Lecture 28

CS 6260 Automated Planning and Learning

Markov Decision Processes

Department of Computer Science and Engineering

Indian Institute of Technology Madras

Some material based on lecture slides for ASU's CSE571 Artificial Intelligence by Siddharth Srivastava.

Logistics

Logistics

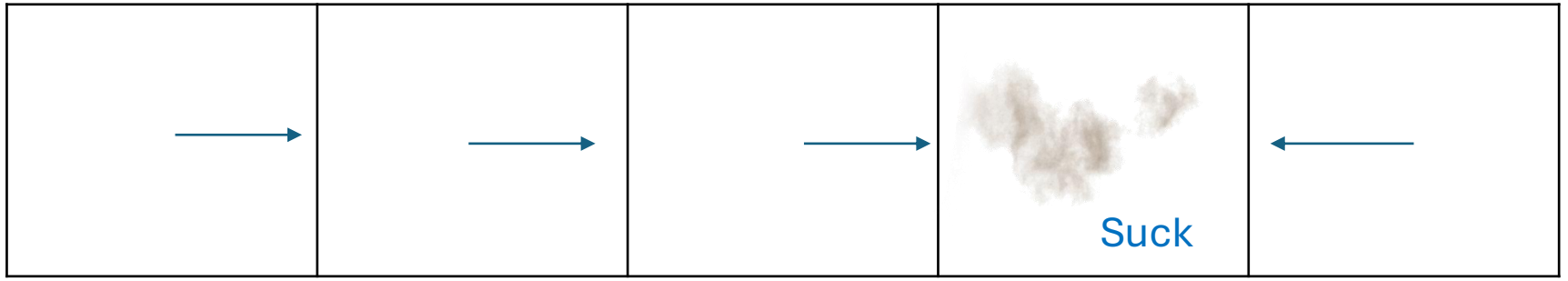
- Office Hours: 1 hour after class every week Tu, W, Th
- ROS based Assignment: Covered in Class on Tuesday Apr 21
  - https://wiki.ros.org/ROS/Tutorials
- Extra Class tomorrow: Friday, Apr 17
  - Class Time: 05:00 – 06:50 PM

MDPs

Forms of Solutions

- Deterministic, fully observable
  - "Classical planning": solution is a plan, or a sequence of actions
- Deterministic, non-observable
  - "Conformant planning": solution is a plan, but often no solutions exist
- Non-deterministic, fully observable:
  - "Contingent planning": solution can be a policy (a function $S \rightarrow A$)

Non-determinism: a move action may fail

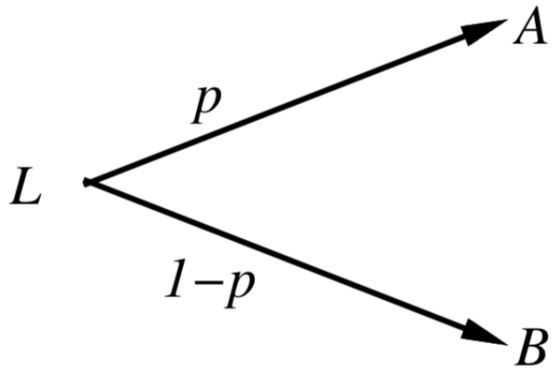


Ingredients for Planning Under Uncertainty With Better Guarantees

- Quantitative information about the environment
  - Analytical model or simulator model
  - What is the probability of dirt in each cell?
  - What is the probability that the robot's wheels slip?
- Better information about the desired states or goals
  - What is the desired state?

Key Required Ingredients

- To talk about desirability of outcomes, we need a way to express preferences
- Let A, B be different situations
  - E.g.:
    - A: Here's a million bucks!
    - B: Toss a coin; Heads$\mapsto 0$; Tails$\mapsto 2.5\ million$
- Notation
  - $A \succ B$: A is preferred to B
  - $A \sim B$: indifference between A and B
  - $A \succsim B$: B is not preferred to A
  - Lottery $L = [p, A; 1-p, B]$



Good News!

