Markov Decision Processes

DAVIDE BACCIU – BACCIU@DI.UNIPI.IT

Introduction

Outline

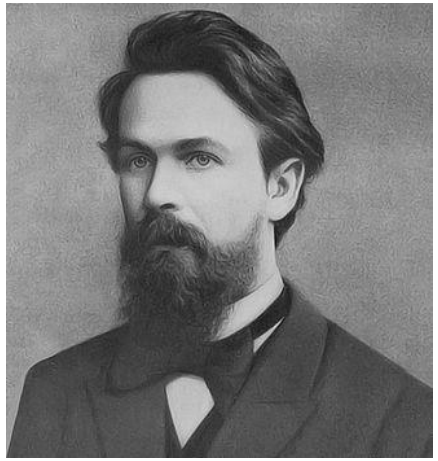
- ✓ Formalizing reinforcement learning with fully observable environment
  - ✓ Markov Processes
  - ✓ Markov Rewards
  - ✓ Markov Decision Processes
- ✓ A recursive formulation for value functions
- ✓ Extensions of the Markov decision process

Introduction to MDPs

- ✓ Markov decision processes formally describe an environment for reinforcement learning
  - ✓ Environment is fully observable
  - ✓ i.e. The current state completely characterises the process
- ✓ Almost all RL problems can be formalised as MDPs, e.g.
  - ✓ Optimal control primarily deals with continuous MDPs
  - ✓ Partially observable problems can be converted into MDPs
  - ✓ Bandits are MDPs with one state

Markov Process

Markov Property



“The future is independent of the past given the present"

Definition (Markov State)

A state S_t is Markov if and only if

$$P(S_{t+1}|S_1, \dots, S_t) = P(S_{t+1}|S_t)$$

- ✓The state captures all relevant information from the history
- ✓Once the state is known, the history may be thrown away
- ✓The state is a sufficient statistics for the future

State Transition Matrix

✓For a starting state s and successor state s', the state transition probability is defined by

$$P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$

✓The state transition matrix **P** defines the transition probabilities from all states s to all successor states s'

$$\boldsymbol{P} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}$$

where each row of the matrix sums to 1 (marginalization)

Markov Process

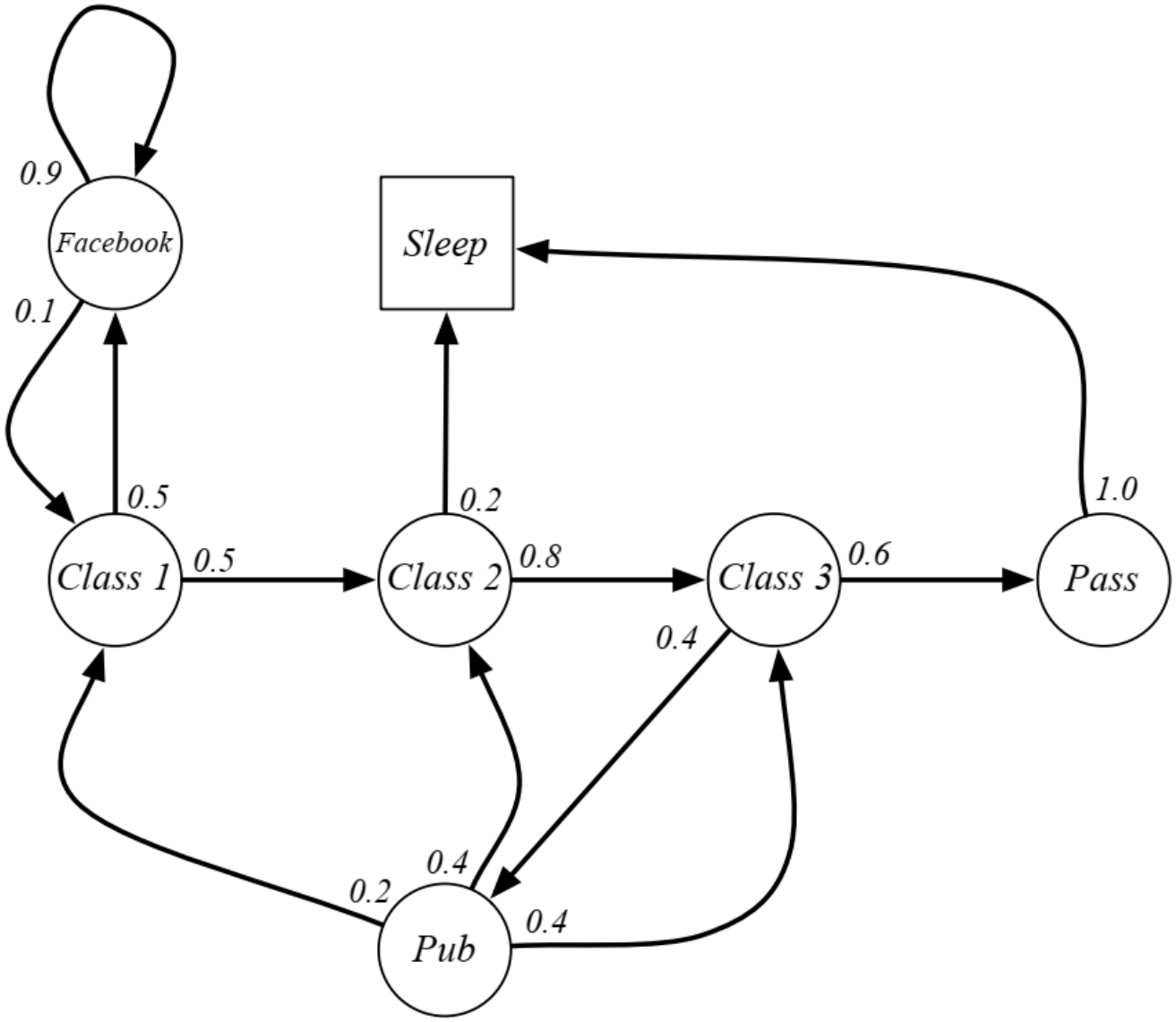
A Markov process is a memoryless random process, i.e. a sequence of random states $S_1, S_2, \ldots$ with the Markov property

Definition (Markov Process)

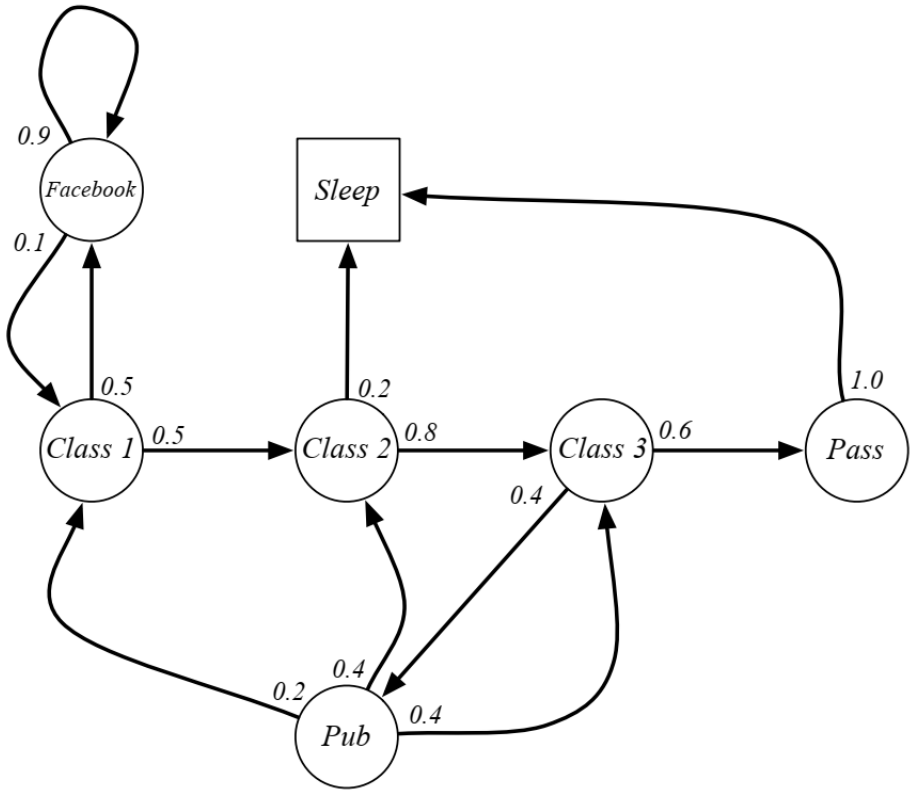
A Markov Process (or Markov Chain) is a tuple $\langle \mathcal{S}, \boldsymbol{P} \rangle$

- $\mathcal{S}$ is a finite set of states
- $\boldsymbol{P}$ is a state transition matrix, such that. $P_{ss'} = P(S_{t+1} = s' | S_t = s)$

