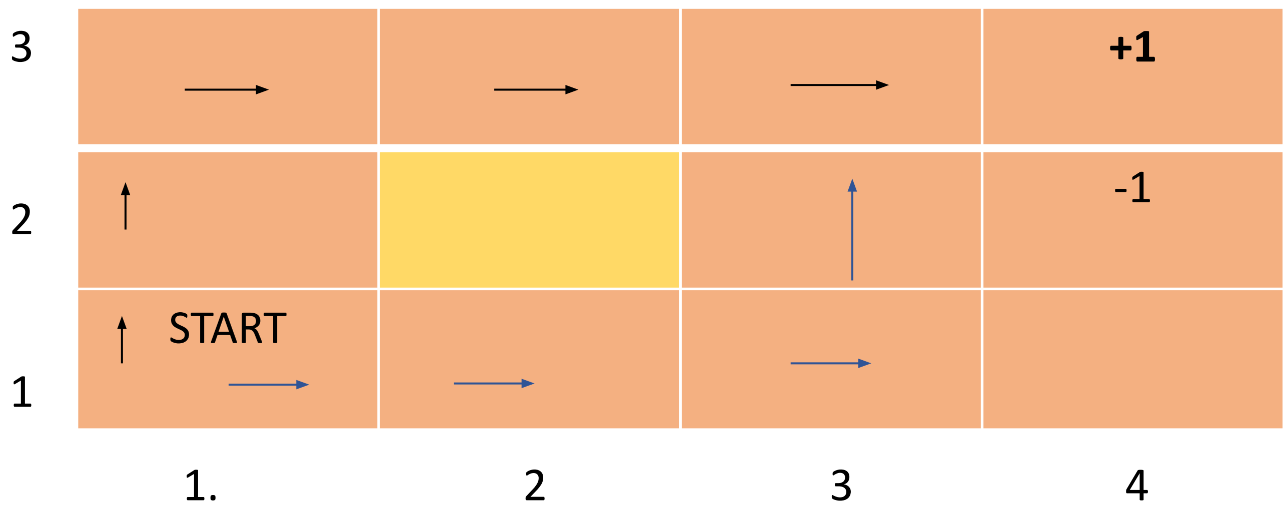
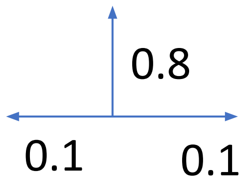
Artificial Intelligence-2
(CSL 7040)

Lecture 9 : Making Complex Decisions

Sequential Decision Making





R(s)= -0.04

Set of actions ={UP, DOWN, RIGHT, LEFT}

Set of Intended Actions ={UP, UP, RIGHT, RIGHT, RIGHT}

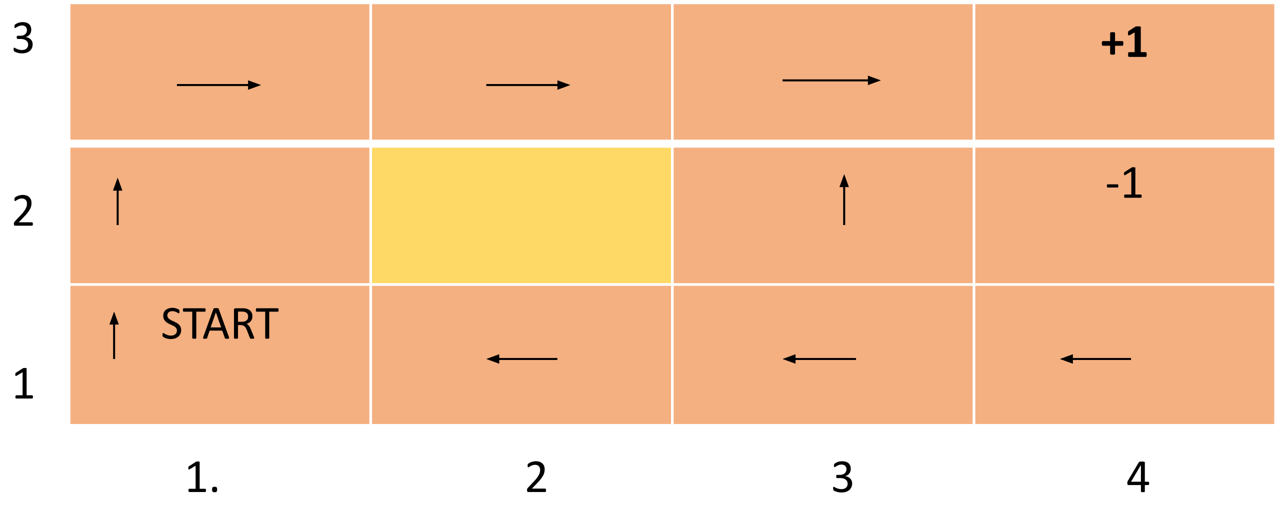
Probability of reaching state +1 only by taking intended actions= 0.8^5=0.32768

