AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

Markov decision Process

Markov Processes — tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

Markov Reward Processes (MRP) — tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

Markov Decision Processes (MDP) — tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

Extensions to MDPs

Courtesy: David Silver Slides MDP Lec2

Markov Decision Process

- *Markov decision processes* formally describe an environment for reinforcement learning
- Where the 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

Courtesy: David Silver Slides MDP Lec2

Markov Property

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

Definition

A state S_t is *Markov* if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \ldots, S_t]$$

- The state captures all relevant information from the history
- Once the state is known, the history may be thrown away
- i.e. The state is a sufficient statistic of the future

Courtesy: David Silver Slides MDP Lec2

State Transition Matrix

For a Markov state s and successor state s', the *state transition probability* is defined by

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

State transition matrix $\mathcal{P}$ defines transition probabilities from all states s to all successor states s',

$$\mathcal{P} = \text{from} \begin{matrix} \\ \\ \\ \end{matrix} \overset{\text{to}}{\begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix}}$$

where each row of the matrix sums to 1.

Courtesy: David Silver Slides MDP Lec2

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.

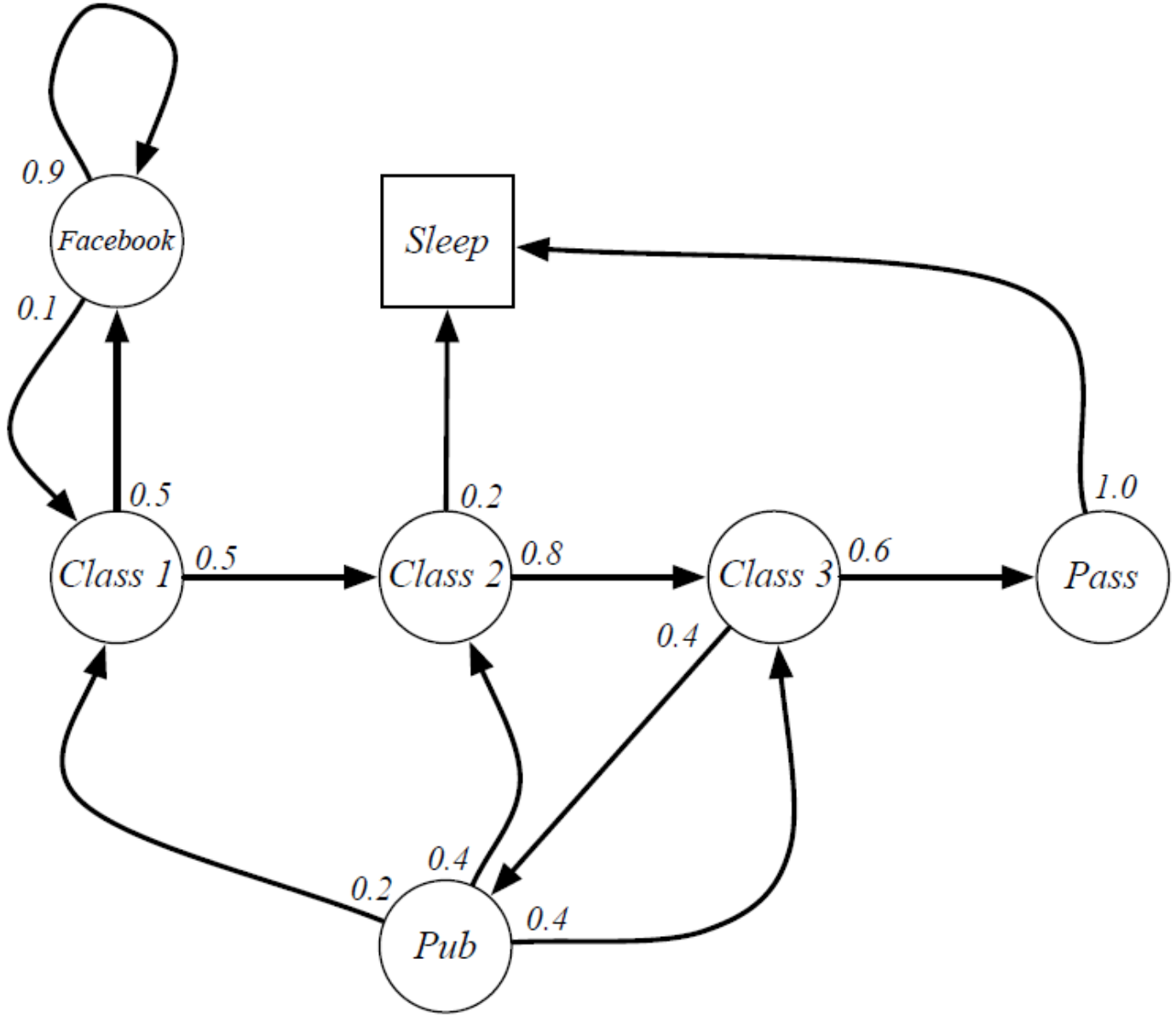
Definition

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

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$

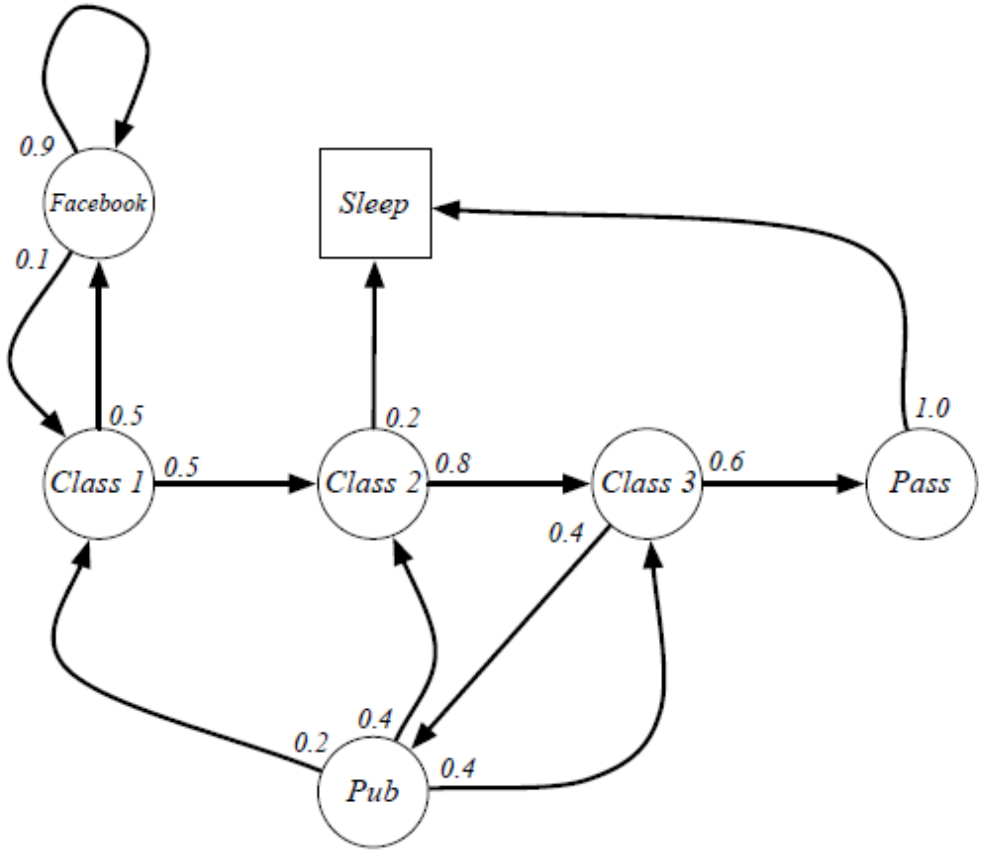
Courtesy: David Silver Slides MDP Lec2

Student Markov Chain



Courtesy: David Silver Slides MDP Lec2

Student Markov Chain : State Transition Matrix



$$\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array} \begin{array}{c} \begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\ \begin{bmatrix} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ & & & & & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{bmatrix} \end{array}$$

Courtesy: David Silver Slides MDP Lec2

Markov Reward Process

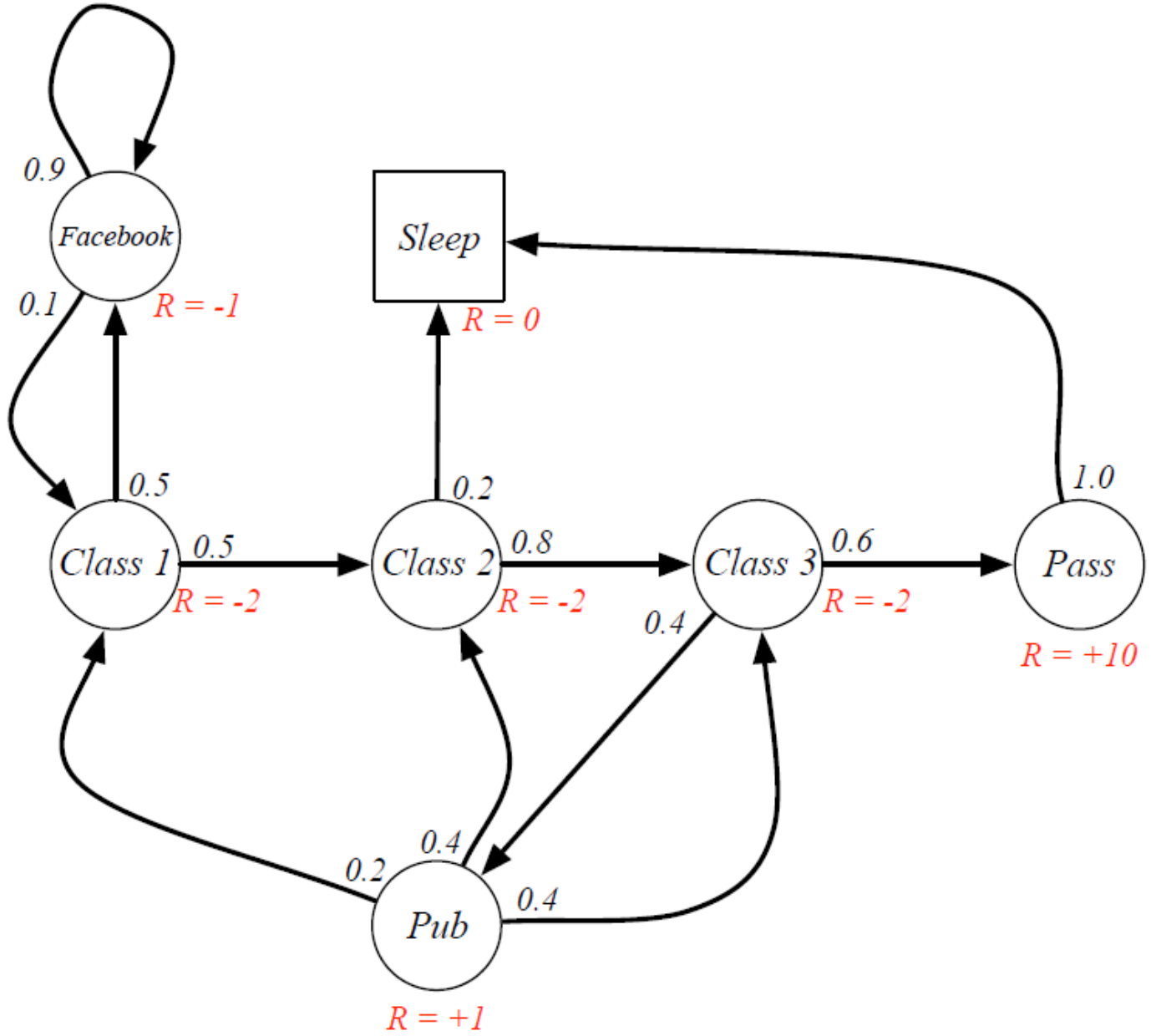
A Markov reward process is a Markov chain with values.

