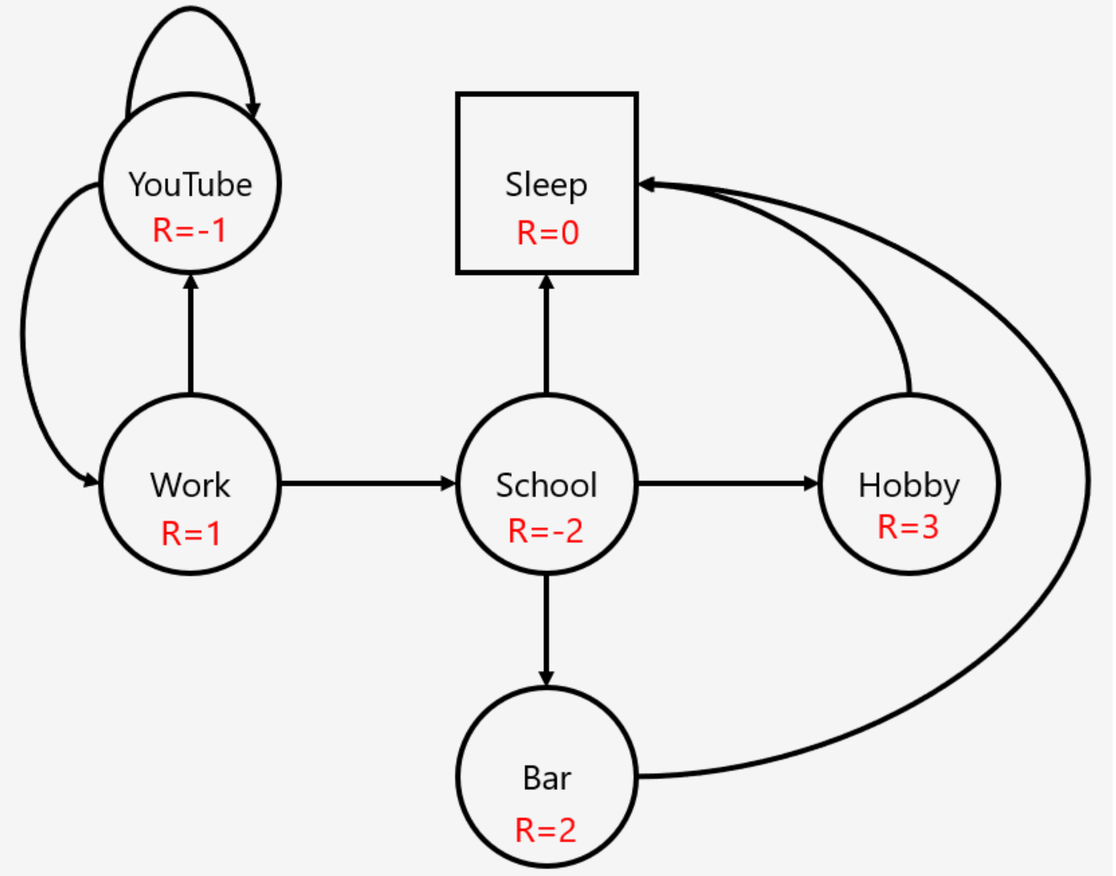
Markov Decision Processes

Markov Decision Process

- Incorporates decision making into HMMs.
  - Add “actions”
  - Add “rewards”
  - Find a “control policy” – triggers an action at any given state resulting in behaviors that maximize the “total reward” collected over time (such reward maximizing control policy is called “optimal”).

- We will formalize MDP in this lecture
- We will describe an algorithm that computes an optimal control policy.