0.1^4*0.8

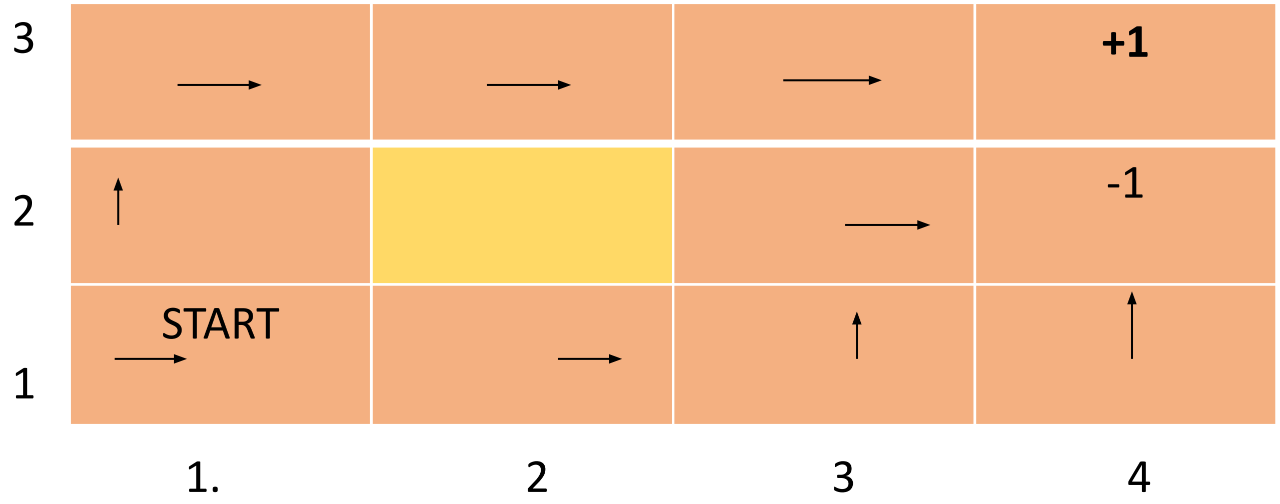
Total probability of reaching +1=0.327+0.1^4*0.8=0.3277

Markovian Decision Process and Policy

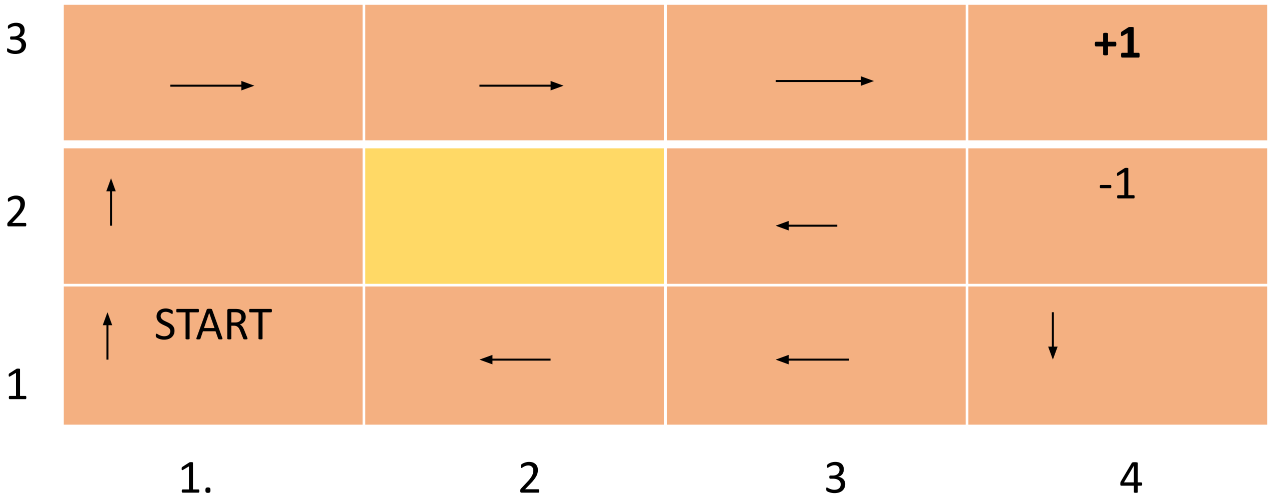
- A sequential decision problem in fully observable environment
  - Set of states
  - Set of ACTIONS(s) in each state
  - Transition model $P(s'|s,a)$
  - Reward function R(s)
- Policy: The solution what an agent should do in a particular state
- $\pi(s) \rightarrow Action\ recommended\ in\ state\ s$
- Quality of the Policy: EU of all the possible environment history
- Optimal Policy: The policy that generates the highest EU (π^*)