Definition

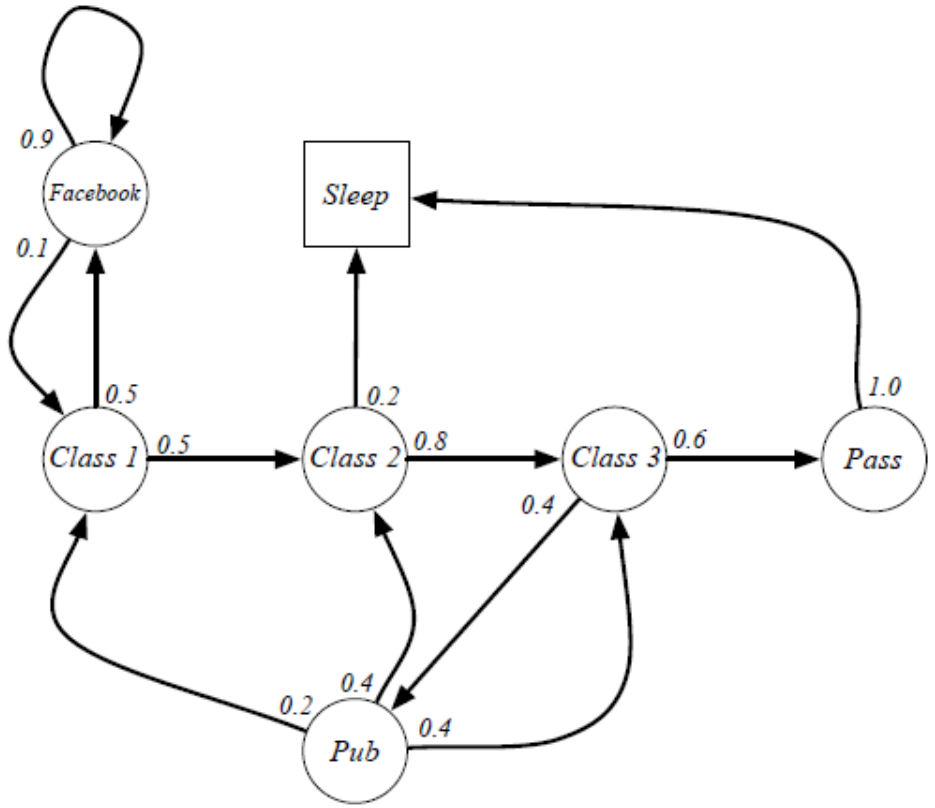
A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

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

Student MRP


0.9
Facebook
0.1
R = -1
Sleep
R = 0
0.5
0.2
1.0
Class 1
0.5
R = -2
Class 2
0.8
R = -2
Class 3
0.6
R = -2
Pass
R = +10
0.4
0.4
0.2
0.4
Pub
R = +1

Sample Episodes



Sample episodes for Student Markov Chain starting from $S_1 = \text{C1}$

$$S_1, S_2, ..., S_T$$

- 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

Value Function

The value function $v(s)$ gives the long-term value of state s

Definition

The *state value function* $v(s)$ of an MRP is the expected return starting from state s

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

Student MRP Returns

Sample returns for Student MRP:
Starting from $S_1 = \text{C1}$ with $\gamma = \frac{1}{2}$

