AIL 722: Reinforcement Learning

Lecture 6: Markov Decision Processes

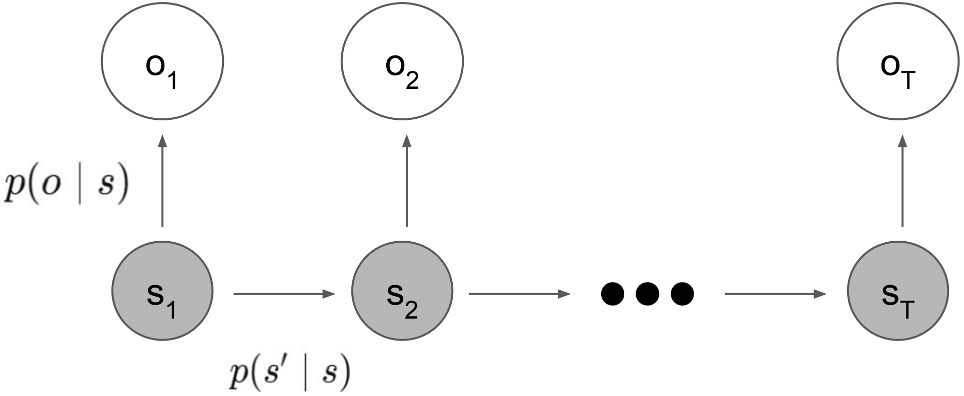
Raunak Bhattacharyya

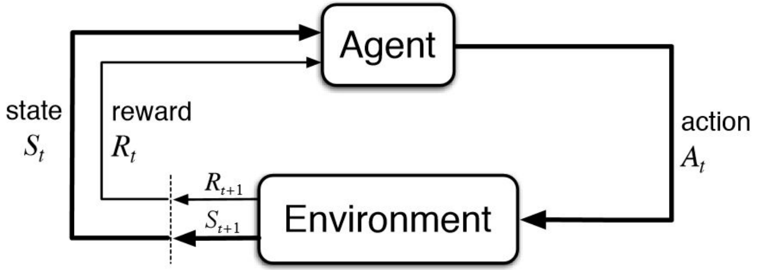
ScAI YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Outline

- MDP recap
- Example applications
- Closer look at a few examples
- Associated constructs

HMM: State Evolution





Source: [Sutton & Barto]

MDP

$$\text{MDP} : \text{Tuple}\langle \mathcal{S}, \mathcal{A}, T, R, \rho \rangle$$

$\mathcal{S}$: State Space

$\mathcal{A}$: Action Space

$T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$: Probabilistic Transition Function

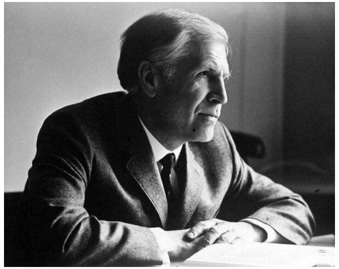
$R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$: Reward Function

ρ : Initial State Distribution

$$s_t, a_t, r(s_t, a_t)$$

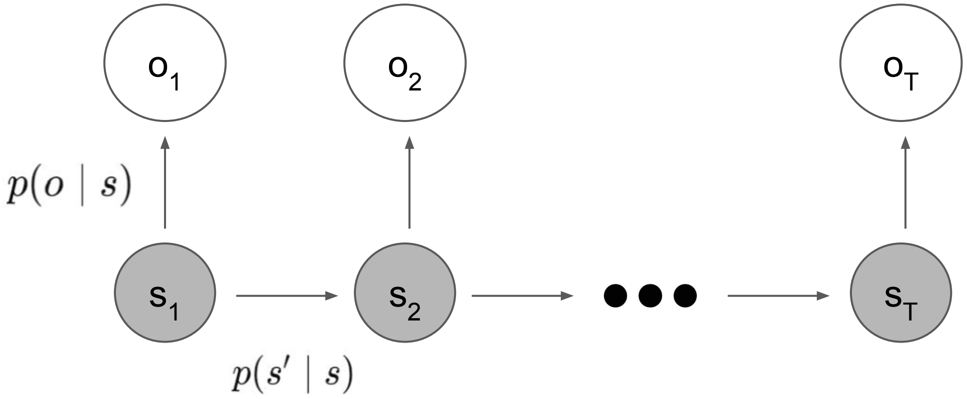
Richard Bellman, Source: [Wikipedia]