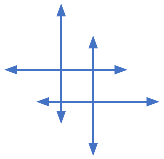
R(s)= -0.04

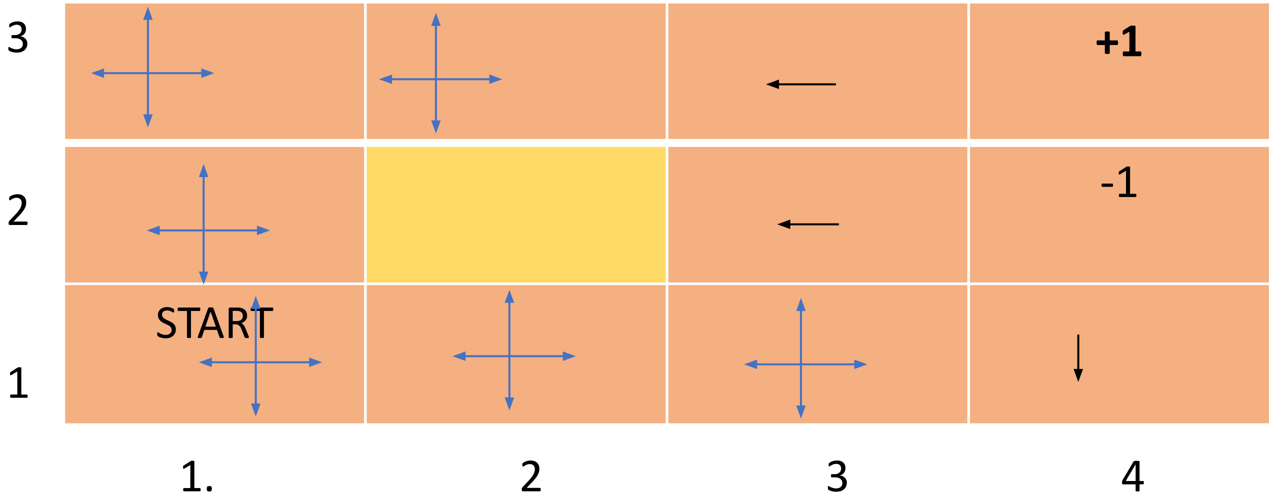


R(s)= -1.8



-0.2<R(s)<0





R(s)= 0.5

Utilities over Time

- $U_h([s_0, s_1, \ldots., s_{N+k}]) = U_h([s_0, s_1, \ldots., s_N]) \; \forall k > 0$
- Optimal policy in finite horizon is non-stationary
- We are dealing here with infinite horizon → don't have any fixed deadline → MDP to have one terminal state

Stationary Preference:

The preference between $[s_0, s_1, \ldots.]$ and $[s_0', s_1', \ldots.]$ if $s_0 = s_0'$ then

Is equivalent to the preference between $[s_1, s_2, \ldots.]$ and $[s_1', s_2', \ldots.]$

Assigning utility to preference

- Additive Reward:

$$U_h([s_0, s_1, s_2, \dots)] = R(s_0) + R(s_1) + R(s_2) + \cdots$$

- Discounted Reward:

$$U_h([s_0, s_1, s_2, \dots)] = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

$$\gamma \rightarrow discount\ factor: 0 \leq \gamma \leq 1$$

$$discount\ factor \equiv interest\ rate\ (\frac{1}{\gamma} - 1)$$

Discount factor

- If there is no terminating state in the environment → history is going to be infinitely long → utility with additive reward = + \infinity → difficult to handle
- Solution??
  1. Set $\gamma < 1$

     $$U_h([s_0, s_1, \dots]) = \sum_{t=0}^{infinity} \gamma^t R(s_t) \leq \sum_{t=0}^{infinity} \gamma^t R_{max} = R\_\max/(1-\gamma)$$
  2. Should chose a policy that guarantees to reach a terminal state → Proper policy
  3. Infinite sequence could be compared in terms of average reward obtained per time step.