- Theorem (Ramsey '31, von Neumann and Morgenstern '44):
  - Given preferences satisfying the constraints (see lecture 12), there exists a real-valued function U such that

$$U(A) \geq U(B) \text{ iff } A \gtrsim B \text{ and}$$

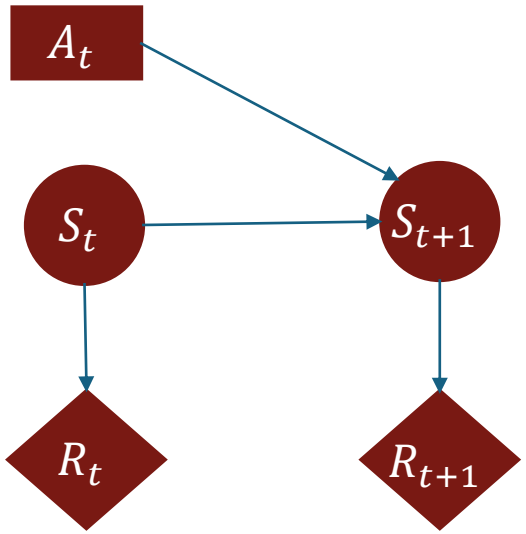
$$U([p_1, S_1; \ldots; p_n, S_n]) = \sum_i p_i U(S_i)$$

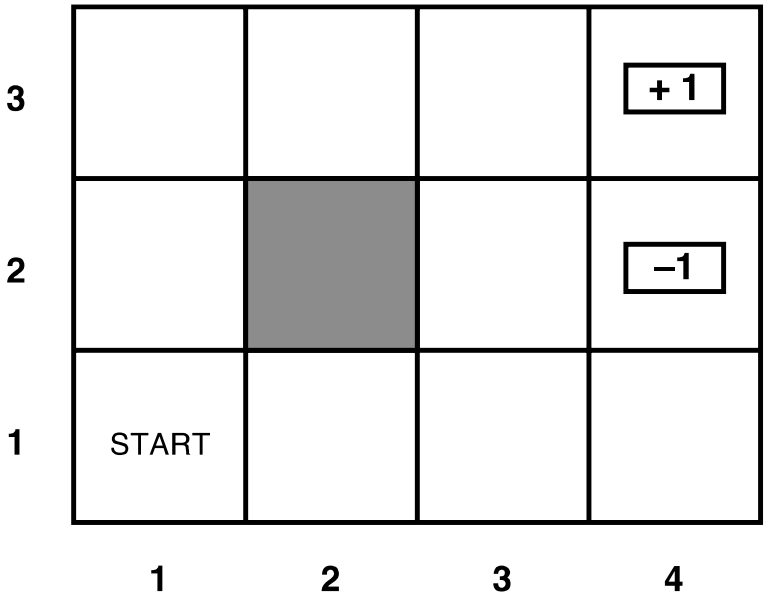
- This is useful:
  - We can use the maximum expected utility in situations where outcomes are not clear!
  - We do need probabilities to be able to do so

- Note: An agent may be perfectly rational without knowing about probabilities or utilities (e.g., use a lookup table!)

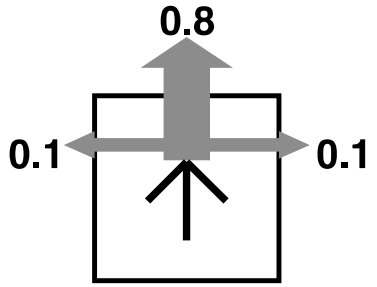
Using Utility Functions to Specify Sequential Decision Problems

- Immediate rewards (utilities)
  - Doing nothing: -0.4
  - Two terminal states: +1, -1
- Transition model is Markovian, stationary





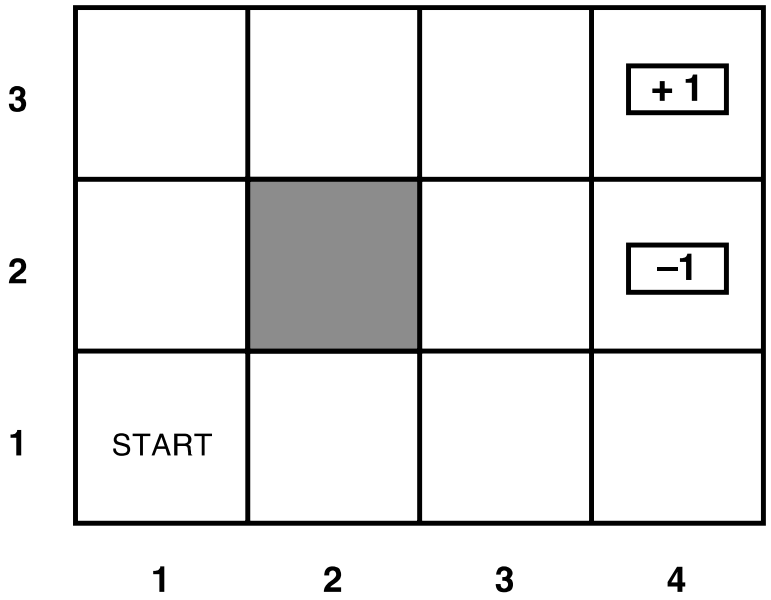
(a)



