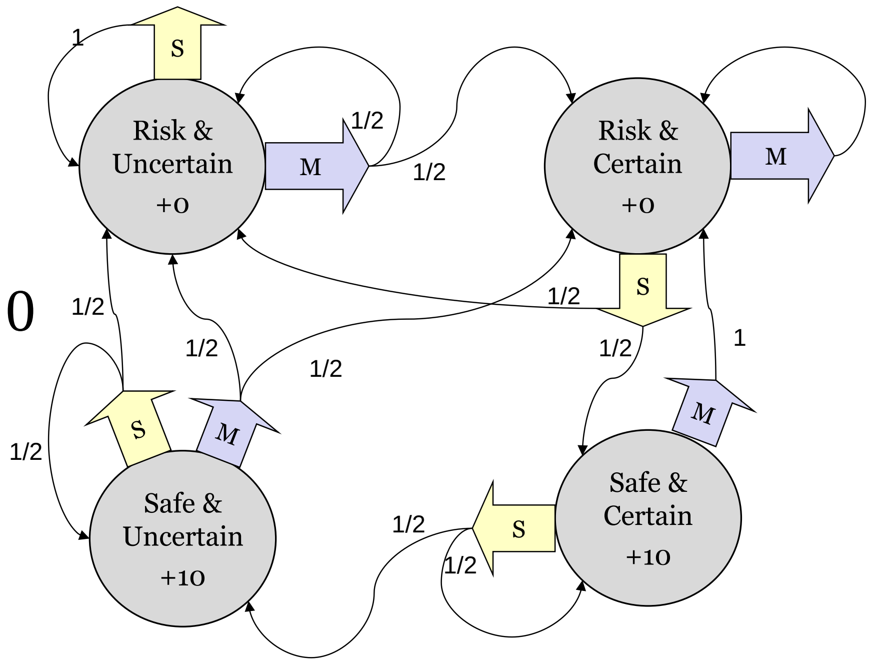
γ=0.9



Markov Decision Process (MDP)

- **Search and Rescue Example**

$$V^{\pi}(s) \leftarrow \max_a \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$



- ◇ **State RU: Move**

$$V^{M}(RU) = 0.9(0.5\times 0 + 0.5\times 0) + 0 = 0$$

- ◇ **State RU: Stay**

$$V^{S}(RU) = 0.9(1\times 0) + 0 = 0$$

- ◇ **State RU:**

$$V^{\pi}(RU) \leftrightarrow \max_a \{V^{M}(RU), V^{S}(RU)\}$$

$$\leftarrow \max_a \{0,0\}$$

$$\leftarrow 0$$

Markov Decision Process (MDP)

- **Search and Rescue Example**

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

◇ **State RC: Move**

$$V^{M}(RC) = 0.9(1 \times 0) + 0 = 0$$

◇ **State RC: Stay**

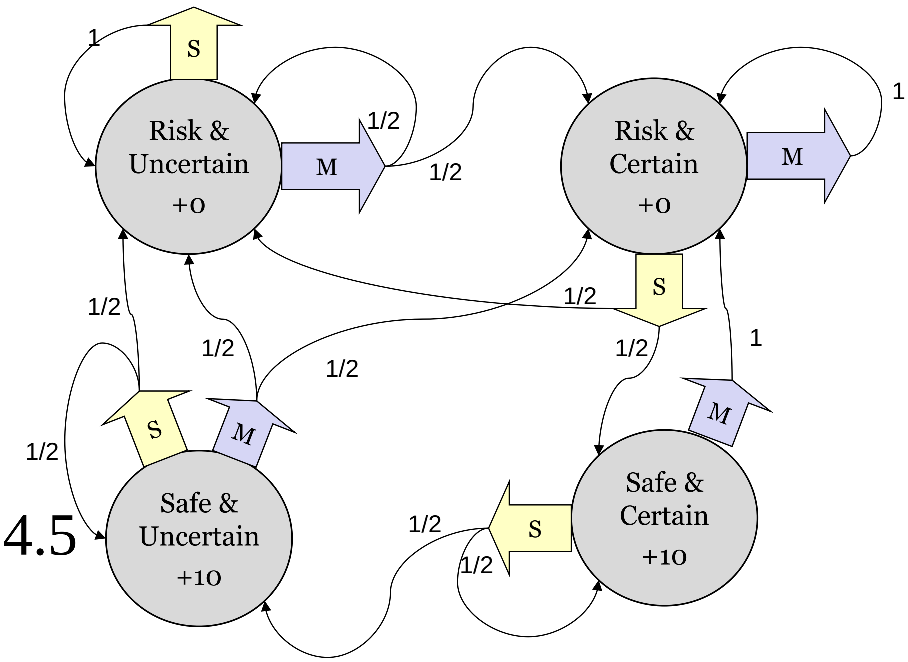
$$V^{S}(RC) = 0.9(0.5 \times 0 + 0.5 \times 10) + 0 = 4.5$$

◇ **State RC:**

$$V^{\pi}(RC) \leftrightarrow \max_{a} \{V^{M}(RC), V^{S}(RC)\}$$

$$\leftarrow \max_{a} \{0, 4.5\}$$

$$\leftarrow 4.5$$



Markov Decision Process (MDP)