Example – Student Markov Chain



Example – Student Markov Chain Episodes

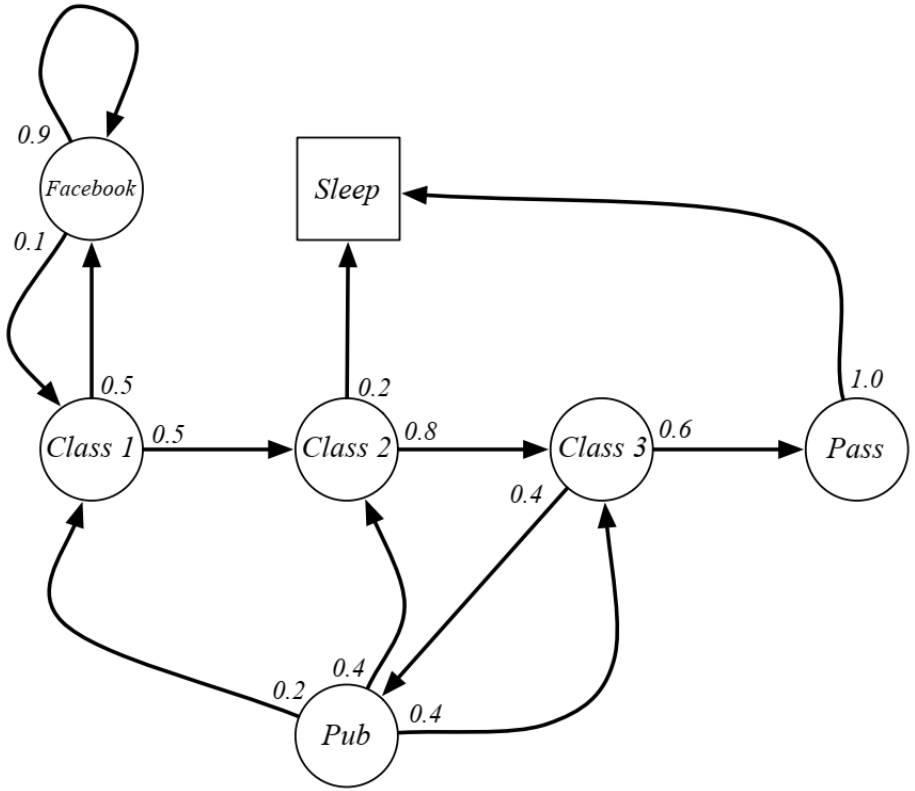


Sample episodes for Student Markov Chain starting from $S_1 = \text{"}Class\ 1\text{"}\ (C1)$

$S_1, S_2, \ldots, S_t$

- ✓ C1 C2 C3 Pass Sleep
- ✓ C1 FB FB C1 C2 Sleep
- ✓ C1 C2 C3 Pub C2 C3 Pass Sleep
- ✓ C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

Example – Student Markov Chain Transition Matrix



Future

	C1	C2	C3	Pass	Pub	FB	Sleep
C1		0.5				0.5	
C2			0.8				0.2
C3				0.6	0.4		
Pass							1.0
Pub	0.2	0.4	0.4				
FB	0.1					0.9	
Sleep							1

Current

$$P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$

Markov Rewards

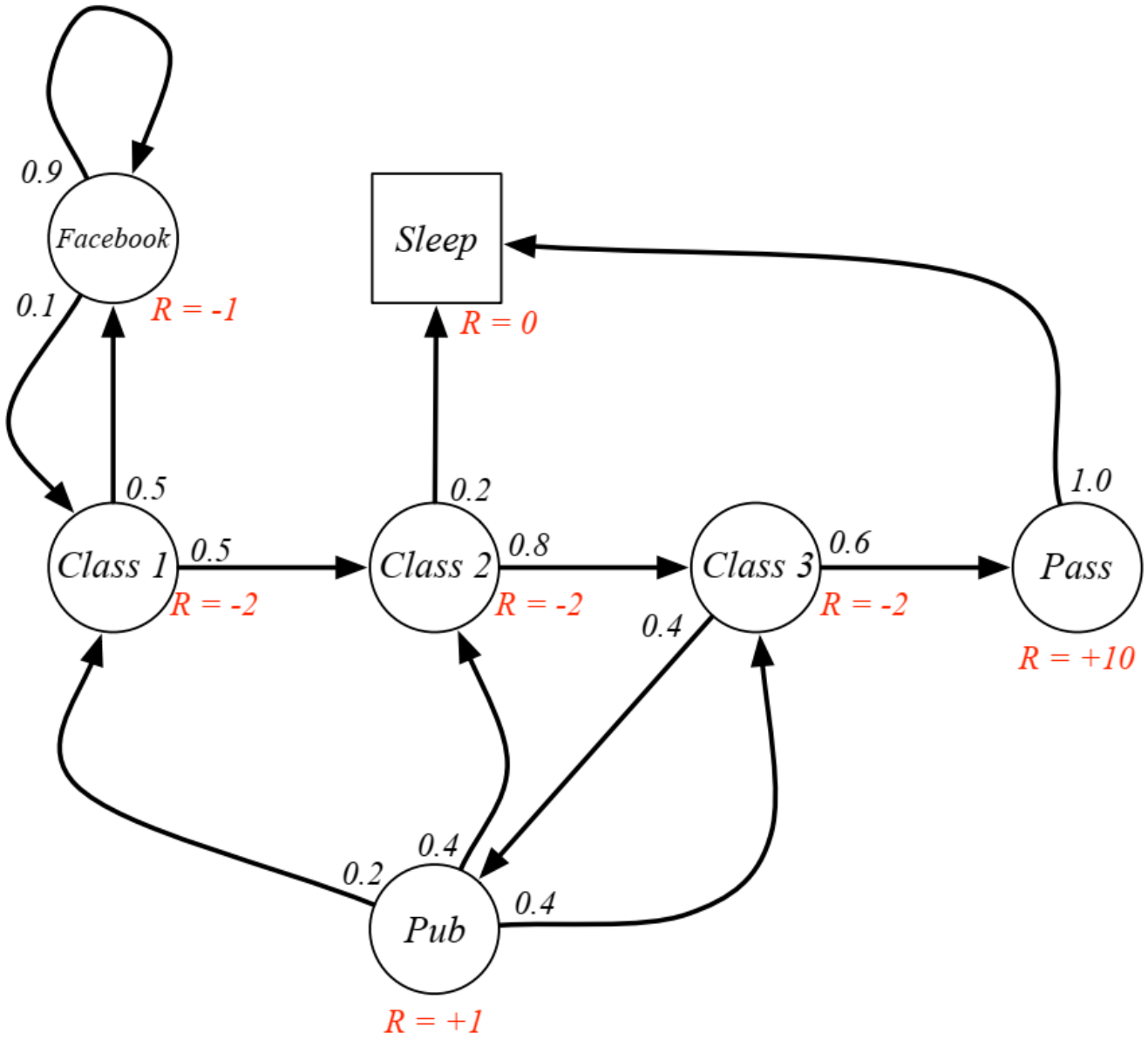
Markov Reward Process

A Markov Reward Process (MRP) is a Markov chain with reward values

Definition (Markov Reward Process)

A Markov Reward Process is a tuple $\langle \mathcal{S}, \boldsymbol{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\boldsymbol{P}$ is a state transition matrix, s.t. $P_{ss'} = P(S_{t+1} = s' | S_t = s)$
- $\mathcal{R}$ is a reward function, s.t. $\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$
- γ is a discount factor, $\gamma \in [0,1]$



Example – Student MRP

Return

Definition (Return)

The return G_t is the total discounted reward from time-step t

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- ✓ The value of receiving reward R after $k+1$ timesteps is $\gamma^k R$
- ✓ γ values immediate reward Vs delayed reward
  - ✓ $\gamma \approx 0$ leads to "myopic" evaluation
  - ✓ $\gamma \approx 1$ leads to "far-sighted" evaluation

On the discount term

- ✓Mathematically convenient to discount rewards
- ✓Avoids infinite returns in cyclic Markov processes
- ✓Uncertainty about the future may not be fully represented
- ✓Application dependent
  - ✓If the reward is financial, immediate rewards may earn more interest than delayed rewards
  - ✓Biological plausibility (animal behaviour shows preference for immediate reward)
- ✓It is sometimes possible to use undiscounted Markov reward processes (i.e. $\gamma = 1$), e.g. if all sequences terminate

Value Function

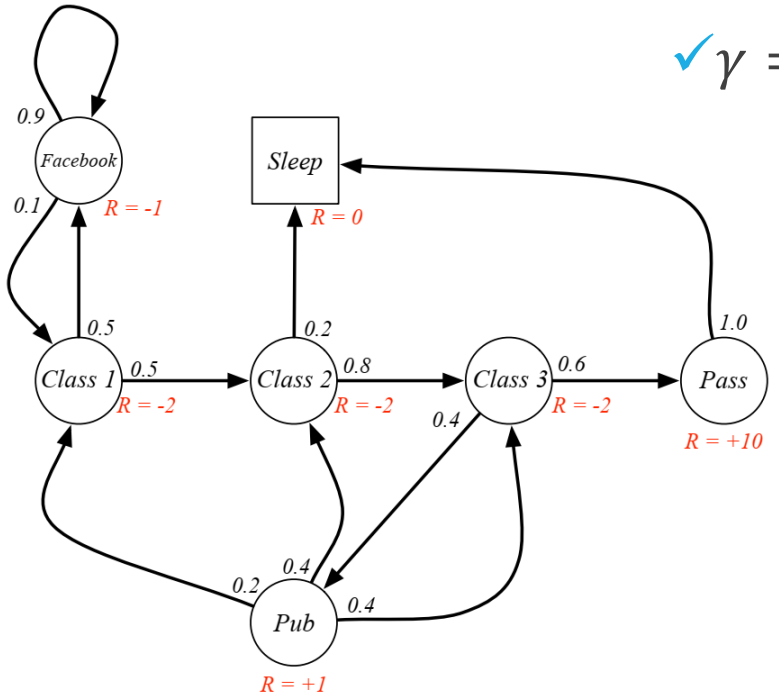
Measures the long-term value of being in a certain state s

Definition (Value Function)