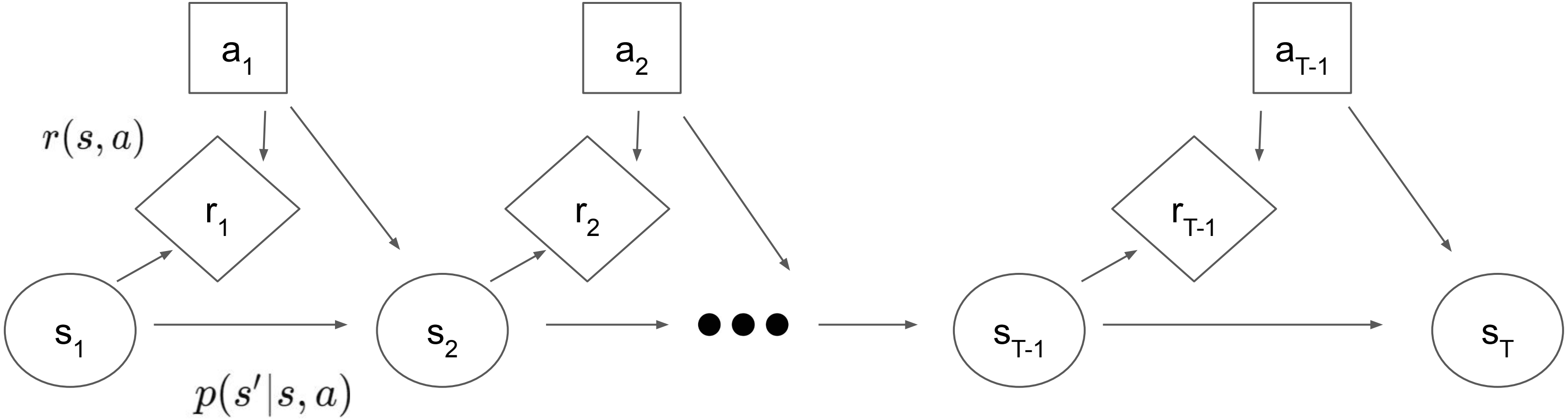
$$x_t, u_t, c(x_t, u_t)$$



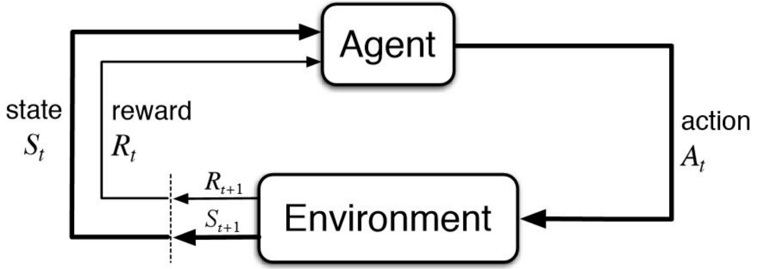
Lev Pontryagin, Source: [Wikipedia]

MDP: State Evolution





MDP Framework



Source: [Sutton & Barto]

- An abstraction for goal-directed behavior
- Whatever the details of sensors, memory and control
  - Any problem of learning goal-directed behavior can be reduced to three signals passing back and forth between an agent and its environment:
    - Represent choices made by the agent (the actions)
    - Represent basis on which choices are made (the states)
    - Define the agent's goal (the rewards)

Example Applications

- Cleaning Robot

Source: [Youtube]

Example Applications

- Cleaning Robot
- Walking Robot

Source: Youtube

Example Applications

- Cleaning Robot
- Walking Robot
- Pole balancing

Source: Youtube

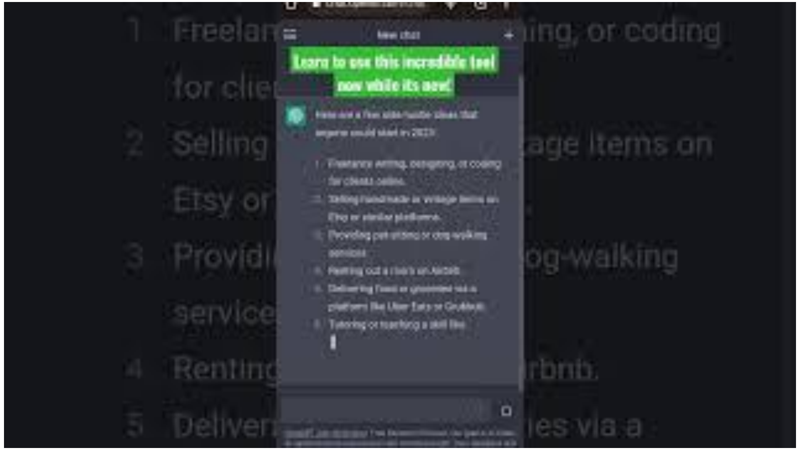
Example Applications

- Cleaning Robot
- Walking Robot
- Pole balancing
- Games: Tetris

Source: [Youtube]