$$G_1 = R_2 + \gamma R_3 + \ldots + \gamma^{T-2} R_T$$

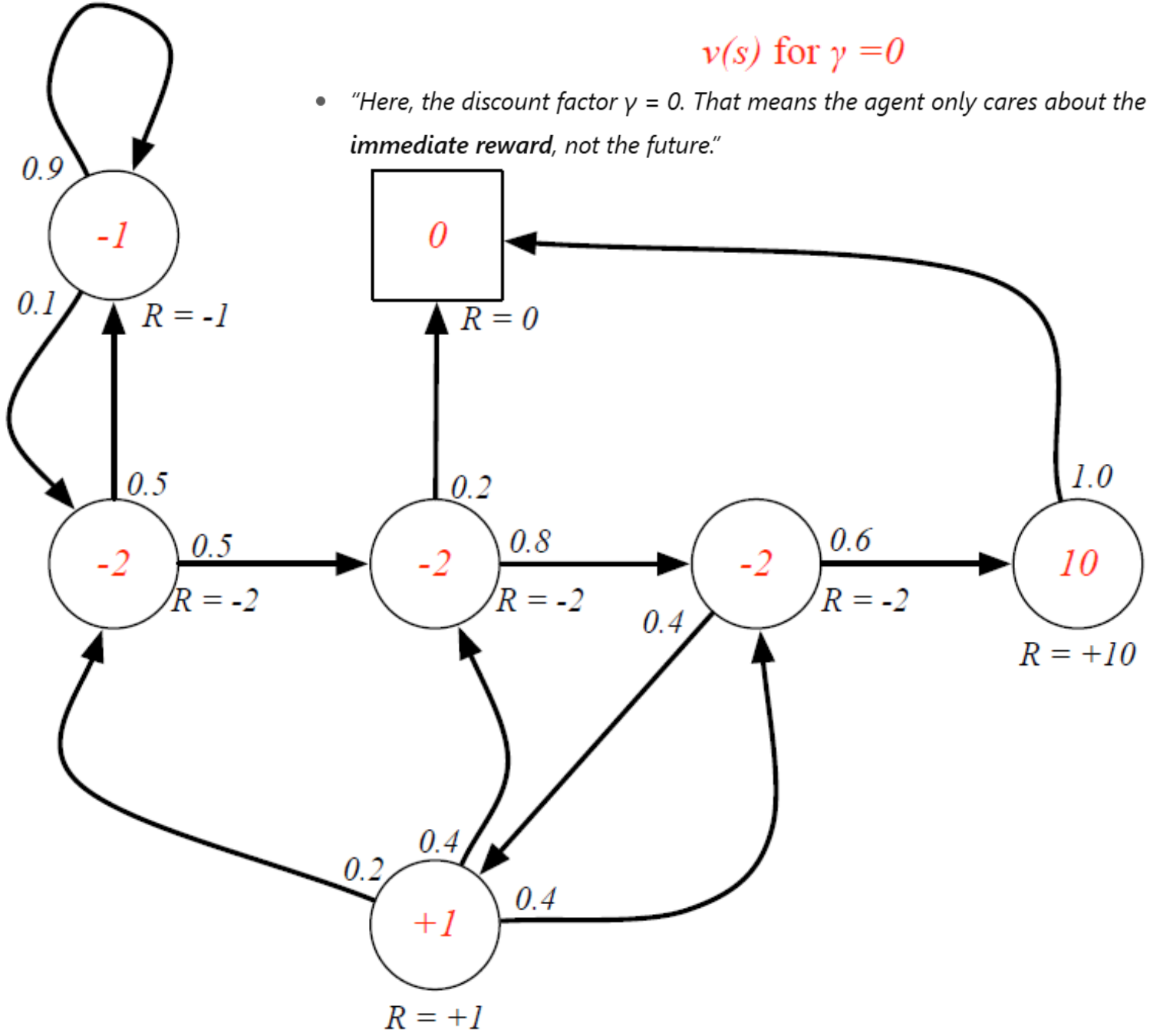
C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	=	-2.25
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	=	-3.125
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	=	-3.41
C1 FB FB C1 C2 C3 Pub C1 ... FB FB FB C1 C2 C3 Pub C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	=	-3.20

Why discount?

Most Markov reward and decision processes are discounted. Why?

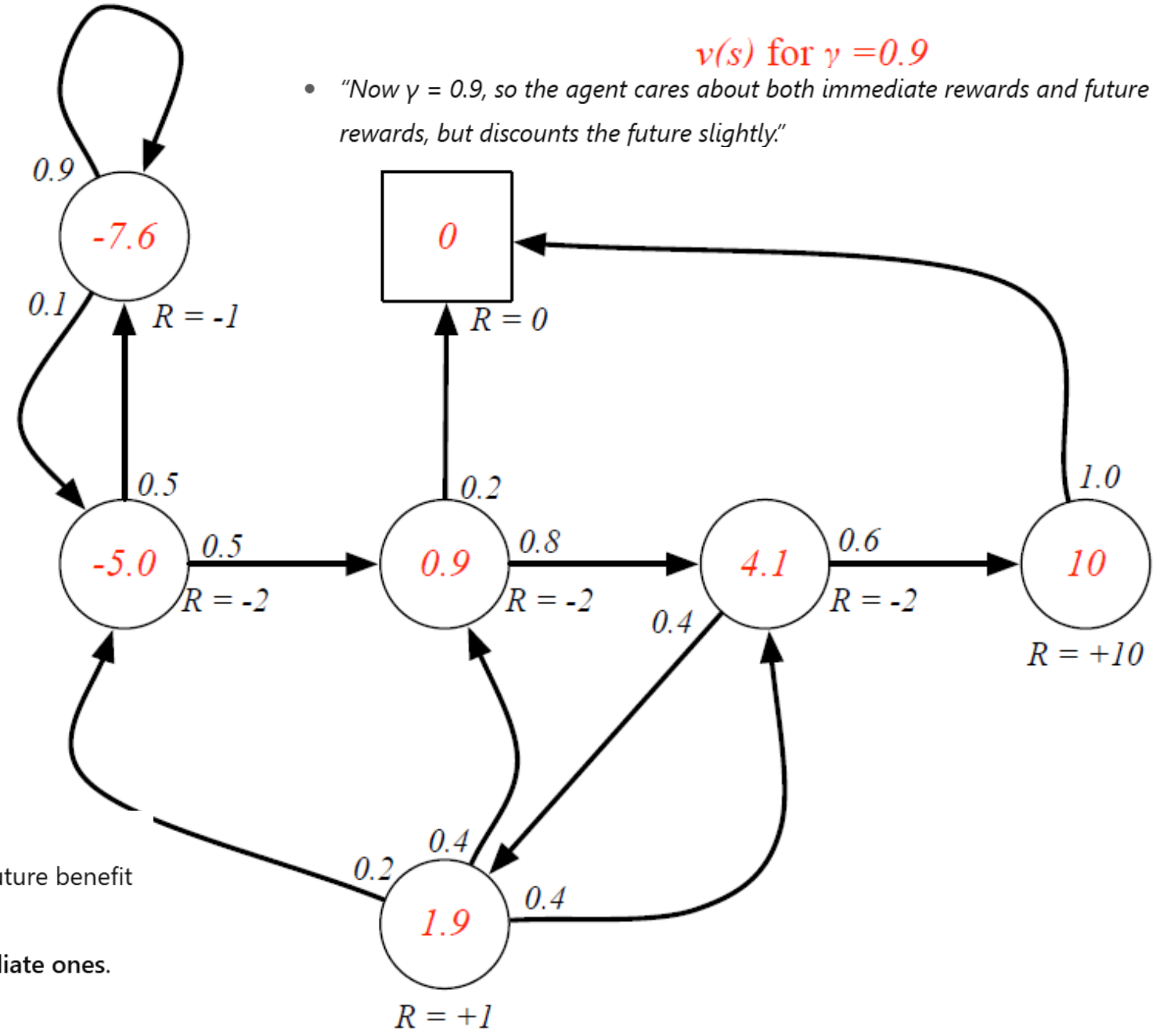
- Mathematically convenient to discount rewards
- Avoids infinite returns in cyclic Markov processes
- Uncertainty about the future may not be fully represented
- If the reward is financial, immediate rewards may earn more interest than delayed rewards
- Animal/human behaviour shows preference for immediate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