The state-value function $v(s)$ of a Markov Reward Process is the expected return starting from state s

$$v(s) = \mathbb{E}[G_t | S_t = s]$$

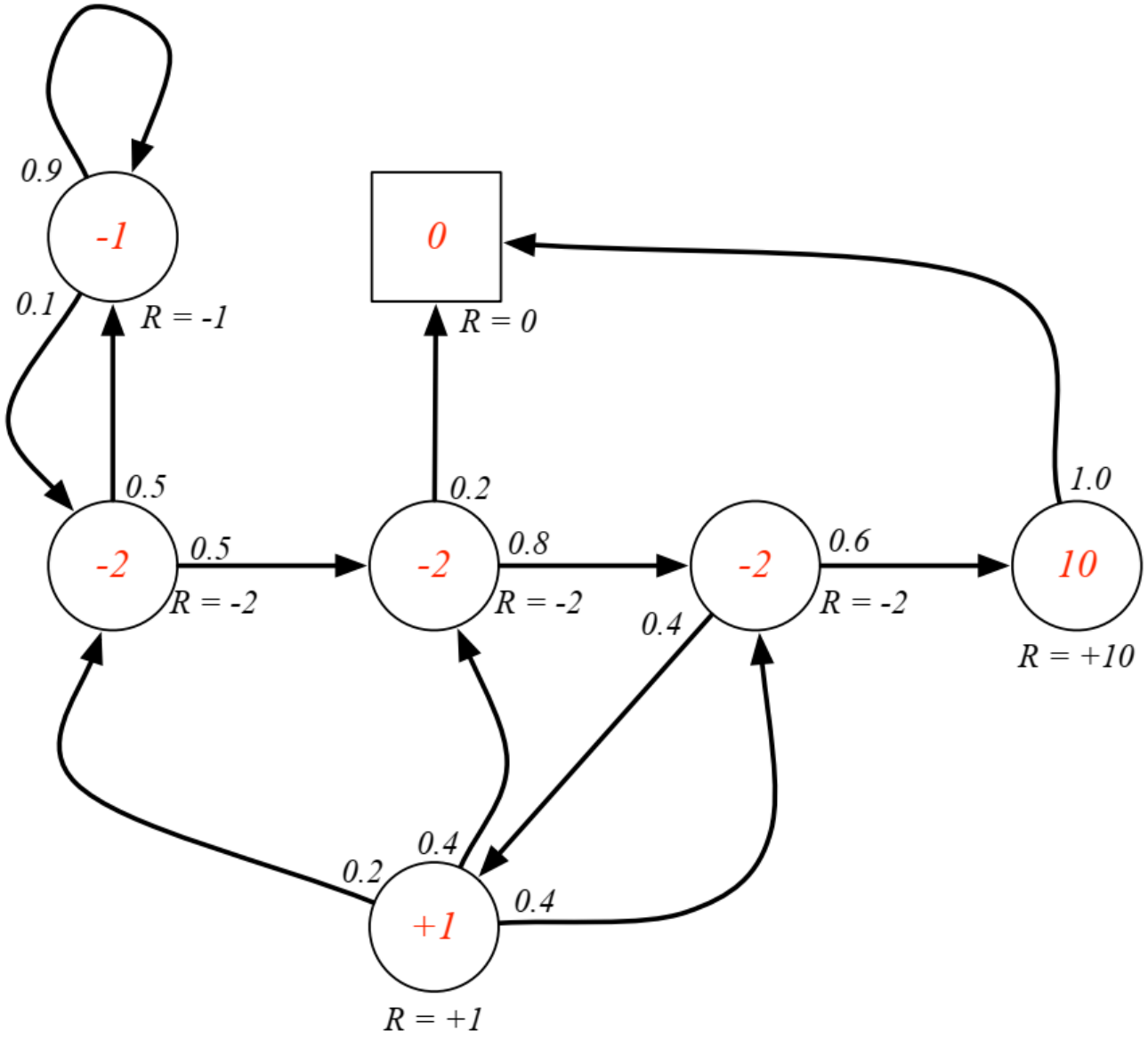
Example – Student MRP Returns



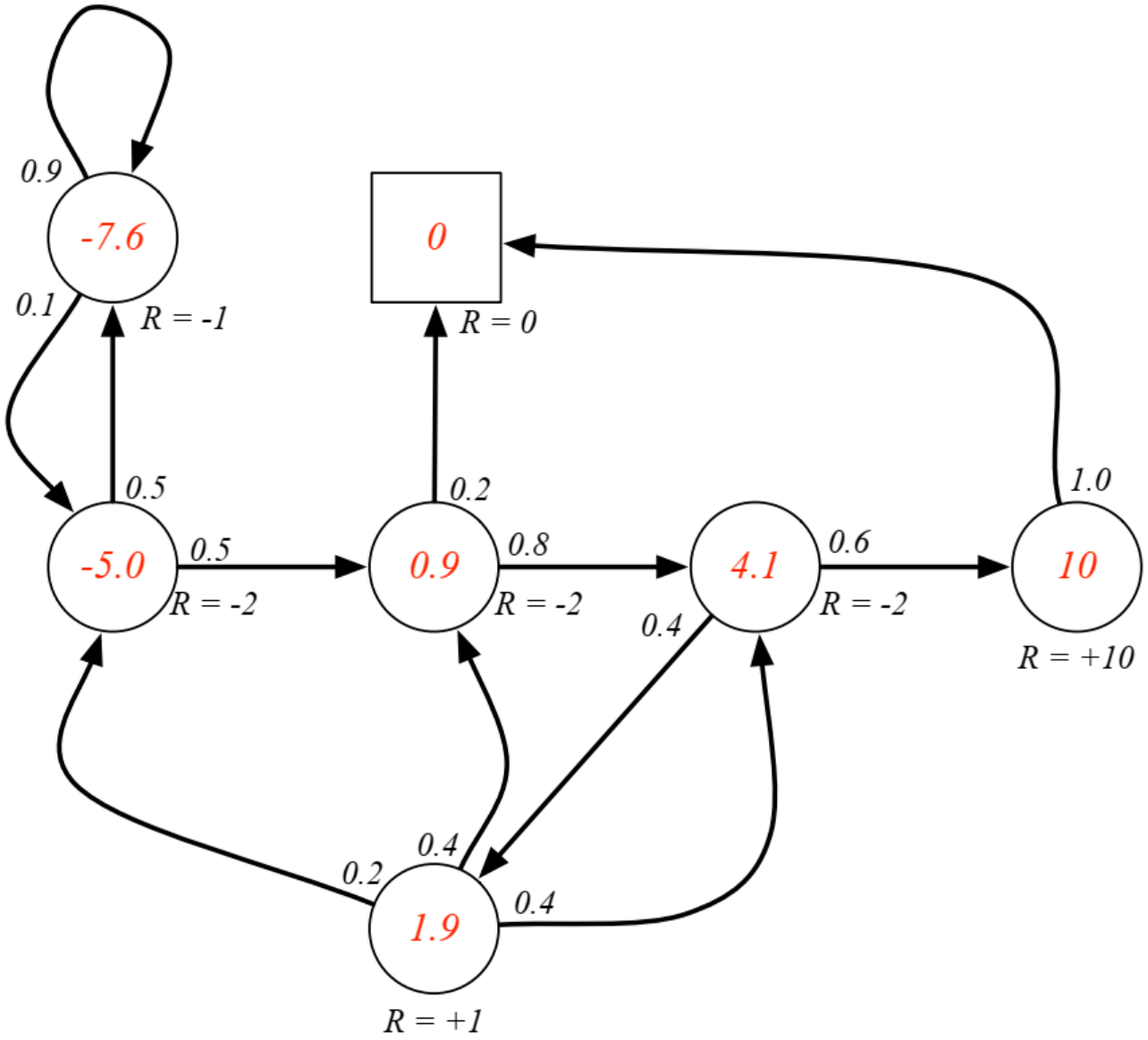
Sample returns for student Markov Reward Process

- ✓ Starting from $S_1 = C1$
- ✓ $\gamma = \frac{1}{2}$

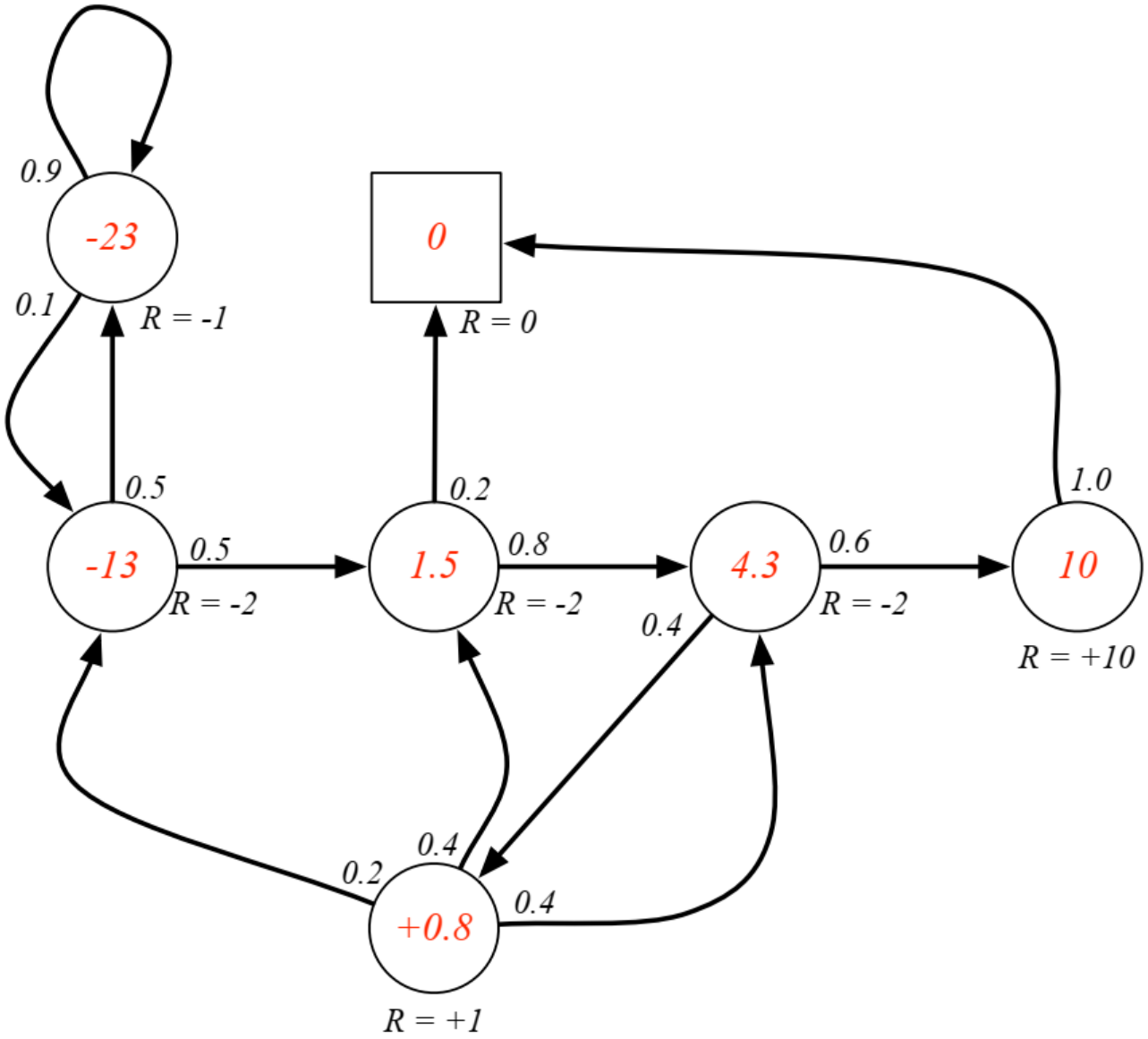
- ✓ C1 C2 C3 Pass Sleep
  - ✓ $v(C1) = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8} = -2.25$
- ✓ C1 FB FB C1 C2 Sleep
  - ✓ ?
- ✓ …