Example Applications

- Cleaning Robot
- Walking Robot
- Pole balancing
- Games: Tetris
- Language: Dialog Systems



Source: [Youtube]

Example Applications

- Cleaning Robot
- Walking Robot
- Pole balancing
- Games: Tetris
- Language: Dialog Systems
- Computer Vision: Object Tracking

Source: [Youtube]

Example Applications

- Cleaning Robot
- Walking Robot
- Pole balancing
- Games: Tetris
- Language: Dialog Systems
- Computer Vision: Object Tracking
- Vision + Language: Image Captioning

Source: Youtube

Example Applications

- Cleaning Robot
- Walking Robot
- Pole balancing
- Games: Tetris
- Language: Dialog Systems
- Computer Vision: Object Tracking
- Vision + Language: Image Captioning
- Server Management

About the Reward

- A way for you to specify what you want the agent to achieve…
    - NOT *how* you want it achieved

- The reward hypothesis

That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

From Reinforcement Learning: An Introduction, Sutton & Barto

Closer Look at Example MDPs

Tetris: MDP Components

- State:
  - Board configuration
  - Shape of block (tetromino)
- Action: Placement
- Reward: Number of rows eliminated
- Dynamics:
  - Wall change
  - Random next tetromino

- Board is 10x20. And every square could be filled/not filled

Queuing Problem

Source: Dreamstime

- Customers line up in a queue. There is only one line. Line is empty initially
- We can serve one customer at a time. There are two modes of service: fast and slow
- Each timestep, a new customer arrives with probability p. The horizon length is T
- Waiting cost: gamma * queue length

Queuing Problem: Formulation

$$\mathcal{S} = \{0, 1, 2, \ldots\} : \text{Length of the queue } x_t \qquad x_0 = 0$$

$$\mathcal{U} = \{\text{Fast (F)}, \text{Slow (S)}\} \qquad \text{Completion probs: } q(F) > q(S)$$

$$c(x_t, u_t) = \gamma x_t + d(u_t) \qquad \text{Service costs: } d(F) > d(S)$$

If $x = 0$:

$$p(x' = 1 \mid x = 0, u = F/S) = p$$

$$p(x' = 0 \mid x = 0, u = F/S) = 1 - p$$

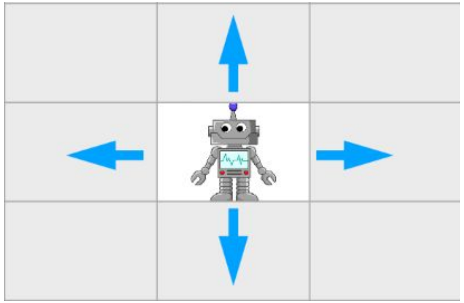
If $x > 0$:

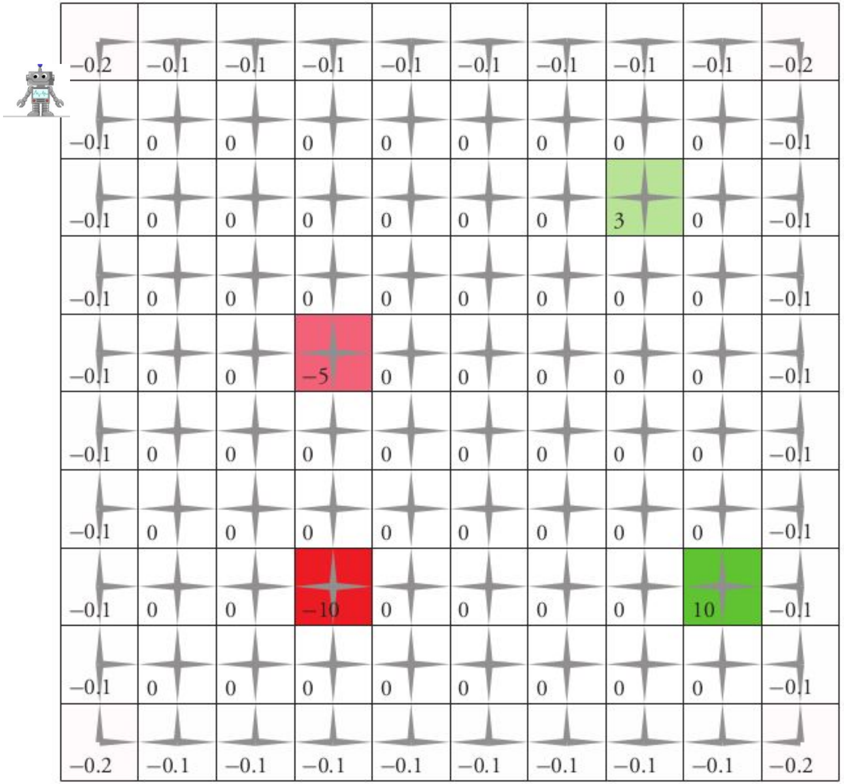
$$p(x' = x + 1 \mid x, u) = p \cdot (1 - q(u))$$

