Reinforcement Learning III

Lecture 21

Markov Models

	States are **Fully Observable**	States are **Partially Observable**
Autonomous (no actions; make predictions)	Markov Chain, Markov Reward Process	Hidden Markov Model (HMM)
Controlled (can take actions)	Markov Decision Process (MDP)	Partially Observable Markov Decision Process (POMDP)

Applications

HMMs: time series ML, e.g. speech + handwriting recognition, bioinformatics

MDPs: used extensively for reinforcement learning

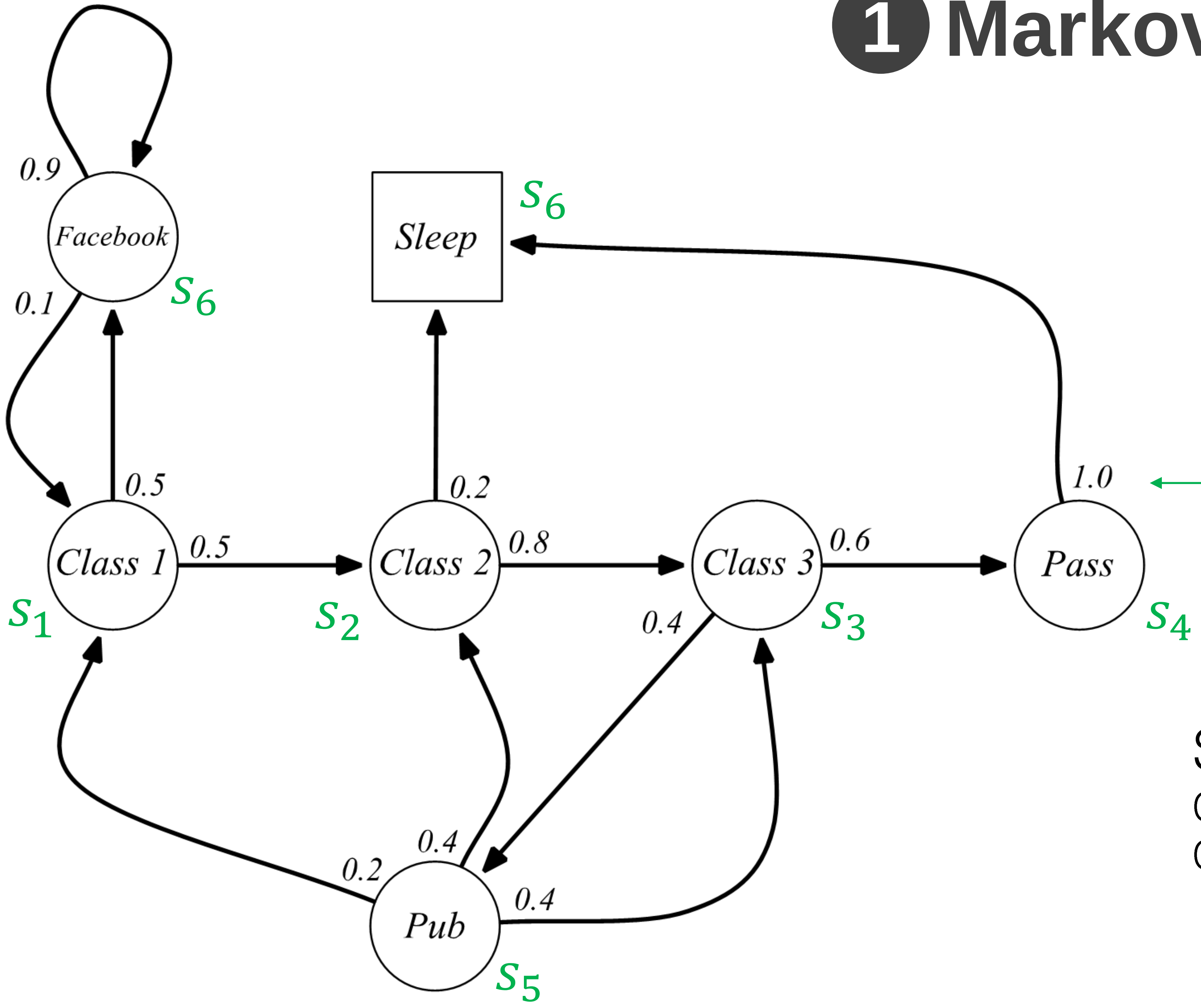