Example: State-Value Function for Student MRP



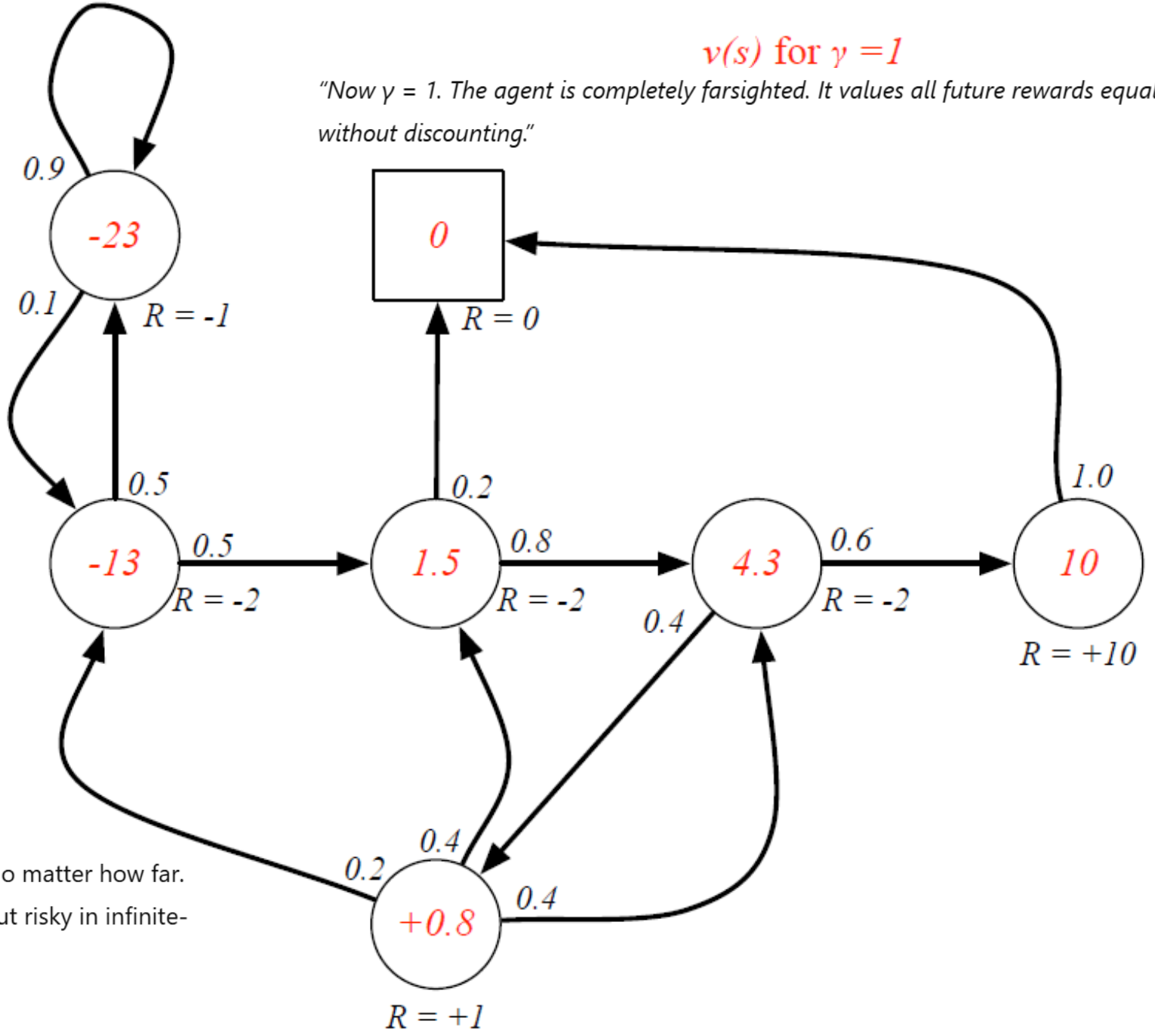
- The student is **short-sighted** — only sees today's pain or pleasure.
- They will never plan for the big payoff of **Pass (+10)**, because the future doesn't matter.
- This is like someone who says: *"Why study? It feels bad now, so I won't do it."*

Example: State-Value Function for Student MRP



- This is **balanced foresight** — the agent weighs today vs. tomorrow.
- Students may accept the −2 pain of studying because they see the future benefit of +10.
- But still, since γ < 1, long-term payoffs are **not as valuable as immediate ones**.

Example: State-Value Function for Student MRP



- This is the **ultimate planner** — cares about the entire future, no matter how far.
- Great for problems where episodes always end (like exams), but risky in infinite-horizon problems because values can blow up.