A Markov Decision Process describing a college student's hypothetical situation.

https://optimization.cbe.cornell.edu/index.php?title=Markov_decision_process

MDP - formalized

A Markov Decision Process (MDP) is a problem described by a tuple $M = (S, A, P, R, \gamma)$, where

- S is a finite set of states
- A is a finite set of actions
- $P : S \times A \times S \to [0, 1]$ is transition probability distribution, i.e., $P(s'|s, a)$ is the probability that the system transitions into state s' when the system is in state s and action a is taken.
- $R : S \to \mathbb{R}$ is a reward function;

and the problem is to find a policy $\pi : \mathrm{S} \to A$ such that

$$V_\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0 = s\right] \qquad V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s_t) \mid s_0 = s\right]$$

is maximized for all $s \in S$.

**In some formulations, Reward R may depend on the state and the action chosen at that state: R(s,a) - as in the example in the previous slide*

*** This of γ as a number strictly between 0 and 1 that is applied to each state transition and discounts the future rewards when compared to near term rewards*

Can you formulate an MDP for a 3x3 grid world?

- An agent is traveling in the grid world. It can move up, down, left, right.
- Transition probabilities:
  - 80% chance of moving in the intended direction
  - 10% chance of veering to the left
  - 10% chance of veering to the right
- Reward:
  - Would like to penalize getting into the slippery area in the center
  - Would like to reward getting into the goal area
  - What else?

- What are the states?
- What are the actions?
- What is the reward function?
- What does the policy look like?

Value Iteration Solves the MDP: Finite Horizon Case

- How does this algorithm? Why does it solve our problem?

Let us define the value function $V_k^*(\cdot)$ as follows:

$$V_k^*(s) := \max_{\pi} \mathbb{E}\left[\sum_{t=k}^{H} \gamma^{t-k} R(s_t) \,|\, s_k = s\right]$$

Value Iteration with Finite Horizon

$V_H(s) \leftarrow 0$ for all $s \in S$;
for $k = H-1, H-2, \ldots, 1, 0$ **do**
  for *all* $s \in S$ **do**
    $V_k(s) \leftarrow \max_{a \in A} \left[R(s) + \gamma \sum_{s' \in S} P(s'|s,a) V_{k+1}(s') \right]$;

Value Iteration Solves the MDP: Infinite Horizon Case

$$V_\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0 = s\right]$$

- How does this algorithm? Why does it solve our problem?

Value Iteration

$V_0(s) \leftarrow 0$ for all $s \in S$;
for $k = 1, 2, \ldots$ **do**
  for *all* $s \in S$ **do**
    $V_{k+1}(s) \leftarrow \max_{a \in A} \left[R(s) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right]$;

We have $\lim_{k \to \infty} V_k(s) = V^*(s)$ for all $s \in S$.