Building blocks for the full RL problem

1	Markov Chain	{state space S, transition probabilities P}
2	Markov Reward Process (MRP)	$\{S, P,$ + rewards R, discount rate $\gamma\}$ adds rewards (and values)
3	Markov Decision Process (MDP)	$\{S, P, R, \gamma,$ + actions $A\}$ adds decisions (i.e. the ability to control)

MDPs form the basis for most reinforcement learning environments

Adapted from David Silver, 2015

❶ Markov Chain Example



$P_{46} = P_{ss'}$

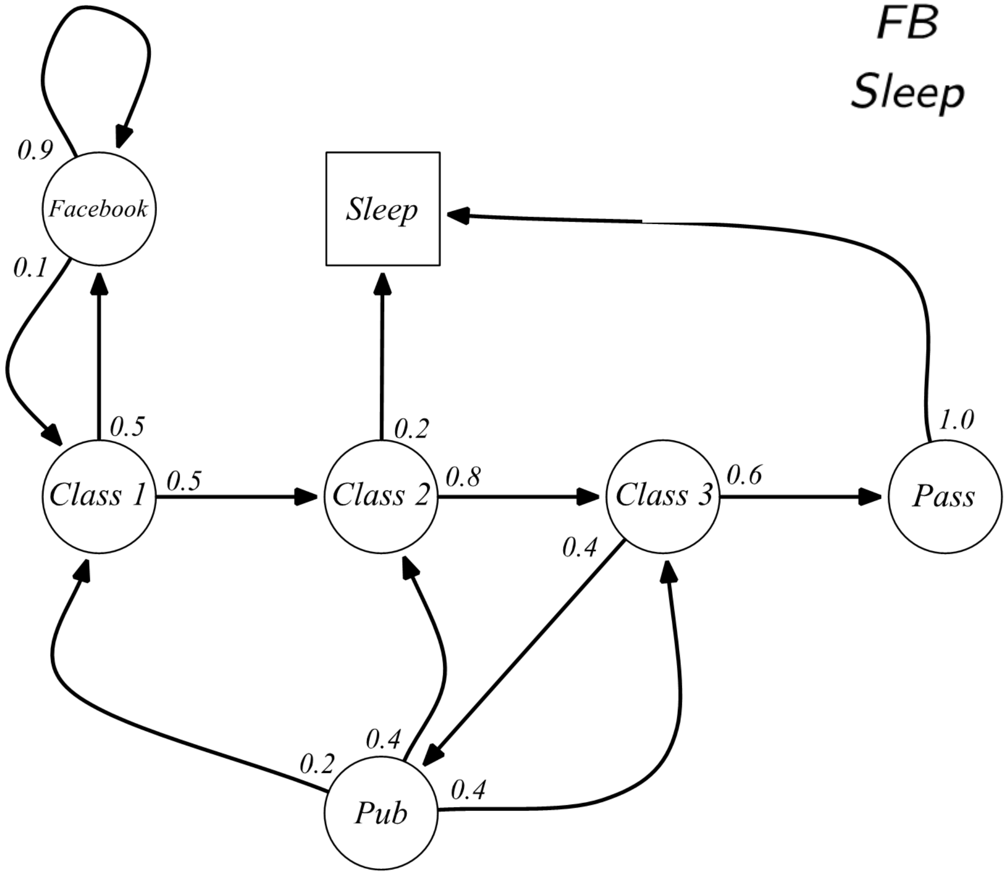
Components:
State space S,
Transition probabilities P