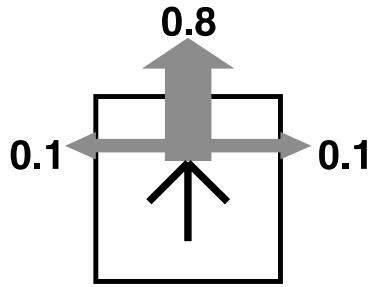
(b)

Markov Decision Processes

- S set of states
  - At(1,1)
- A set of actions
- P transition model
  - $P(s'|s,a)$
- $R: S \rightarrow \mathbb{R}$ reward, or utility function



(a)

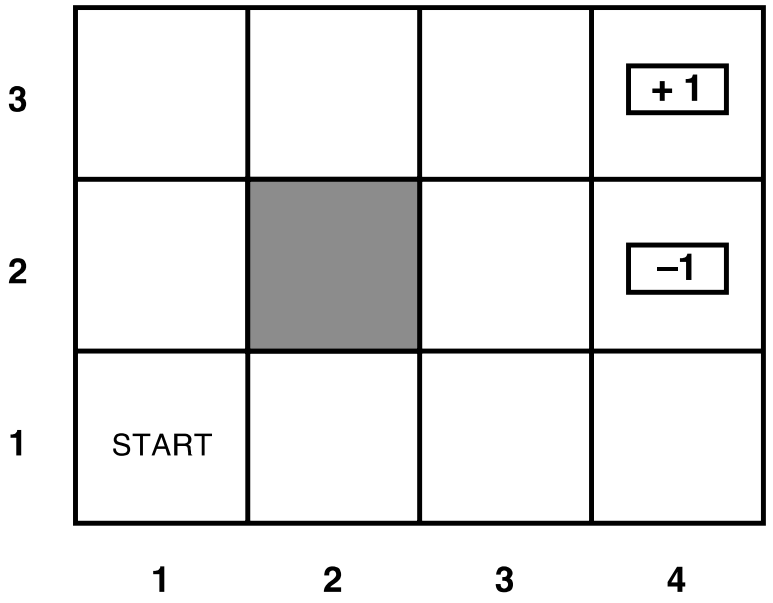


(b)

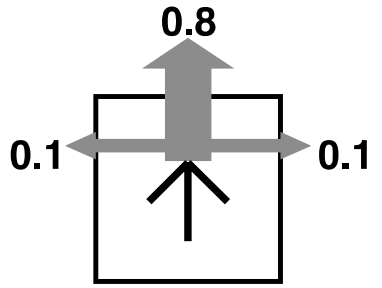
- Agent can "drift", end up in unintended states
- Solutions take the form of policies

Markov Decision Processes

- Policy: $S \rightarrow A$
- Every state is mapped to an action
- How should we evaluate a policy?
  - Evaluate the executions it may produce
- Need a function that determines the utility of an execution history
- Candidates:
  - Sum of utilities of states
  - Max state-utility in the execution
  - Min state-utility in the execution