Example – Student MRP State-Value Function ($\gamma = 0$)



Example – Student MRP State-Value Function ($\gamma = 0.9$)



Example – Student MRP State-Value Function ($\gamma = 1$)

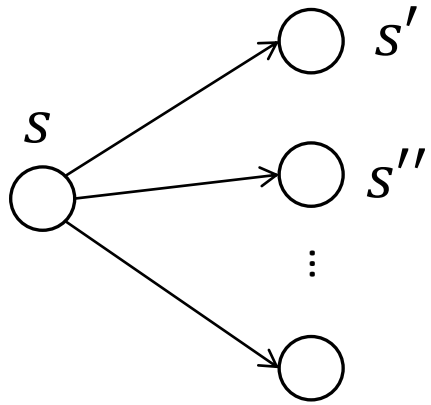
Bellman Equation for MRPs

- ✓ The value function $v(S_t)$ can be decomposed into two parts
  - ✓ Immediate reward R_{t+1}
  - ✓ Discounted value of successor state $\gamma v(S_{t+1})$

$$
\begin{aligned}
v(s) &= \mathbb{E}[G_t | S_t = s] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|S_t\right] \\
&= \mathbb{E}[R_{t+1} + \textstyle\sum_{k=1}^{\infty} \gamma^k R_{t+k+1} \,|S_t] \\
&= \mathbb{E}\left[R_{t+1} + \gamma\left(R_{t+2} + \textstyle\sum_{k=2}^{\infty} \gamma^k R_{t+k+1}\right)|S_t\right] \\
&= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t] \\
&= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t]
\end{aligned}
$$

Bellman Equation for MRPs – Which future state?

$$v(s) = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t = s]$$

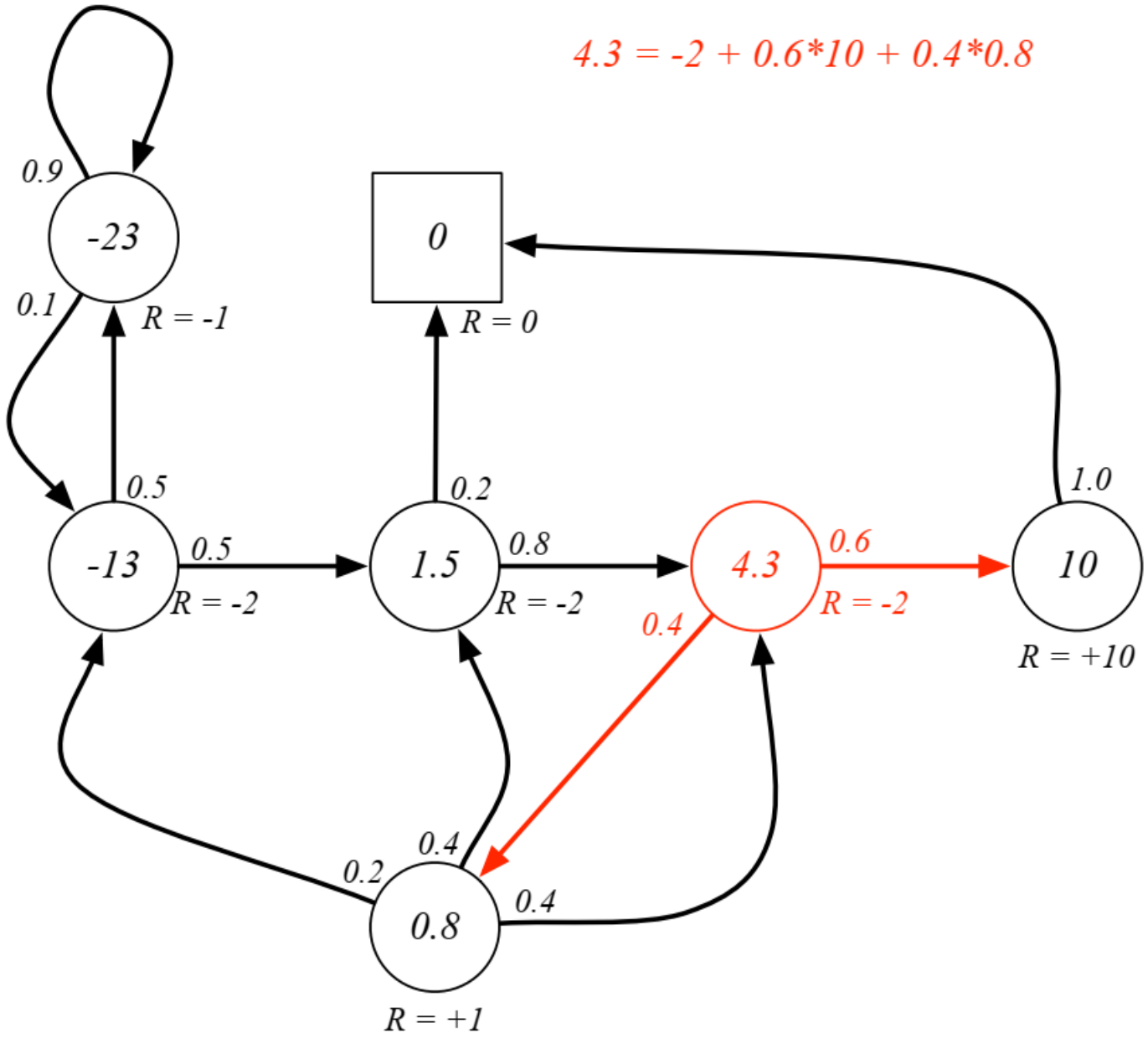


Reward function $\mathcal{R}_s$

$$v(s) = \mathbb{E}[R_{t+1} | S_t = s] + \gamma \mathbb{E}[v(S_{t+1}) | S_t = s]$$

→ The expected state-value of being in any state reachable from s

$$v(s) = \mathcal{R}_s + \gamma \sum_{s'} P_{ss'} v(s')$$



Example – Bellman Equation for Student MRP (γ=1)

Bellman Equation – Matrix Form

Considering n available states

$$v = \mathcal{R} + \gamma \boldsymbol{P} v$$

$$v = \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} \qquad \mathcal{R} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} \qquad \boldsymbol{P} = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}$$

Provides us with a nice linear system

Solving the linear Bellman Equation

$$v = \mathcal{R} + \gamma \boldsymbol{P} v = (\boldsymbol{I} - \gamma \boldsymbol{P})^{-1} \mathcal{R}$$

- ✓ Computational complexity is $\mathrm{O}(\mathrm{n}^3)$
  - ✓ Direct solution only feasible for small MRPs
- ✓ Iterative methods for large MRPs
  - ✓ Dynamic programming
  - ✓ Monte-Carlo evaluation
  - ✓ Temporal-Difference learning