Sample Episodes:
C1,C2,Sleep
C1,FB,FB,FB,C1,C2,C3,Pass,Sleep

Example from David Silver, UCL, 2015

Markov Chain

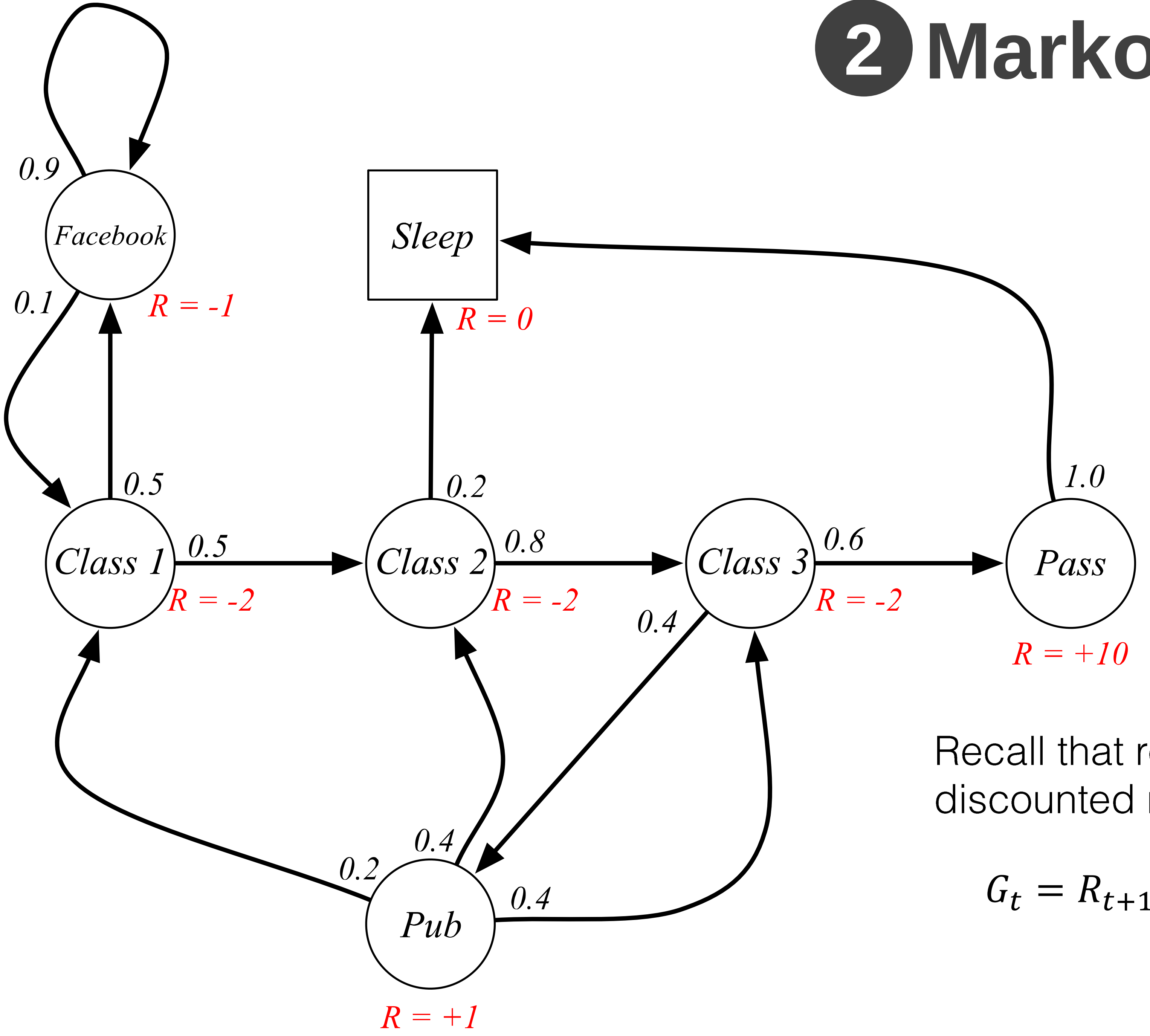
$$
\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array}
\begin{array}{c}
\begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\
\left[\begin{array}{ccccccc}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & &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