Bellman Equation for MRPs

The value function can be decomposed into two parts:

- immediate reward R_{t+1}
- discounted value of successor state $\gamma v(S_{t+1})$

$$
\begin{aligned}
v(s) &= \mathbb{E}\left[G_t \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma\left(R_{t+2} + \gamma R_{t+3} + \ldots\right) \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]
\end{aligned}
$$

Value of current state = **reward now** + **discounted value of next state**

By Law of expectation

$$\mathbb{E}[G_{t+1} \mid S_t = s] = \mathbb{E}\Big[\mathbb{E}[G_{t+1} \mid S_{t+1}] \Big| S_t = s\Big]$$

Law of expectation

The law says:

$$\mathbb{E}[X \mid Y] = \mathbb{E}\big[\,\mathbb{E}[X \mid Z] \,\big|Y\big]$$

for any random variables X, Y, Z.

Here:

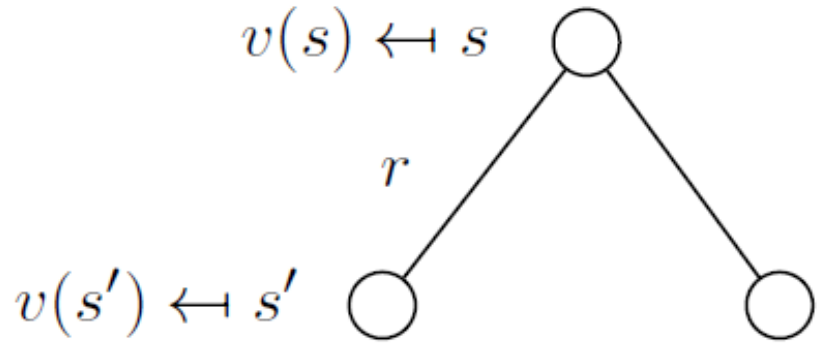
- $X = G_{t+1}$ (the random future return)
- $Y = S_t$ (current state)
- $Z = S_{t+1}$ (next state)

So:

$$\mathbb{E}[G_{t+1} \mid S_t = s] = \mathbb{E}\Big[\,\mathbb{E}[G_{t+1} \mid S_{t+1}] \,\Big|\, S_t = s\Big]$$

Bellman Equation for MRPs

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



$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

- $v(s)$:
  The **value function** of state s. It represents the *expected long-term return* if we start in state s.
- R_s:
  The **expected immediate reward** when in state s.
  (Sometimes written as $R(s)$).
- γ:
  The **discount factor** $(0 \leq \gamma \leq 1)$, which controls how much we value future rewards compared to immediate ones.
- $\sum_{s' \in S} P_{ss'} v(s')$:
  The **expected value of successor states**, weighted by their transition probabilities:
  - $P_{ss'} = \Pr[S_{t+1} = s' \mid S_t = s]$ is the probability of going from state s to s'.
  - $v(s')$ is the long-term value of being in state s'.

$$\mathbb{E}[f(X)] = \sum_{i=1}^{n} p_i \, f(x_i)$$

Base Definition of Expectation

For a **discrete random variable** X that can take values $x_1, x_2, \ldots, x_n$ with probabilities

$$P(X = x_i) = p_i,$$

the **expectation (expected value)** of a function $f(X)$ is:

$$\mathbb{E}[f(X)] = \sum_{i=1}^{n} p_i \, f(x_i)$$

Bellman Equation in Matrix Form

The Bellman equation can be expressed concisely using matrices,

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

where v is a column vector with one entry per state

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{11} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

Solving the Bellman Equation

- The Bellman equation is a linear equation
- It can be solved directly:

$$
\begin{aligned}
v &= \mathcal{R} + \gamma \mathcal{P} v \\
(I - \gamma \mathcal{P})\, v &= \mathcal{R} \\
v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R}
\end{aligned}
$$

- Computational complexity is $O(n^3)$ for n states
- Direct solution only possible for small MRPs
- There are many iterative methods for large MRPs, e.g.
  - Dynamic programming
  - Monte-Carlo evaluation
  - Temporal-Difference learning

we saw the **Markov Reward Process**. This was just a Markov chain with values. “In an MRP, the student just drifts along according to probabilities. No choices are made.”

But real life isn’t just drifting. We actually make decisions. Should I go to class or go to the pub? Should I quit or keep studying? These choices change what happens next.”

So we extend the MRP to include actions. That gives us a **Markov Decision Process**

Markov Decision Process

A Markov decision process (MDP) is a Markov reward process with decisions. It is an *environment* in which all states are Markov.