Markov Decision Processes

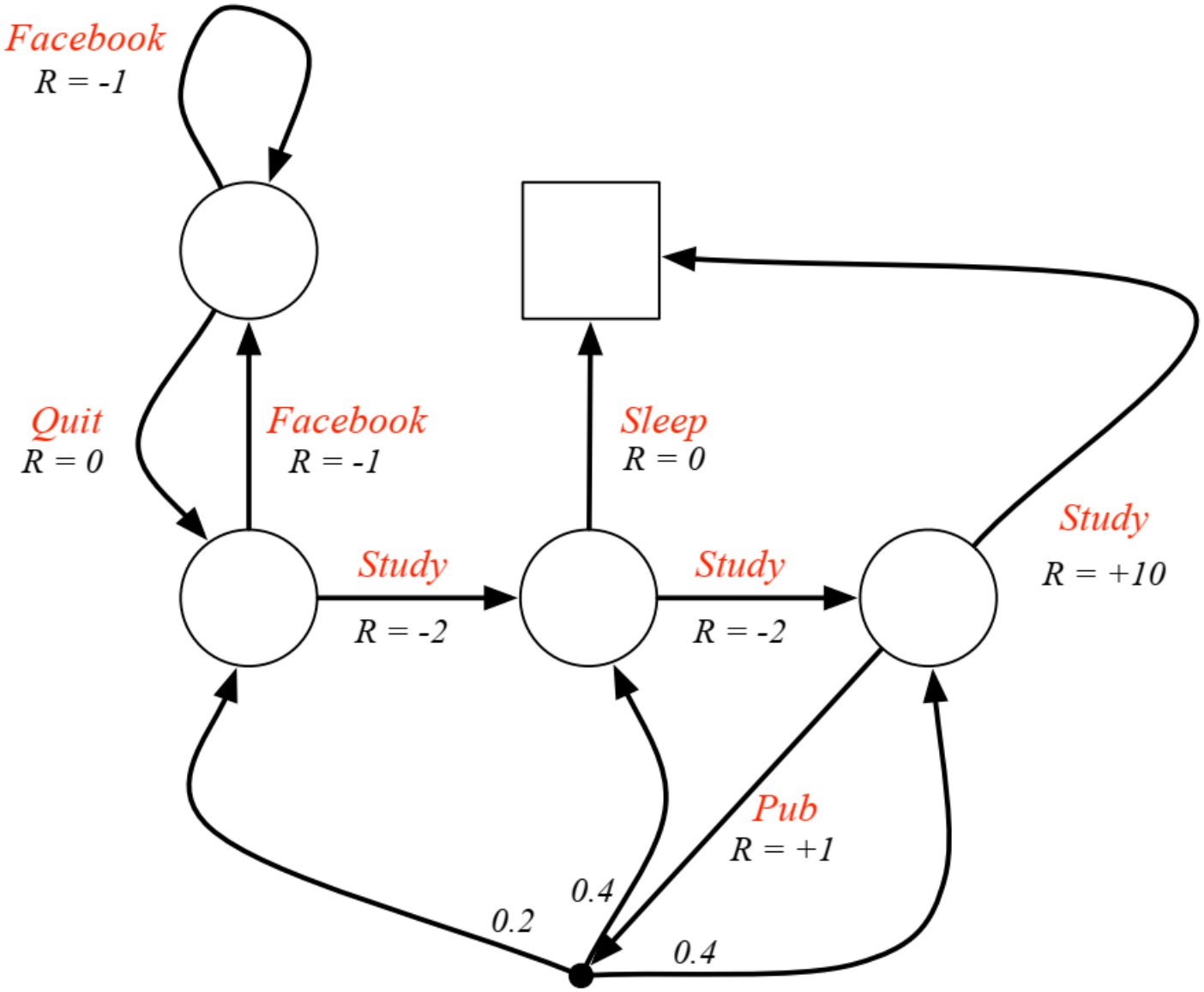
Markov Decision Process

A Markov Decision Process (MDP) is a Markov reward process with actions. It is an environment in which all states are Markov

Definition (Markov Decision Process)

A Markov Reward Process is a tuple $\langle \mathcal{S}, \mathcal{A}, \boldsymbol{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions a
- $\boldsymbol{P}$ is a state transition matrix, s.t. $P_{ss'}^{a} = P(S_{t+1} = s' | S_t = s, A_t = a)$
- $\mathcal{R}$ is a reward function, s.t. $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
- γ is a discount factor, $\gamma \in [0,1]$



Example – Student Markov Decision Process

Policy - Definition

Definition (Policy)

A policy π is a distribution over actions a given states s

$$\pi(a|s) = P(A_t = a|S_t = s)$$

- ✓Define the behavior of an agent
- ✓MDP policies depend only on the current state (Markovian)
- ✓Policies are stationary (time-independent): $\mathrm{A_t} \sim \pi(\cdot\,|s), \forall t > 0$

Under Policy

Given an MDP $\langle \mathcal{S}, \mathcal{A}, \boldsymbol{P}, \mathcal{R}, \gamma \rangle$ and a policy π

✓The state sequence $S_1, S_2, \ldots$ is a Markov process $\langle \mathcal{S}, \boldsymbol{P}^{\pi} \rangle$ (under policy)

✓The state and reward sequence $S_1, R_2, S_2 \ldots$ is a Markov reward process $\langle \mathcal{S}, \boldsymbol{P}^{\pi}, \mathcal{R}^{\boldsymbol{\pi}} \rangle$ (under policy), such that

$$P_{ss'}^{\pi} = \sum_{a \in \mathcal{A}} \pi(a|s) P_{ss'}^{a}$$

$$\mathcal{R}_{s}^{\pi} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}_{s}^{a}$$

Value Function (with policy)

Definition (Value Function)

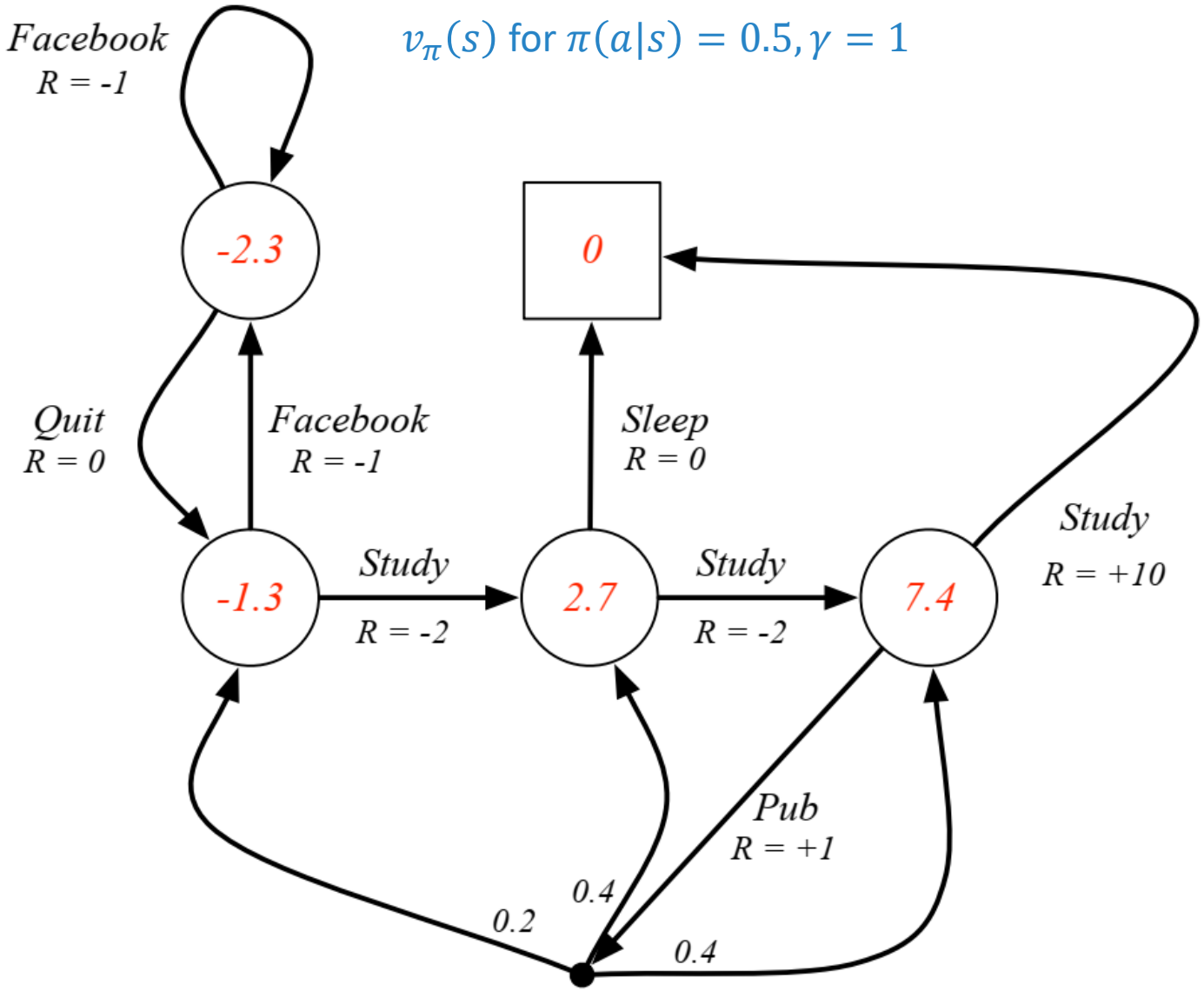
The state-value function $v_\pi(s)$ of an MDP is the expected return starting from state s and following policy π

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$$

Definition (Action-Value Function)

The action-value function $q_\pi(s,a)$ is the expected return starting from state s, taking action a, and then following policy π

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$



Example – Student State-Value Function

Bellman Expectation Equation – Value and Action-Value Functions

The state-value function can again be decomposed into immediate reward plus discounted value of successor state

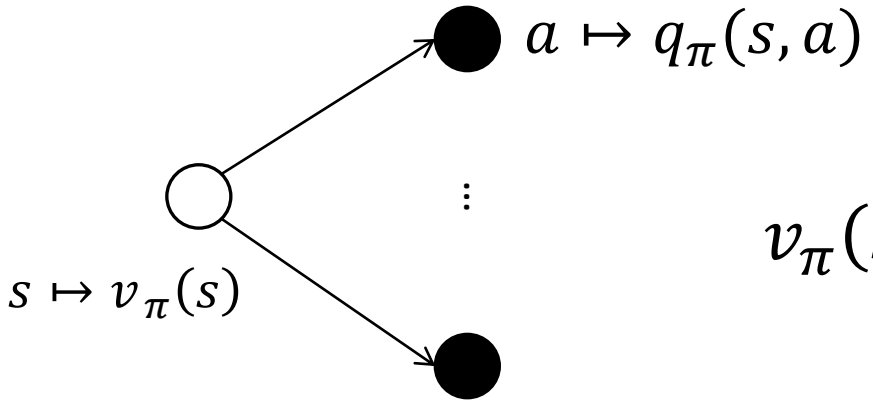
$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

Similarly, we can decompose the action-value function

$$q_\pi(s, a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

Both come from the recursive nature of return G_t

Bellman Expectation for v_π (I)

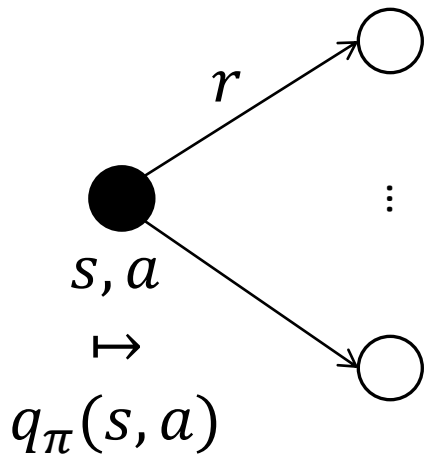


$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

Expectation with respect to the actions that can be taken starting from s

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

Bellman Expectation for q_π (I)