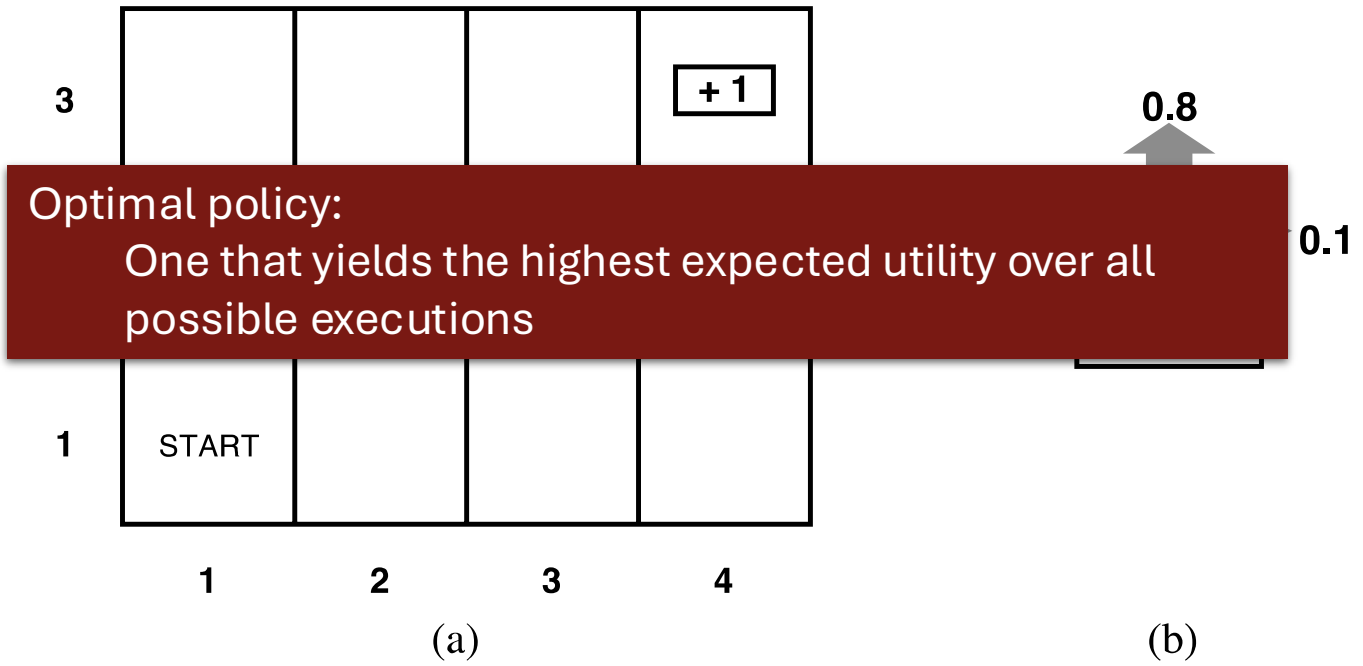
(a)



(b)

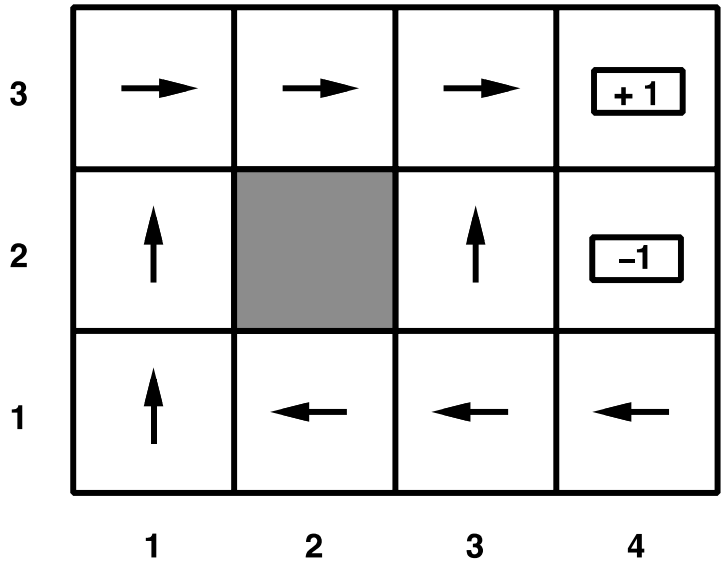
Markov Decision Processes

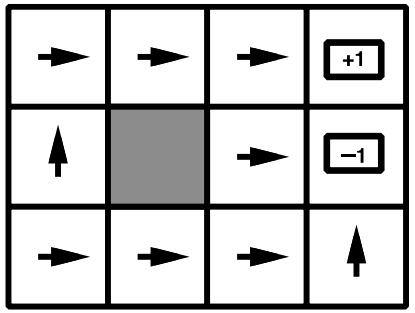
- Policy: $S \rightarrow A$
- Every state is mapped to an action
- How should we evaluate a policy?
  - Evaluate the executions it may produce
- Need a function that determines the utility of an execution history
- Candidates:
  - Sum of utilities of states
  - Max state-utility in the execution
  - Min state-utility in the execution