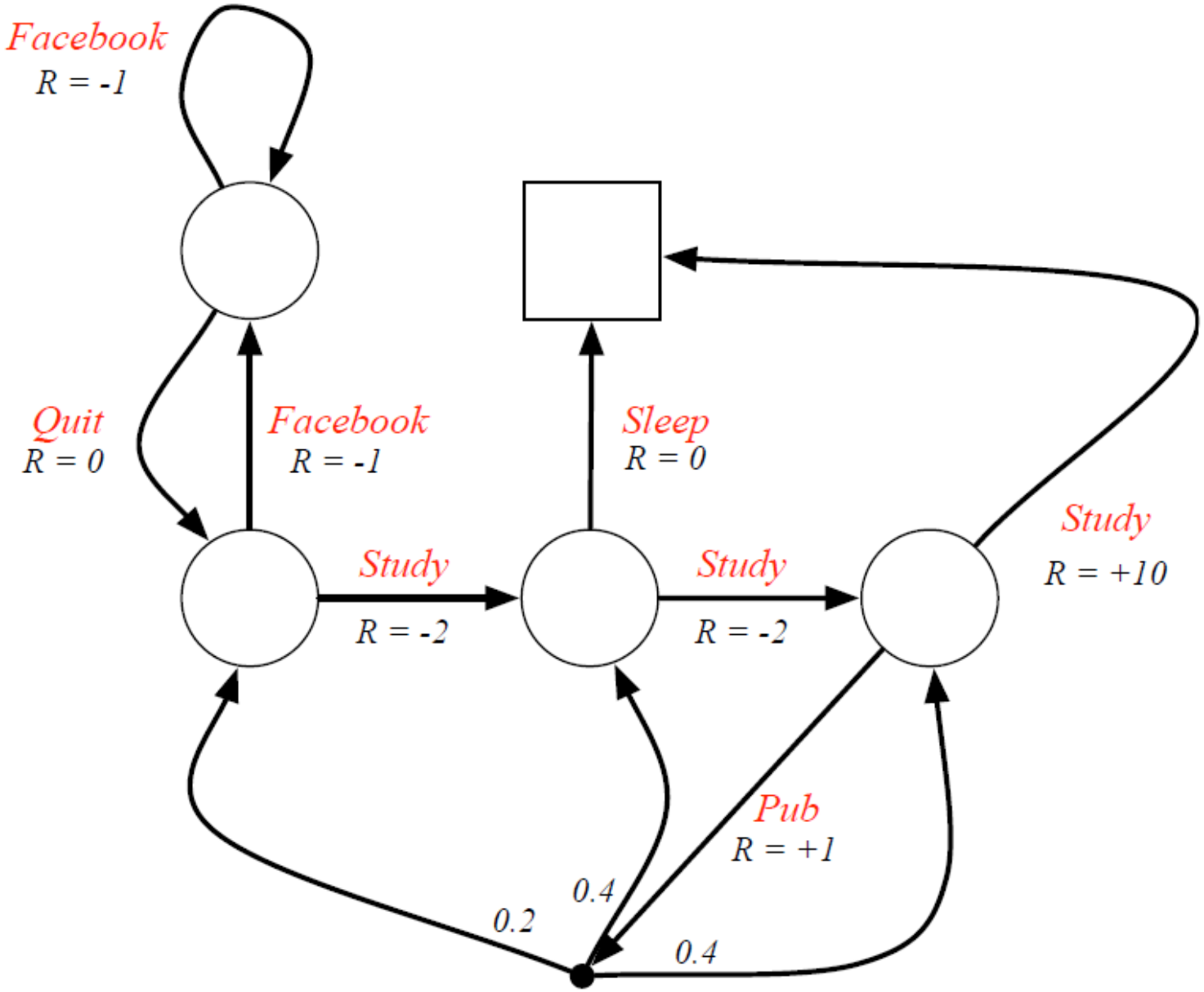
Definition

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

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

Example: Student MDP

"Now the agent is active — the student decides what to do, and that decision changes the path and the reward."



We’ve now defined MDPs — states, actions, rewards, transitions. But the big question is: how does the agent decide which action to take? That’s where a **policy** comes in.

So the policy is the agent’s brain: it tells us what to do in each state.

Policy (1)

- A_t: the action chosen at time t.
- $\pi(\cdot \mid S_t)$: the policy's probability distribution over actions given the current state S_t.
- "$\sim$" means "is sampled from" or "is drawn according to".

Definition

A *policy* π is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

- A policy fully defines the behaviour of an agent
- MDP policies depend on the current state (not the history)
- i.e. Policies are *stationary* (time-independent), $A_t \sim \pi(\cdot|S_t), \forall t > 0$

👉 *"At every timestep t, the action A_t is drawn from the policy's action distribution given the current state S_t."*

- Policy π: *"In Class, study with 0.7 probability, Facebook with 0.3 probability. At Pub, always stay (prob 1). At Sleep, terminate."*
- That's a policy — it tells us what the agent tends to do.

Policy(2)

Now let's see what happens when we fix a policy. Once a policy is chosen, the MDP reduces to a simpler process — essentially back to an MRP

Once actions are folded into the probabilities, you're left with an MRP

$$\langle S, A, P, R, \gamma \rangle + \pi \quad \rightarrow \quad \langle S, P^{\pi}, R^{\pi}, \gamma \rangle$$

- Given an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and a policy π
- The state sequence $S_1, S_2, \ldots$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^{\pi} \rangle$
- The state and reward sequence $S_1, R_2, S_2, \ldots$ is a Markov reward process $\langle \mathcal{S}, \mathcal{P}^{\pi}, \mathcal{R}^{\pi}, \gamma \rangle$
- where

Transition probabilities under policy π:

→ weighted average of transitions based on policy.

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

- Reward under policy π:

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

→ expected reward, given what the policy tends to do.

Value Function

Once we fix a policy, we want to evaluate: how good is it? That's where value functions come in

Definition

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

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

Definition: expected return starting from state s, and then following policy π.

if I start in this state and keep following the policy, what's the average long-term reward I can expect?

Definition

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

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

Definition: expected return starting from state s, **taking action a first**, then following policy π.

This lets us compare actions. From this state, how good is it if I take this action and then follow the policy?

Bellman Expectation Equation

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

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

The action-value function can similarly be decomposed,

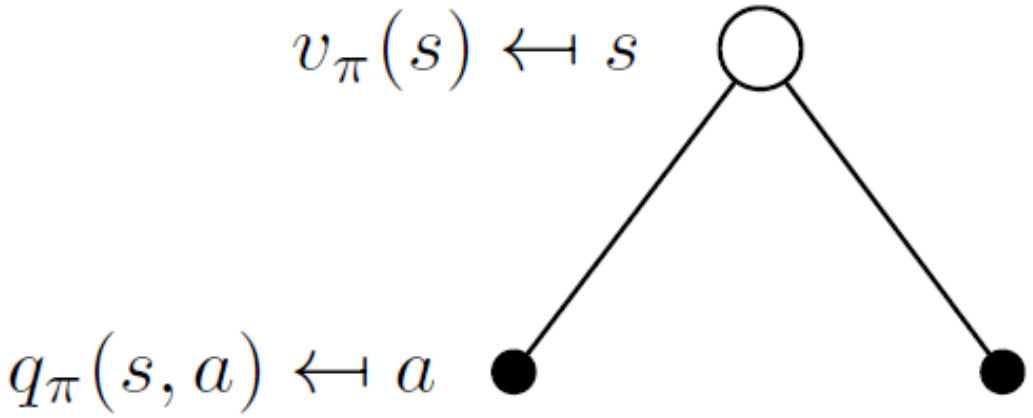
$$q_\pi(s, a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a \right]$$

Bellman Expectation Equation for V_π (1)

When you're at state s, your action is not deterministic — it's chosen according to the policy distribution $\pi(a|s)$.