$$p(x' = x \mid x, u) = (1 - p) \cdot (1 - q(u)) + p \cdot q(u$$

$$p(x' = x - 1 \mid x, u) = q(u) \cdot (1 - p)$$

Grid World

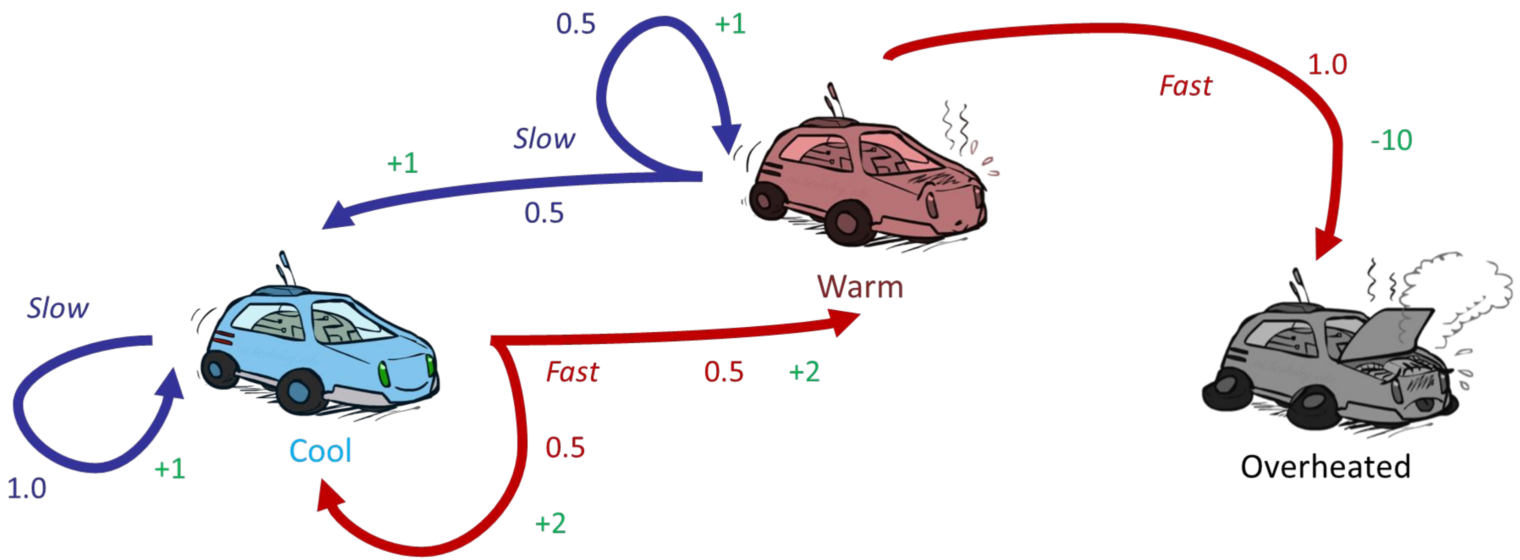




- 10x10 grid
- Up, down, left, right
- 0.7 **correct** dir (as instructed), 0.1 rest
- Green cells are absorbing (end state)