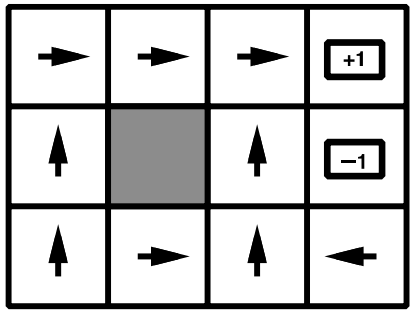
Optimal policy:
One that yields the highest expected utility over all possible executions

Optimal Policies (Utility=Sum of Rewards)

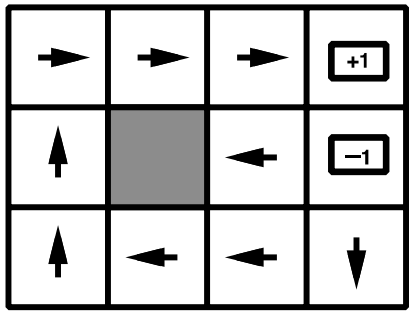




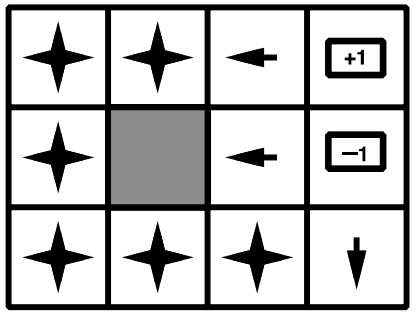
$R(s) < -1.6284$



$-0.4278 < R(s) < -0.0850$



$-0.0221 < R(s) < 0$



$R(s) > 0$

Horizon, Utilities and Policies

- Finite horizon: agent has a fixed time N after which the execution "ends"
  - $U([s_1, \dots s_{N+k}]) = U([s_1, \dots, s_N])$

- In such cases, optimal action in a state can depend on time
  - Optimal policy is *non-stationary*
  - E.g., optimal actions in a timed exam

- Infinite horizon: optimal action at a state is independent of time
  - Policies for infinite horizon problems are simpler

Horizon, Utilities and Policies

- However, we expect the agent's utility function to be stationary:
  - Preference between state sequences that have the same prefix depend only on the non-common parts
    - $[s_0, s_1, \ldots, s_n]$ vs $[s_0, s'_1, \ldots, s'_n]$
    - $U([s_0, s_1, \ldots, s_n]) > U([s_0, s'_1, \ldots, s'_n])$ iff $U([s_1, \ldots, s_n]) > U([s'_1, \ldots, s'_n])$

- The desirability of a situation is independent of how far into the future it occurs
- Note: it does depend on the states!

Stationary Utility Functions

- Stationary preferences lead to only two possible ways to assign utilities to sequences

  - Additive rewards
    - $U([s_0, \ldots, s_n]) = R(s_0) + \cdots + R(s_n)$
    - Problem: infinite horizon… infinite reward
    - How would we compare solutions?

  - Discounted rewards ($0 \leq \gamma \leq 1$)
    - $U([s_0, \ldots, s_n]) = R(s_0) + \gamma R(s_1) \ldots + \gamma^n R(s_n)$
    - Utility of an infinite sequence (bounded reward) is guaranteed to converge!

Stationary Utility Functions

- Stationary preferences lead to only two possible ways to assign utilities to sequences

  - Additive rewards
    - $U([s_0, \ldots, s_n]) = R(s_0) + \cdots + R(s_n)$
    - Problem: infinite horizon… infinite reward
    - How would we compare solutions?

  - Discounted rewards ($0 \leq \gamma \leq 1$)
    - $U([s_0, \ldots, s_n]) = R(s_0) + \gamma R(s_1) \ldots + \gamma^n R(s_n)$
    - Utility of an infinite sequence (bounded reward) is guaranteed to converge!

Solving MDPs

- Expected utility of using a policy π starting in s
  - Let $S_1, \ldots S_k, \ldots$ be the RVs denoting states generated at that timestep
  - $U^{\pi}(s) = E[\sum_t \gamma^t R(S_t)]$