Optimal policies and Utilities of the States

- Assume s→ Initial state; s_t→ random variable: agent reaches here at time t after executing the policy π
- EU by executing the policy π:

$$U^{\pi}(s) = E[\sum_{t=0}^{\propto} \gamma^t R(s_t)]$$

Expectation w.r.t. probability distribution over state sequences determined by s and π

$$\pi_s^* = \arg\max_{\pi} U^{\pi}(s)$$

Discounted utilities with finite horizon → optimal policy is independent of the starting state. Actions can't be independent → policy function specify action for each state

- π_a^* and π_b^* those should not disagree with another optimal policy π_c^* $\rightarrow$ single policy π^*

True Utility of a State

- $U^{\pi^*}(s)$→ Expected sum of discounted rewards after executing optimal policy

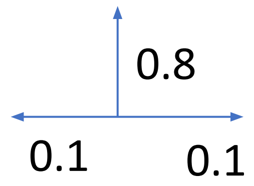
- $R(s) \rightarrow Short\ term\ reward\ for\ being\ in\ the\ sate\ s$
- $U(s) \rightarrow long - term\ total\ reward\ from\ s\ onwards$

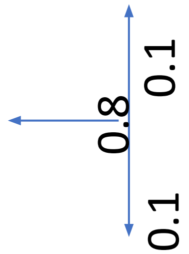
$$\pi^*(s) = argmax_{a \in \mathrm{A}(s)} \sum_{s'} P(s' | s \quad , a) U(s')$$

Value Iteration

- To calculate an optimal policy □ calculate utilities in each state and use the state utilities to select an optimal action in each state

3	**0.812**	**0.868**	**0.918**	**+1**
2	0.762		0.660	-1
1	START 0.705	0.655	0.611	0.338
	1	2	3	4





Bellman Equation for Utilities:

- Utility of a state = Immediate reward for the state + expected discounted utility of the next state, assuming the agent will take the optimal action
- $U(s) = R(s) + \gamma \max_{a \in \mathrm{A}(s)} \sum_{s\prime} P(s'|s,a) U(s')$

$U(1,1) = -0.04 + \gamma \max[\ 0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1), \rightarrow Up,$
$0.9U(1,1)0.1\mathrm{U}(1,2), \mathrm{Left} \qquad 0.9\mathrm{U}(1,1) +$
$0.1(2,1) \rightarrow \mathrm{Down}, \qquad 0.8\mathrm{U}(2,1) + 0.1\mathrm{U}(1,2) + 0.1\mathrm{U}(1,1) \rightarrow \mathrm{Right}]$

Value Iteration Algorithm

- $U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in \mathrm{A}(s)} \sum_{s'} P(s'|s,a) U_i(s')$

Function: Value-iteration

Partially Observable MDPs (POMDPs)

- $b(s)$ →Probabilities assigned to the actual state s by the belief state b
- Prev. belief state could change depending upon the action (a) and evidence (e).

$$b'(s') = \alpha P(e|s') \sum_{s} P(s'|s,a)b(s)$$

3	**Y**	**X**	**Z**	**+1**
2	A			-1
1	START	B		
	1	2	3	4

Partially Observable MDPs (POMDPs)

- $b' = FORWARD\ (b, a, e)$
- POMDP → Optimal action depends only on agent's current belief→ optimal policy $\pi^*(b)$
- $P(e|a, b) = \sum_{s'} P(e|a, s', b)P(s'|a, b) = \sum_{s'} P(e|s')P(s'|a, b) = \sum_{s'} P(e|s') \sum_s P(s'|s, a)b(s$

- Probability of reaching b' from b with action a
- $P(b'|b, a) = P(b'|a, b) = \sum_e P(b'|e, a, b)P(e|a, b) = \sum_e P(b'|e, a, b) \sum_{s'} P(e|s') \sum_s P(s'|s, a)b(s)$
- Decision cycle:
  - Given current state, execute $a = \pi^*(b)$
  - Receive the evidence e
  - Set current belief FORWARD(b,a,e) and repeat
- Reward function of belief states= $\rho(b) = \sum_s b(s)R(s)$
- The POMDP is boiling down to an MDP in belief space instead of state space