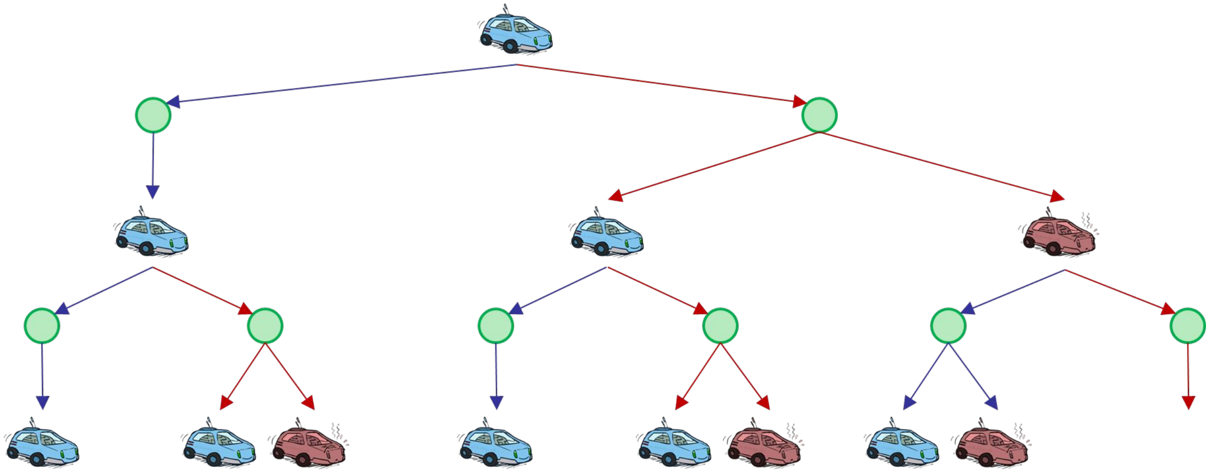
Figure from Algorithms for Decision Making, Mykel Kochenderfer

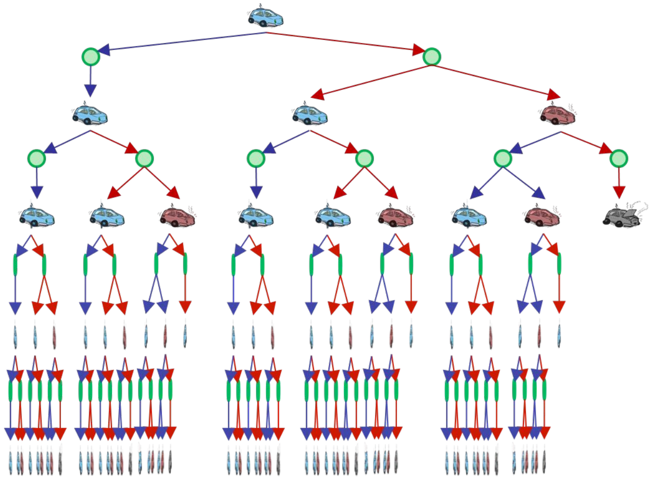
Racing Problem



Example from [CS580 at George Mason](), Instructor: Xuesu Xiao

Racing Problem





What is a Solution?

Search problem: path (sequence of actions)

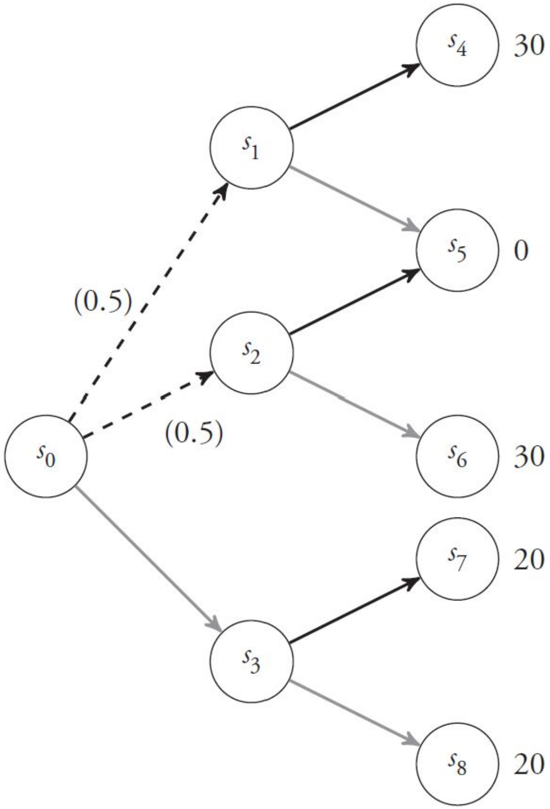
MDP:

Definition: policy

A **policy** π is a mapping from each state $s \in$ States to an action $a \in$ Actions(s).

From Stanford CS 221, Instructor: Percy Liang

Open Loop Plan



$$U(\text{up, up}) = 0.5 \times 30 + 0.5 \times 0 = 15$$
$$U(\text{up, down}) = 0.5 \times 0 + 0.5 \times 30 = 15$$
$$U(\text{down, up}) = 20$$
$$U(\text{down, down}) = 20$$

Open loop plan chooses down action from s_0

From Decision Making under Uncertainty, Mykel Kochenderfer