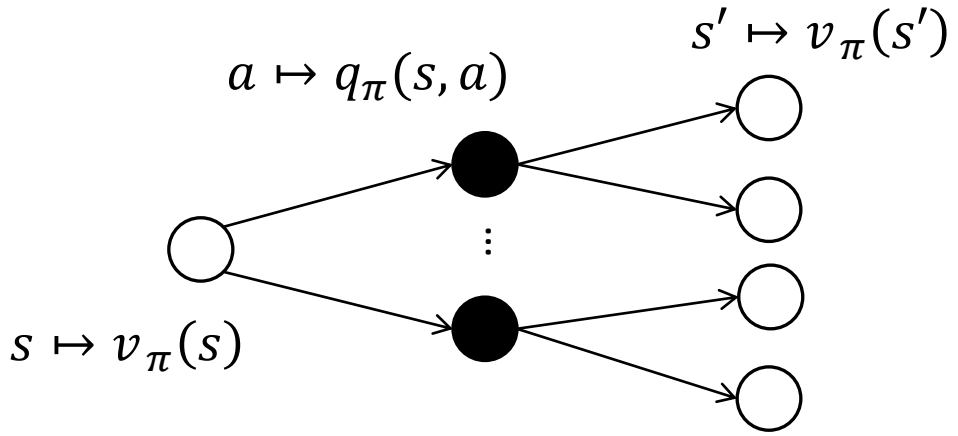


$$q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

Expectation with respect to the states reachable from s having taken action a

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s')$$

Bellman Expectation for v_π (II)

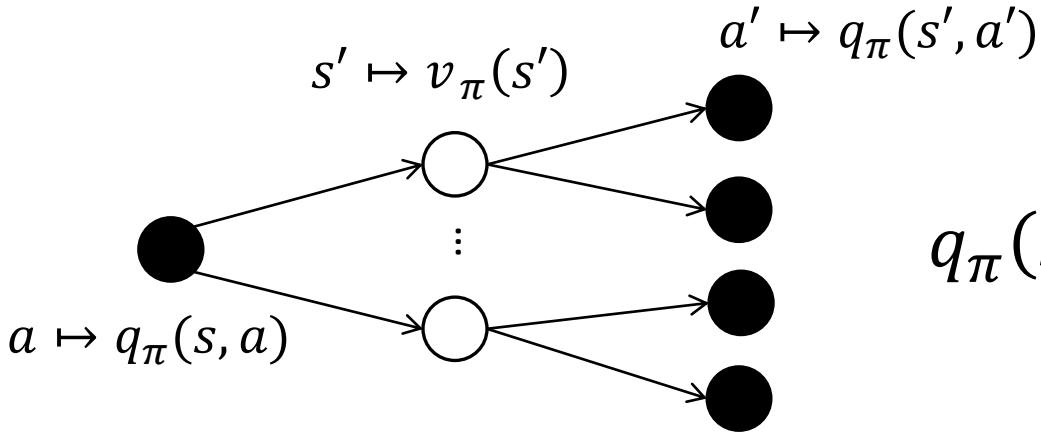


$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

The expected return of being in a state reachable from s through action a and then continue following policy

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s') \right)$$

Bellman Expectation for q_π (II)



$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s')$$

The expected return of any action a' taken from states reachable from s through action a (and then follow policy)

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a')$$

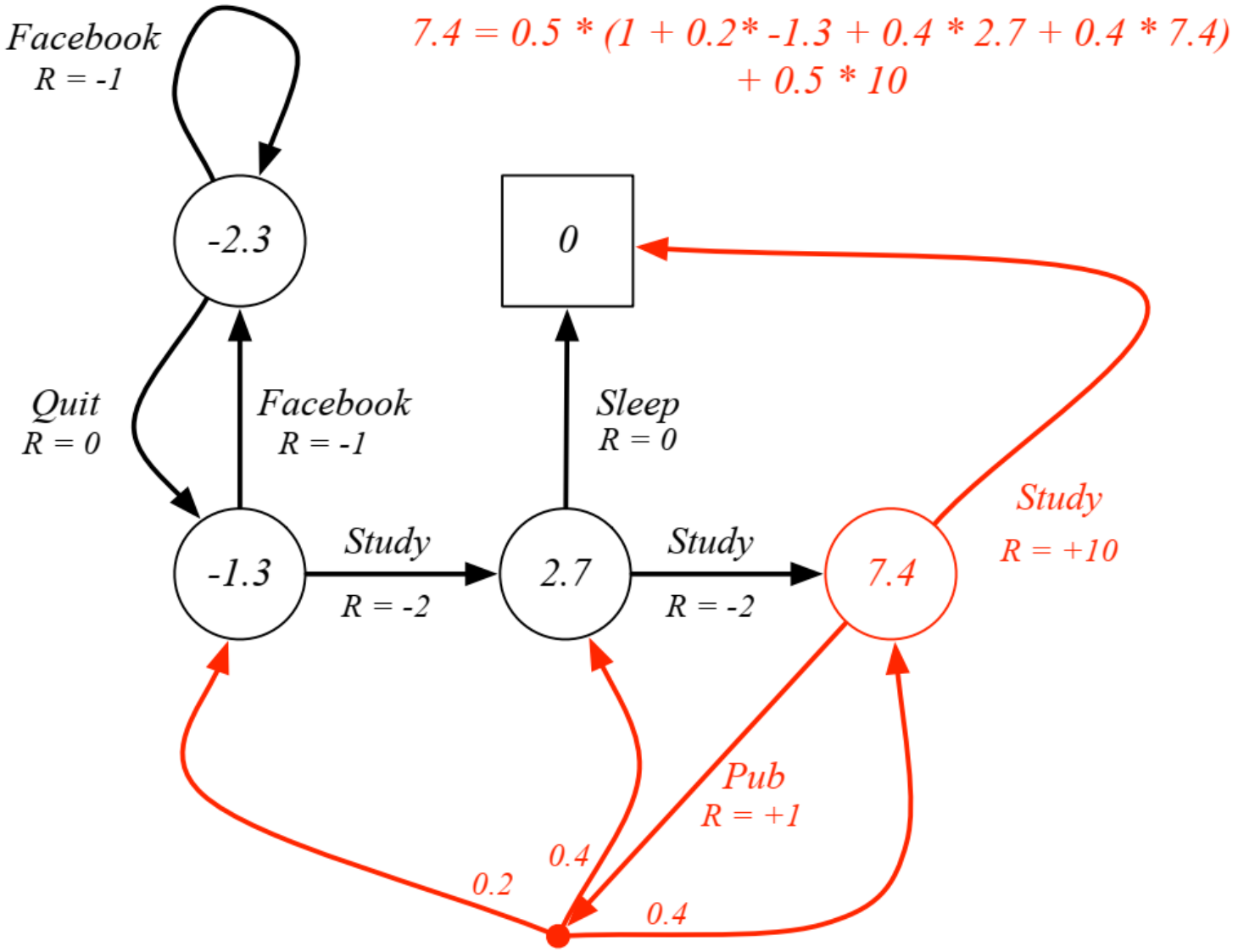
Bellman Expectation Equation – Matrix Form

Again a linear system

$$v_\pi = \mathcal{R}^\pi + \gamma \boldsymbol{P}^{\boldsymbol{\pi}} v_\pi$$

With direct solution

$$v_\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}^{\boldsymbol{\pi}})^{-1} \mathcal{R}^\pi$$



Example – Bellman Expectation in Student MDP

Optimal Value Function

Definition (Optimal State/Action Functions)

The optimal state-value function $v_*(s)$ is the maximum value function over all policies