- **Search and Rescue Example**

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

◇ **State SC: Move**

$$V^{M}(SC) = 0.9(1 \times 0) + 10 = 10$$

◇ **State SC: Stay**

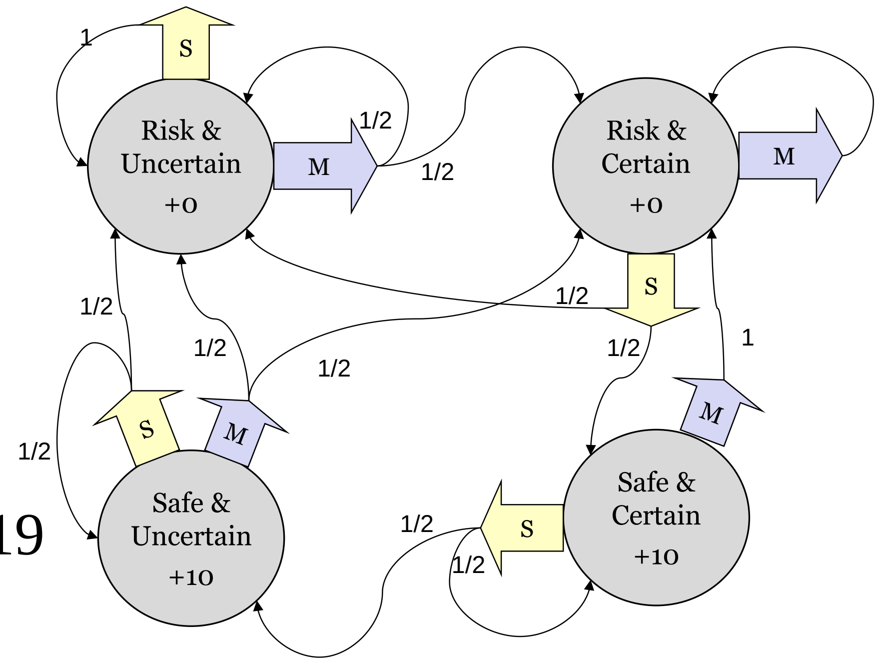
$$V^{S}(SC) = 0.9(0.5 \times 10 + 0.5 \times 10) + 10 = 19$$

◇ **State SC:**

$$V^{\pi}(SC) \leftrightarrow \max_{a} \{V^{M}(SC), V^{S}(SC)\}$$

$$\leftarrow \max_{a} \{10, 19\}$$

$$\leftarrow 19$$



Markov Decision Process (MDP)

- **Search and Rescue Example**

$$V^{\pi}(s) \leftarrow \max_{a} \gamma \sum_{s'} p(s' \mid s, a) v(s') + R(s)$$

- ◇ **State SU: Move**

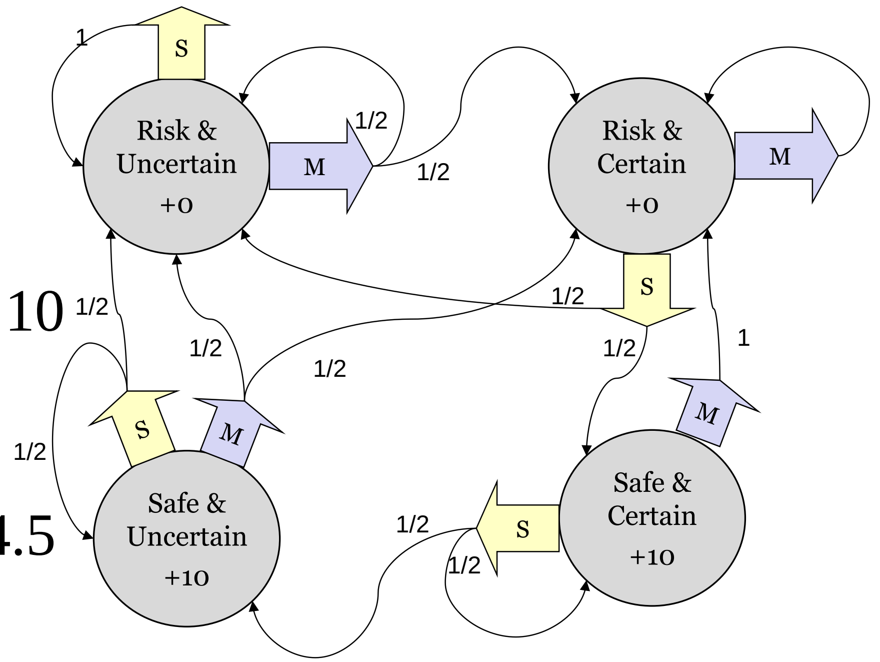
$$V^{M}(SU) = 0.9(0.5 \times 0 + 0.5 \times 0) + 10 = 10$$

- ◇ **State SU: Stay**

$$V^{S}(SU) = 0.9(0.5 \times 0 + 0.5 \times 10) + 10 = 14.5$$

- ◇ **State SU:**

$$V^{\pi}(SU) \leftrightarrow \max_{a} \{V^{M}(SU), V^{S}(SU)\}$$
$$\leftarrow \max_{a} \{10, 14.5\}$$
$$\leftarrow 14.5$$



Markov Decision Process (MDP)

- **Search and Rescue Example**

Summary of first iteration

$$\pi(s) = \arg\max_a \sum_{s'} p(s' \mid s, a) V(s')$$

State	RU	RC	SC	SU
V	0	4.5	19	14.5
π(s)	M/S	S	S	S

Repeat a number of iterations until convergence (small change in the values of the states)