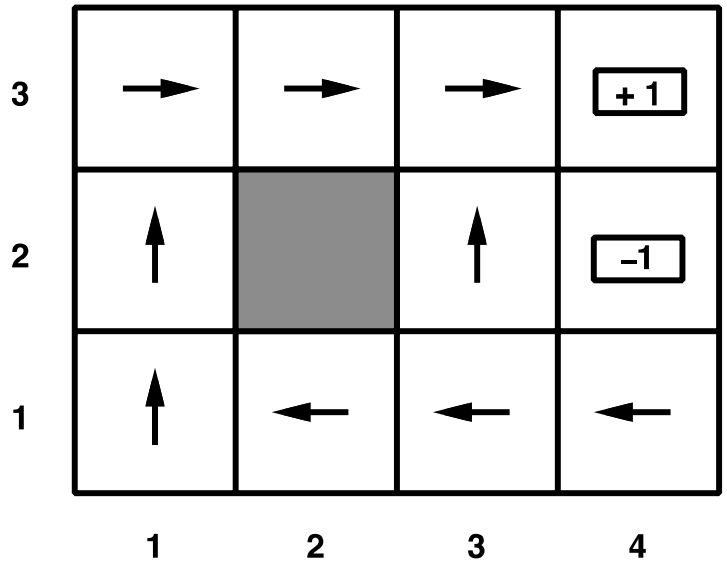
- We want the "best" policy
  - $\pi^*(s) = argmax_{\pi} U^{\pi}(s)$

  Give me the policy that does the best, starting from s

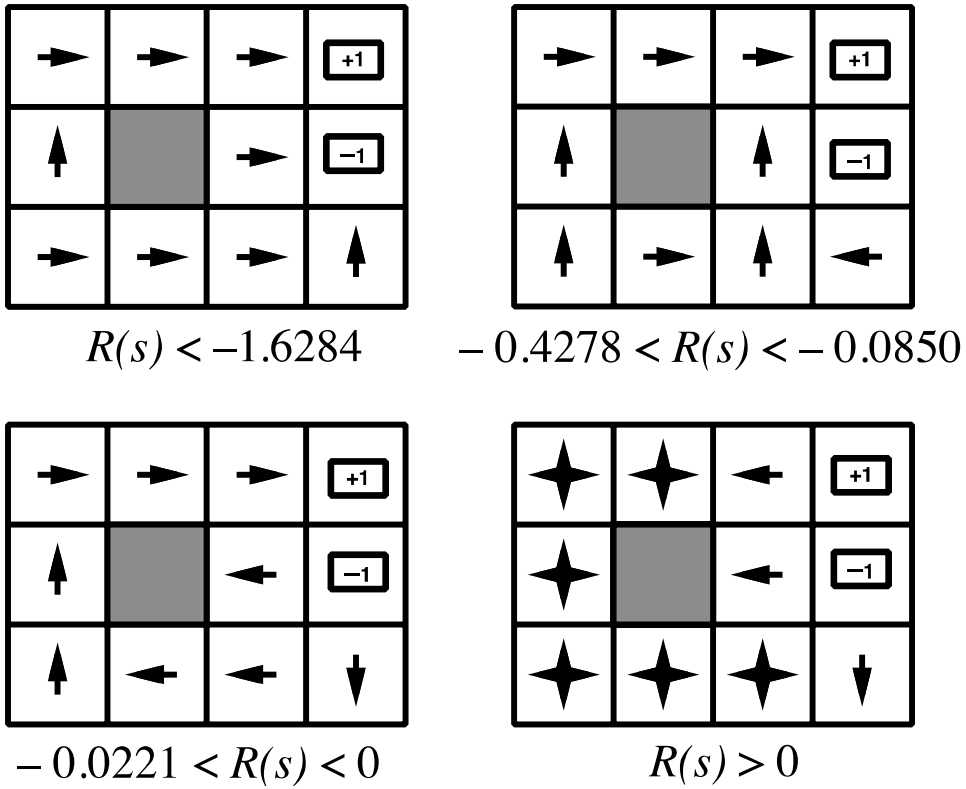
- Should the optimal action depend on the timestep?

Solving MDPs

- Should the optimal action depend on the timestep?



(a)



(b)

Solving MDPs

- Expected utility of using a policy π starting in s
  - Let $S_1, \ldots S_k, \ldots$ be the RVs denoting states generated at that timestep
  - $U^{\pi}(s) = E[\sum_t \gamma^t R(S_t)]$

- We want the "best" policy
  - $\pi^*(s) = argmax_{\pi} U^{\pi}(s)$

  Give me the policy that does the best, starting from s

- For infinite horizon problems, π^* is independent of the starting state
  - i.e., the optimal action at a state doesn't depend on what you did before reaching it!

- $V(s)$: utility of a state under the optimal policy