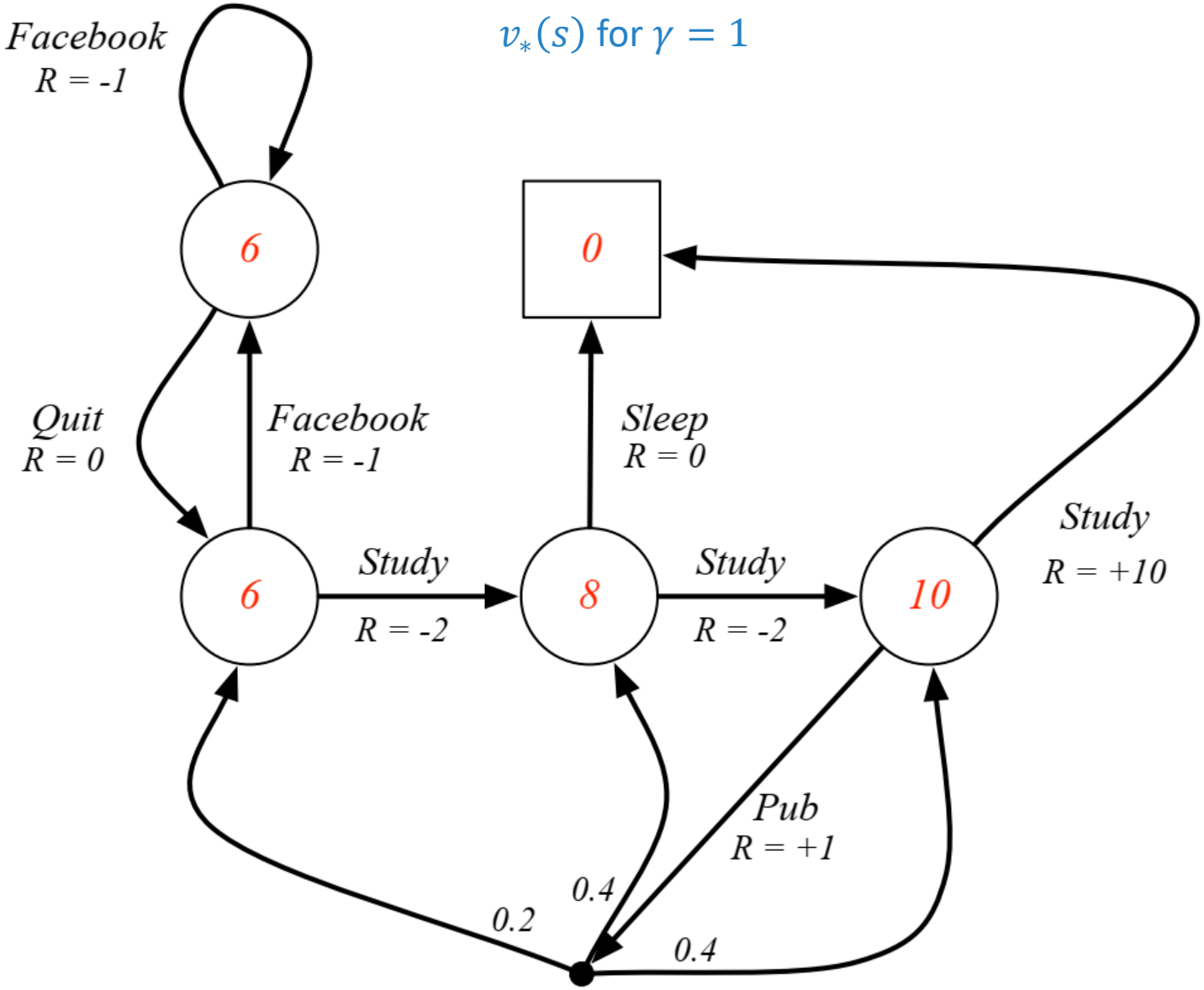
$$v_*(s) = \max_\pi v_\pi(s)$$

The optimal action-value function $q_*(s,a)$ is the maximum action-value function over all policies

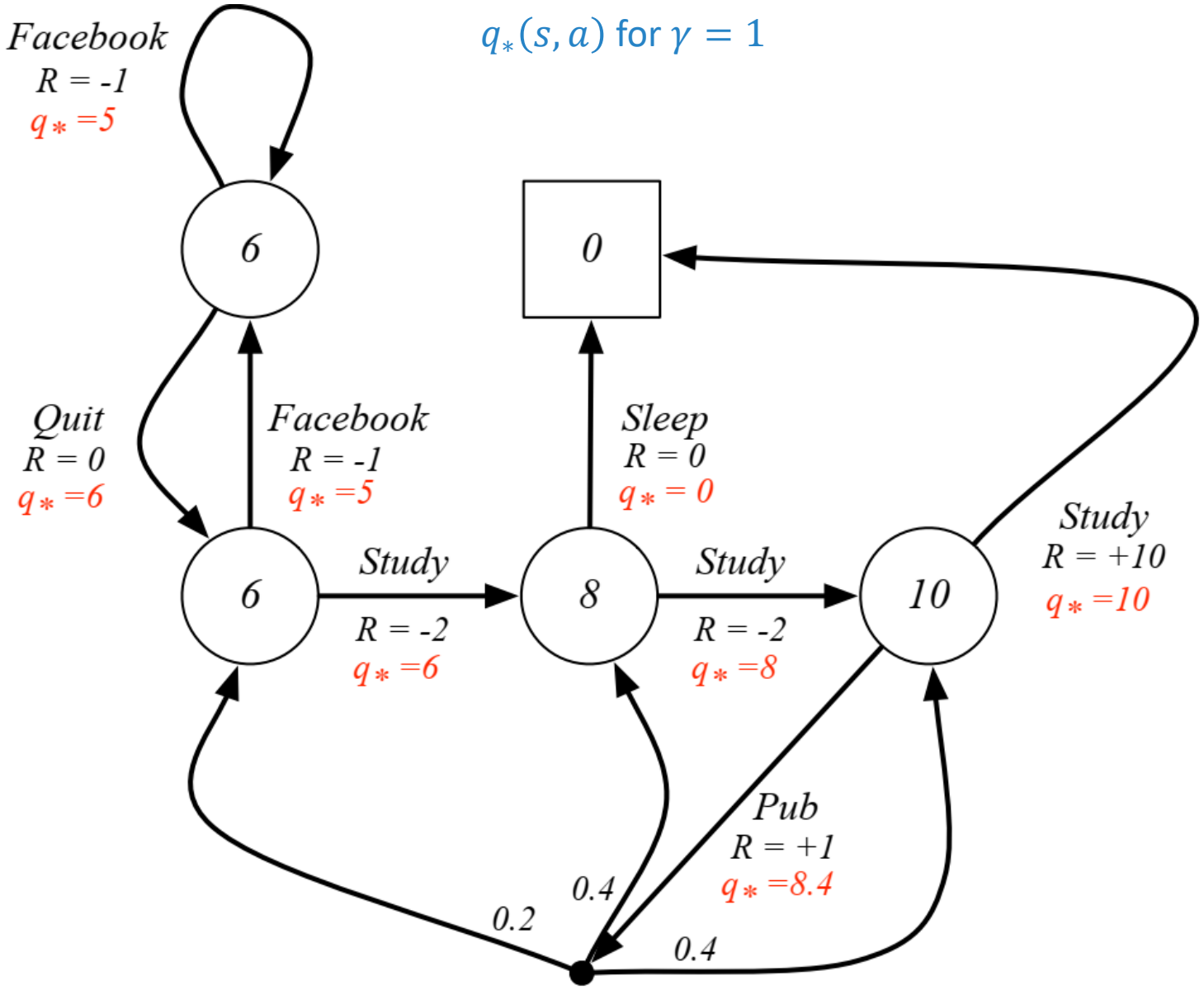
$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

✓The optimal value function determines the best possible performance in the MDP

✓An MDP is solved when we know the optimal value function



Example – Optimal Value Function for Student MDP



Example – Optimal Action-Value Function for Student MDP

Optimal Policy

Define a partial ordering over policies

$$\pi \geq \pi' if\ v_\pi(s) \geq v_{\pi'}(s), \forall s$$

Theorem

For any Markov Decision Process

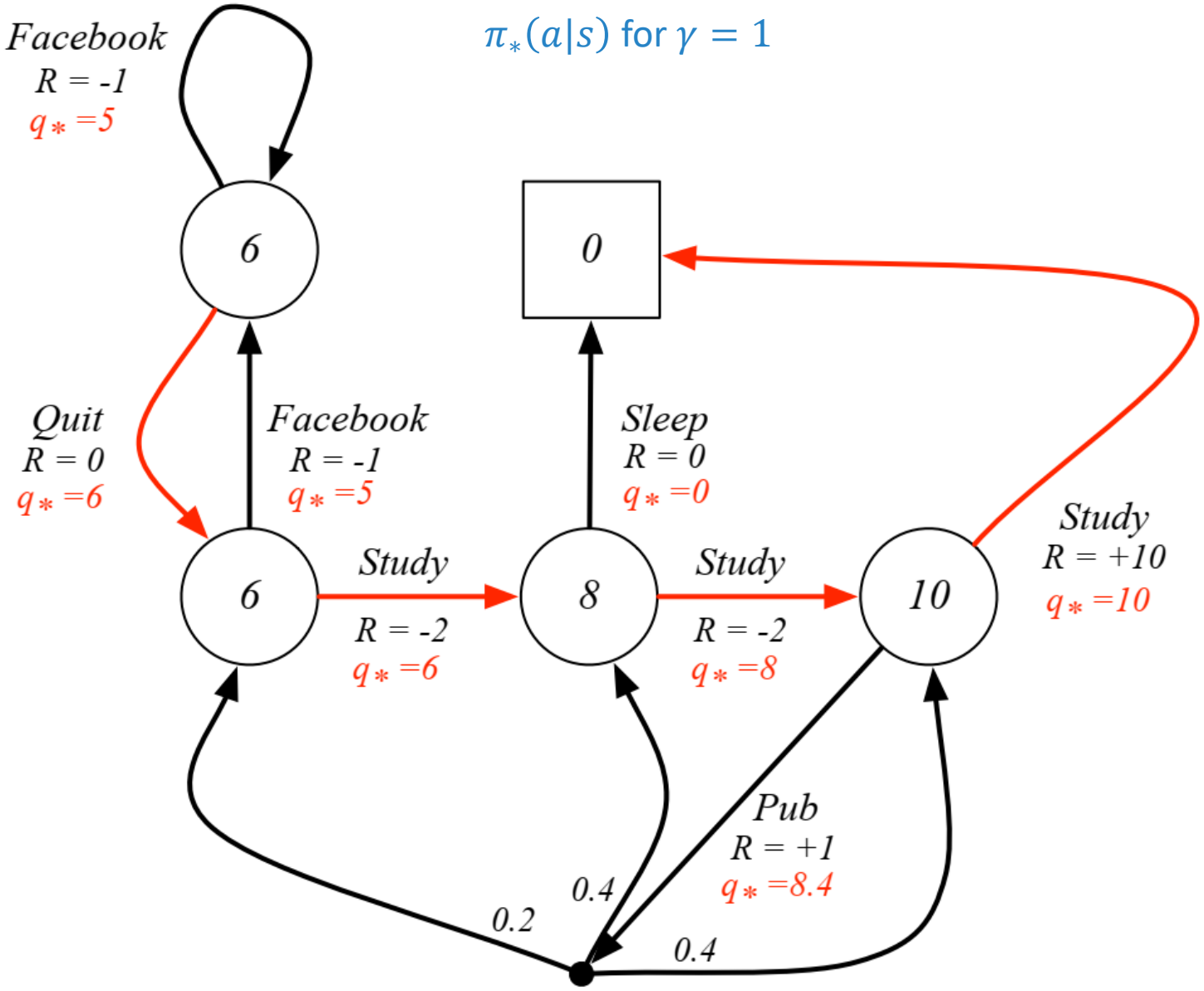
- ✓ There exist an optimal policy π_* that is better than or equal than all other: $\pi_* \geq \pi, \forall \pi$
- ✓ All optimal polices achieve the optimal value function: $v_{\pi_*}(s) = v_*(s)$
- ✓ All optimal polices achieve the optimal action-value function: $q_{\pi_*}(s,a) = q_*(s,a)$

