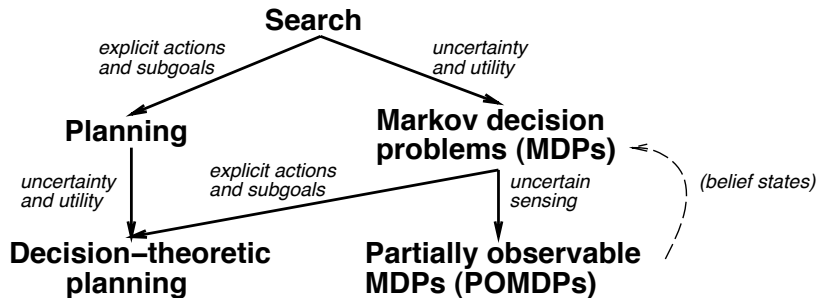
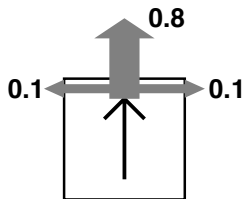
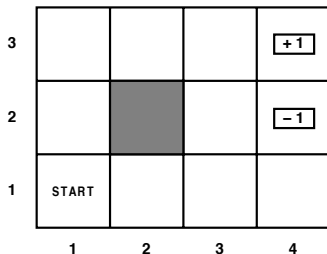


Sequential Decision Problems



Markov Decision Process

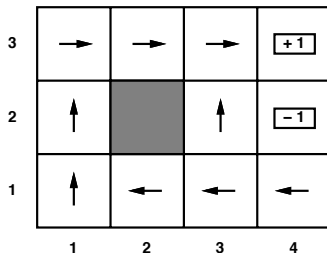


- States $s \in S$, actions $a \in A$
- Model $T(s, a, s') \equiv P(s'|s, a)$
- Reward function $R(s)$ (or $R(s, a)$, $R(s, a, s')$)

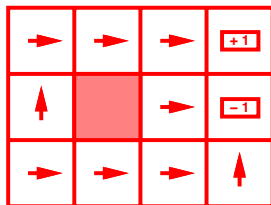
$$R(s) = \begin{cases} -0.04 & \text{(small penalty) for nonterminal states} \\ \pm 1 & \text{for terminal states} \end{cases}$$

Solving MDPs

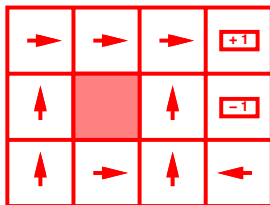
- In search problems, aim is to find an optimal *sequence*
- In MDPs, aim is to find an optimal *policy* $\pi(s)$
 - Best action for every possible state s
 - Cannot predict where one will end up
- Optimal policy maximizes *expected sum of rewards*
- Optimal policy when state penalty $R(s)$ is -0.04 :



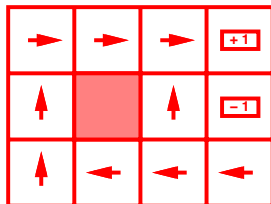
Reward and Optimal Policy



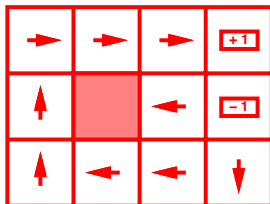
$r = [-\infty : -1.6284]$



$r = [-0.4278 : -0.0850]$



$r = [-0.0480 : -0.0274]$



$r = [-0.0218 : 0.0000]$

Utility of State Sequences

- Need to understand preferences between *sequences* of states
- Typically consider stationary preferences on reward sequences

$$[r, r_0, r_1, r_2, \dots] \succ [r, r'_0, r'_1, r'_2, \dots] \Leftrightarrow [r_0, r_1, r_2, \dots] \succ [r'_0, r'_1, r'_2, \dots]$$

- Theorem: Only two ways to combine rewards over time:
 - 1) *Additive* utility function:

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

- 2) *Discounted* utility function: For discount factor γ

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

Utility of States

- Utility $U(s)$ of a *state* (a.k.a. its *value*)
 - Expected (discounted) sum of rewards (until termination)
 - Assume optimal actions
- Choosing the best action is just MEU
 - Maximize the expected utility of the immediate successors
 - Utilities of the states are given

3	0.812	0.868	0.912	+1
2	0.762		0.660	-1
1	0.705	0.655	0.611	0.388
	1	2	3	4

3	→	→	→	+1
2	↑		↑	-1
1	↑	←	←	←
	1	2	3	4

The Optimal Policy

- Given a policy π , the overall (discounted) utility

$$U_\pi = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

- U_π a random variable as s_t are random
- Optimal policy corresponds to the MEU

$$\pi^* = \operatorname{argmax}_\pi E[U_\pi] = \operatorname{argmax}_\pi E \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi \right]$$

Dynamic Programming: The Bellman Equation

- Simple relationship among utilities of neighboring states
- Expected sum of rewards = current reward
+ $\gamma \times$ Expected sum of rewards after taking best action
- Bellman equation (1957):

$$U(s) = R(s) + \gamma \max_a \sum_{s'} U(s') T(s, a, s')$$

- One equation per state s
 - n nonlinear equations in n unknowns

Value Iteration Algorithm

- Main Idea
 - Start with arbitrary utility values
 - Update to make them locally consistent
 - Everywhere locally consistent \Rightarrow Global optimality
- Repeat for every s simultaneously until “no change”

$$U(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} U(s') T(s, a, s') \quad \forall s$$