Back to the College Student Example

$$V^*(s) = max_a[R(s,a) + \gamma \sum_{s' \epsilon S} P(s'|s,a)V^*(s')]$$

$$V^*(Hobby) = max_a[3 + (1)(1.0 * 0)] = 3$$

$$V^*(Bar) = max_a[2 + 1(1.0 * 0)] = 2$$

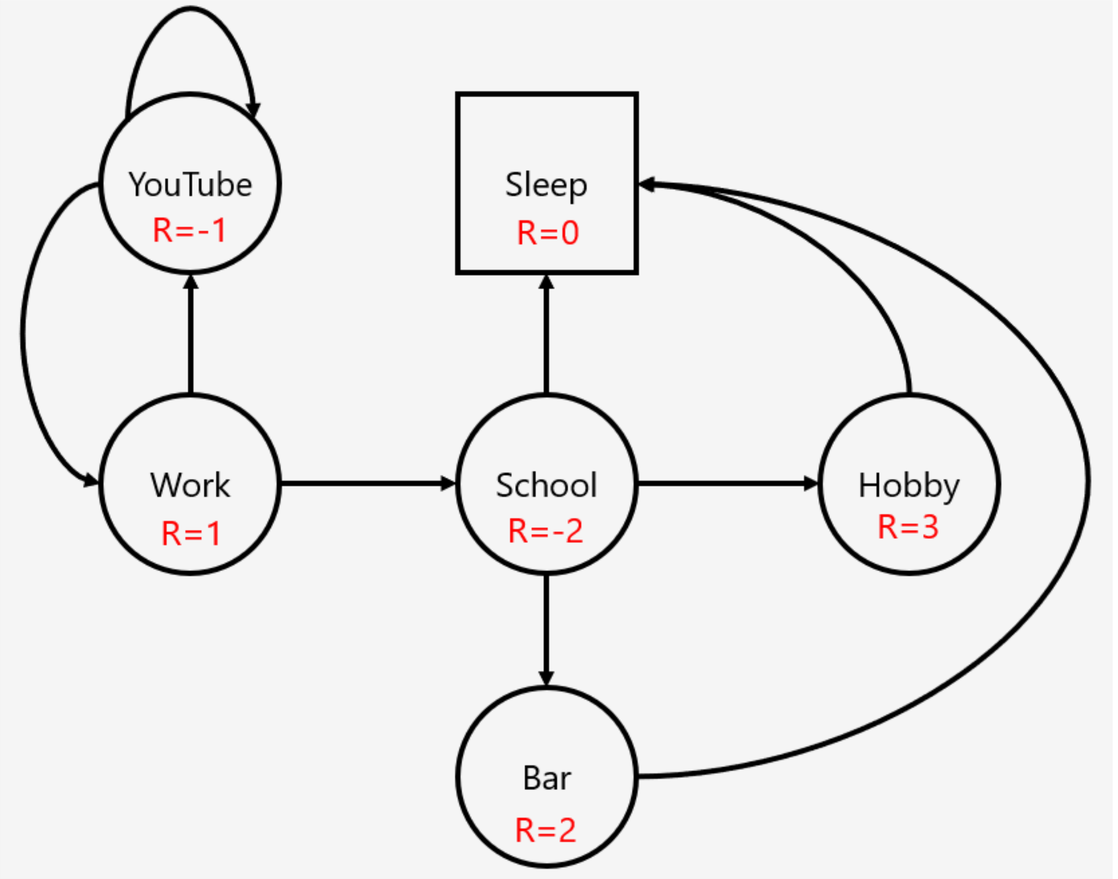
$$V^*(Sleep) = max_a[0 + 1(1.0 * 0)] = 0$$

$$V^*(School) = max_a[-2 + 1(1.0 * 2), -2 + 1(1.0 * 0), -2 + 1(1.0 * 3)] = 1$$

$$V^*(YouTube) = max_a[-1 + 1(1.0 * -1), -1 + 1(1.0 * 1)] = 0$$

$$V^*(Work) = max_a[1 + 1(1.0 * 0), 1 + 1(1.0 * 1)] = 2$$

$$\Pi^*(s) = argmax_a[R(s,a) + \gamma \sum_{s' \epsilon S} P^a_{ss'} V(s')]$$



A Markov Decision Process describing a college student's hypothetical situation.

Partially-observable Markov Decision Processes

The dynamical system is represented as follows:

- S: a finite set of state;
- A: a finite set of actions;
- O: a finite set of observations;
- $P : S \times A \times S \to [0, 1]$ is the transition probabilities;
- $Q : S \times A \to [0, 1]$ is the observation function.

The agent does not know the state, but gets noisy observations of the state.

Typically we intend to maximize state dependent reward: R(s) or R(s,a)

The policy must depend on the history: $\pi : (o_1, o_2, o_3, \ldots, o_k) \mapsto a$

POMDPs and Belief States

- **Belief** is a distribution over all states!

- Policy can be a mapping from a “belief state” to an action - instead of a mapping from a history of observations!

Let B denote the set of all probability distributions over S. (Note that this is a finite-dimensional continuum set, when S is a finite set.) A policy is a mapping from B into A. Let Π denote the set of all such policies. Then Π contains an optimal policy.