So to get the value of the state, you average over all possible actions:

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

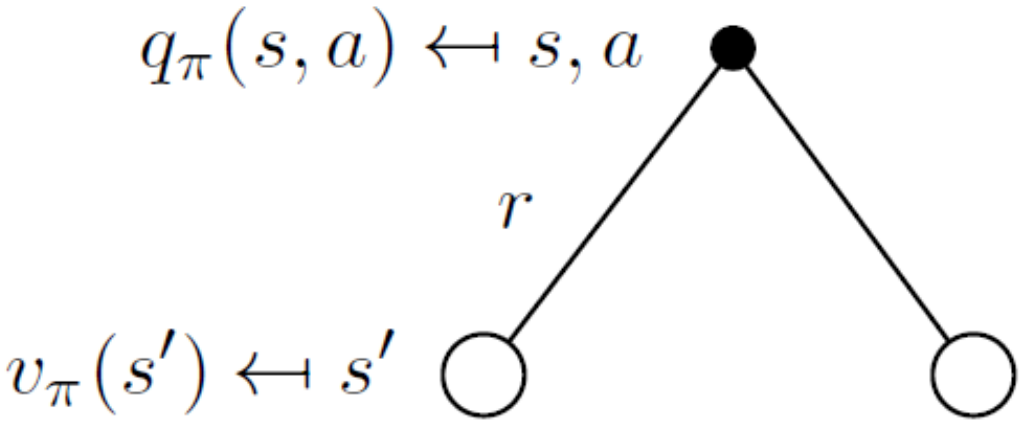


If I knew exactly which action a to take in state s, I can talk about the expected return from that choice.
policy might choose different actions with different probabilities. So the value of the state must be the **average** of all action values, weighted by how likely the policy is to pick them

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

This is just the law of total expectation: expected value of returns = expected value over actions of expected return given each action

Bellman Equation for q_π (1)



R_s^a: expected immediate reward for taking action a in state s.

$P_{ss'}^a$: probability of transitioning from s to s' given action a.

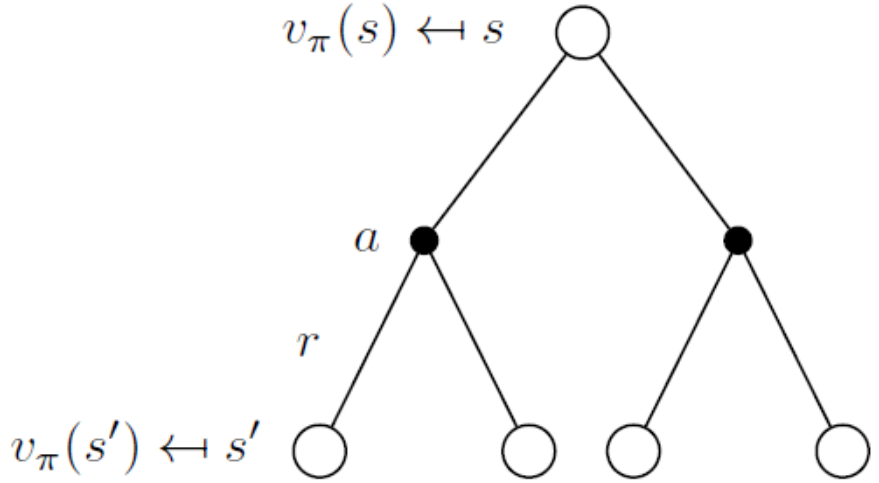
$v_\pi(s')$: value of the next state.

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

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

This is like $v_\pi(s)$, but we condition on taking a specific action a in state s.

Bellman Expectation Equation for v_π (2)

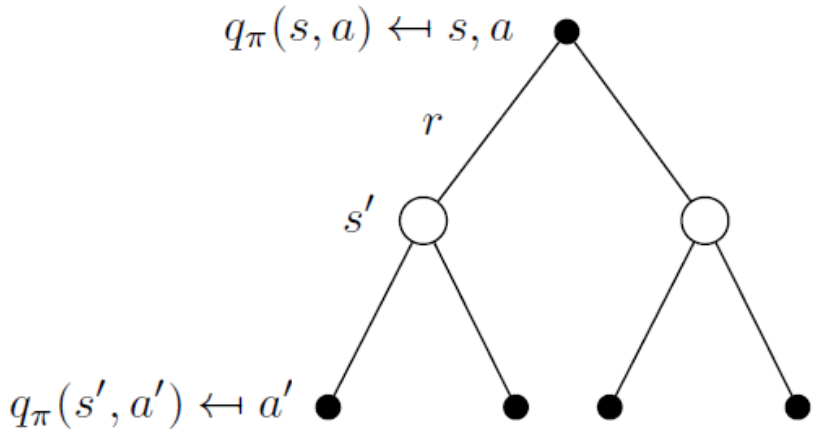


$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

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

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

Bellman Expectation Equation for q_π (2)



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

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

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

Namah Shivaya