Finding an Optimal Policy

An optimal policy can be found by maximising over $q_*(s,a)$

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in \mathcal{A}} q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

- ✓There is always a deterministic optimal policy for any MDP
- ✓If we know $q_*(s,a)$, we straightforwardly find the optimal policy



Example – Optimal Policy for Student MDP

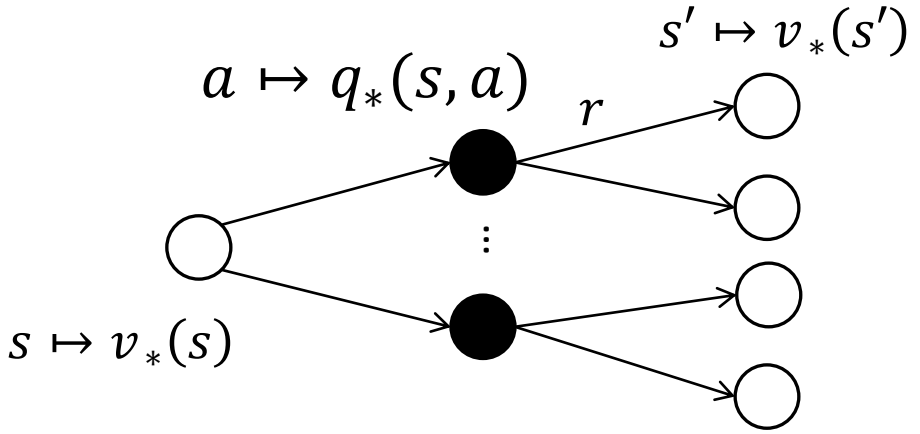
Bellman Optimality Equations

Optimal value functions are recursively related Bellman-style

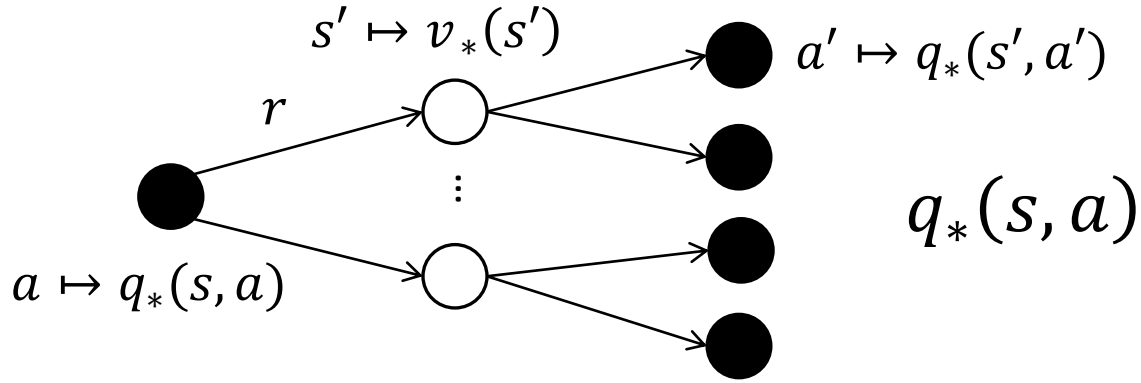
$$v_*(s) = \max_{a \in \mathcal{A}} q_*(s,a)$$

$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_*(s')$$

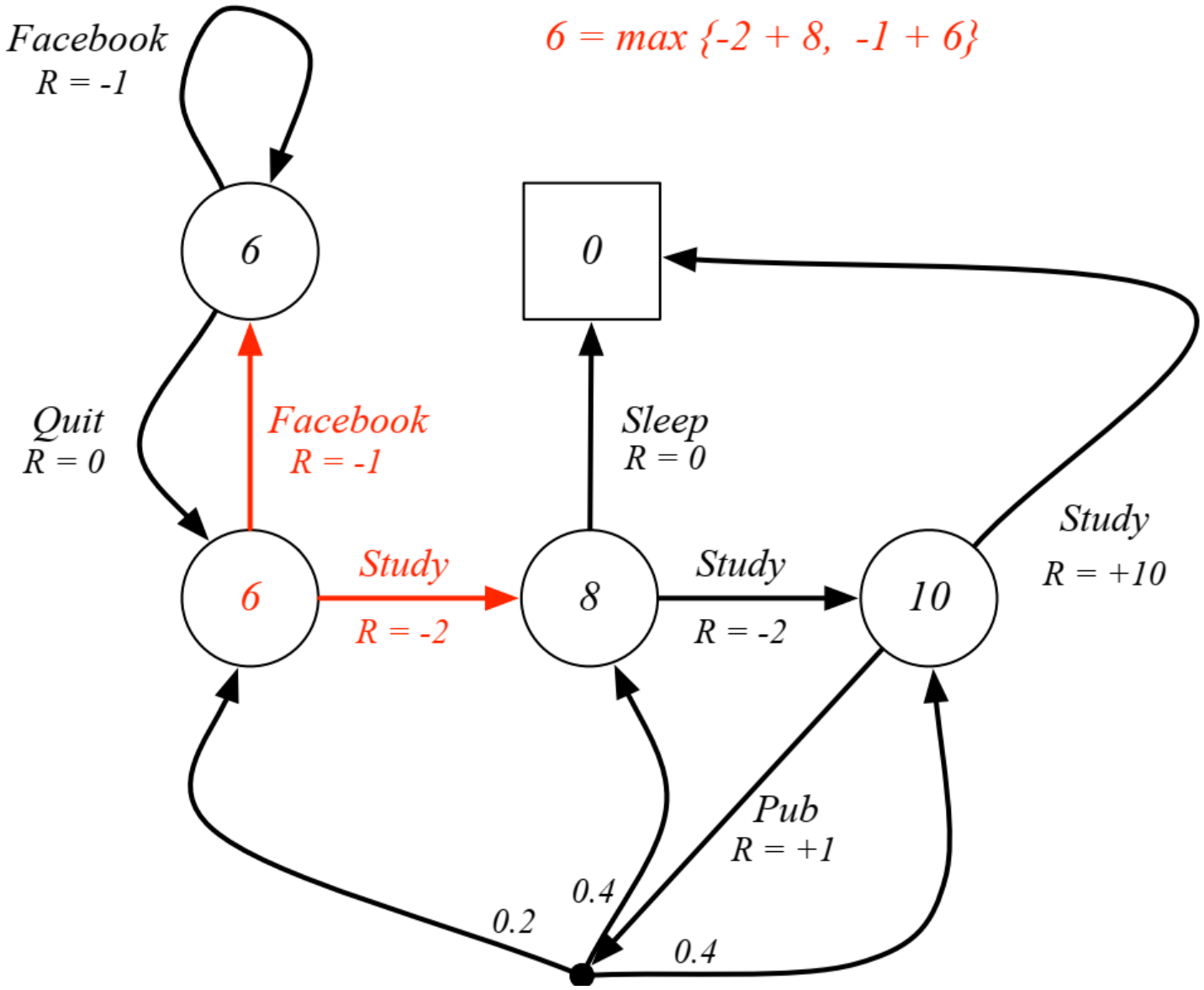
Bellman Optimality Equations - v_*, q_*



$$v_*(s) = \max_{a \in \mathcal{A}} \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_*(s')$$



$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \max_{a' \in \mathcal{A}} q_*(s',a')$$



Example – Bellman Optimality Equation

Solving the Bellman Optimality Equation

- ✓ Bellman Optimality Equation is non-linear
- ✓ No closed form solution (in general)
- ✓ Many iterative solution methods
  - ✓ Value Iteration
  - ✓ Policy Iteration
  - ✓ Q-learning
  - ✓ SARSA

MDP Extensions

Infinite MDPs

- ✓ Countably infinite state and/or action spaces
- ✓ Continuous state and/or action spaces
  - ✓ Closed form for linear quadratic model (LQR)
- ✓ Continuous time
  - ✓ Requires partial differential equations
  - ✓ Hamilton-Jacobi-Bellman (HJB) equation
  - ✓ Limiting case of Bellman equation as time-step $\rightarrow 0$

Partially Observable MDP (POMDP)

✓A Partially Observable Markov Decision Process is an MDP with hidden states

✓A Hidden Markov Model with actions

Definition (POMDP)

A POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \boldsymbol{P}, \mathcal{R}, \mathcal{Z}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions a
- $\mathcal{O}$ is a finite set of observations
- $\boldsymbol{P}$ is a state transition matrix, s.t. $P_{ss'}^{a} = P(S_{t+1} = s' | S_t = s, A_t = a)$
- $\mathcal{R}$ is a reward function, s.t. $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
- $\mathcal{Z}$ is an observation function
- γ is a discount factor, $\gamma \in [0,1]$

Belief States

Definition (History)

A history H_t is a sequence of actions, observations and rewards

$$H_t = A_0 O_1 R_1, \dots, A_{t-1} O_t R_t$$

Definition (Belief State)

A belief state $b(h)$ is a distribution over states conditioned on the history h

$$b(h) = [P(S_t = s_1 | H_t = h), \dots, P(S_t = s_n | H_t = h)]$$

Wrap-up

Take (stay) home messages

- ✓Markov decision processes are a formalism to describe a fully-observable environment for reinforcement learning
  - ✓A state-transition system enriched with actions and reward
  - ✓Leverage Markov assumption to separate future from the past
- ✓A recursive formulation for value functions
  - ✓ Using Bellman equations
- ✓ Any MDP allows for an optimal policy
  - ✓Maximisation process on the state-value function
  - ✓Recursive and nonlinear (no closed form)
- ✓MPD can be relaxed to infinite and continuous actions/state and partially observable environments (through belief instead of deterministic states)

Next Lecture

Planning by Dynamic Programming

- ✓ A.K.A. solving a known MDP
- ✓ Dynamic programming
  - ✓ A method for solving complex problems by breaking them down into subproblems
- ✓ Policy Evaluation & Iteration
- ✓ Value Evaluation