State transition probability matrix, $P_{ss'}$

Example from David Silver, UCL, 2015

❷ Markov Reward Process



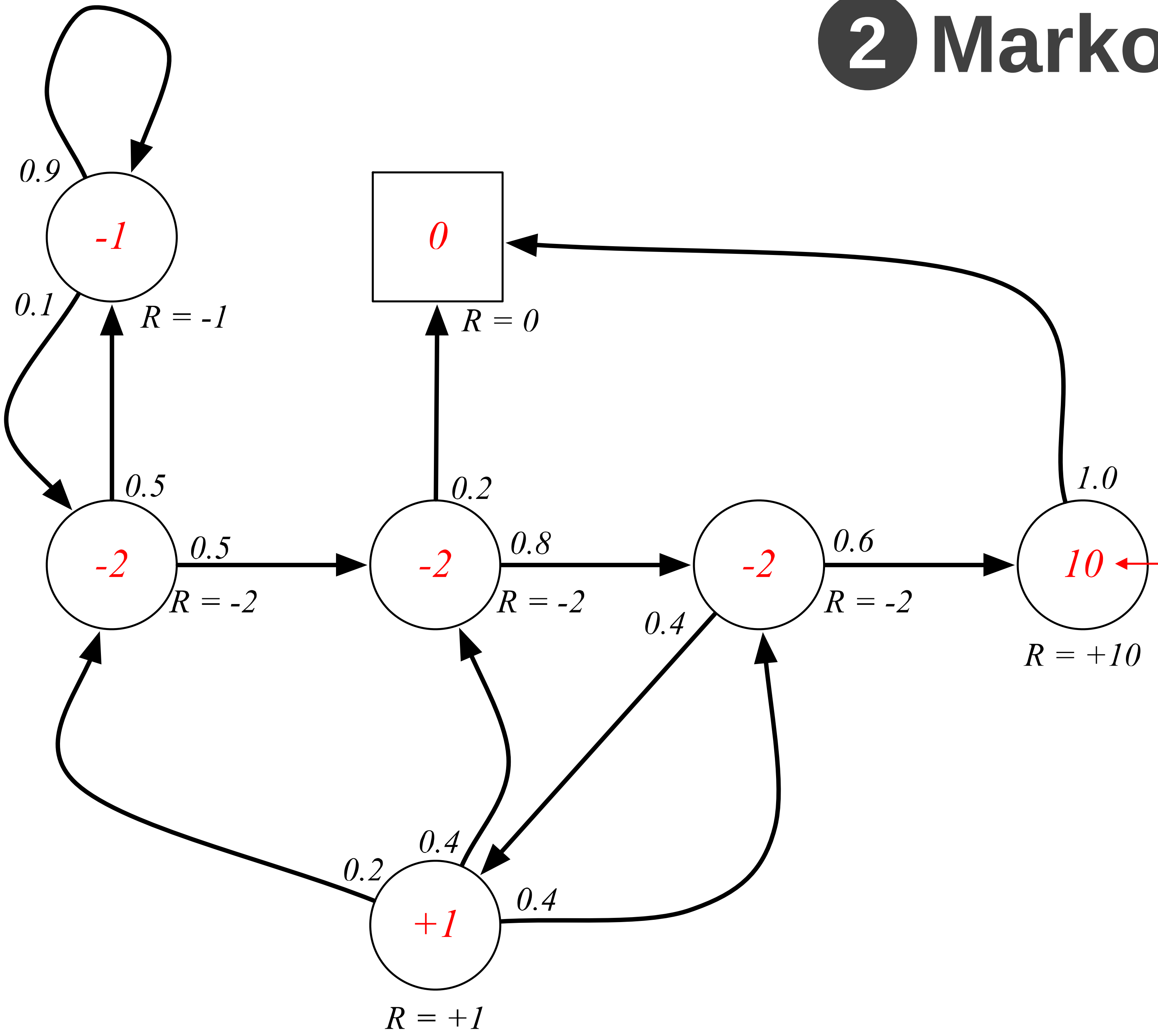
Components:
State space S,
Transition probabilities, P
Rewards, R
Discount rate, γ

Recall that returns, let's call G_t, are the total discounted rewards from time t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Example from David Silver, UCL, 2015

2 Markov Reward Process



Components:
State space S,
Transition probabilities, P
Rewards, R
Discount rate, γ

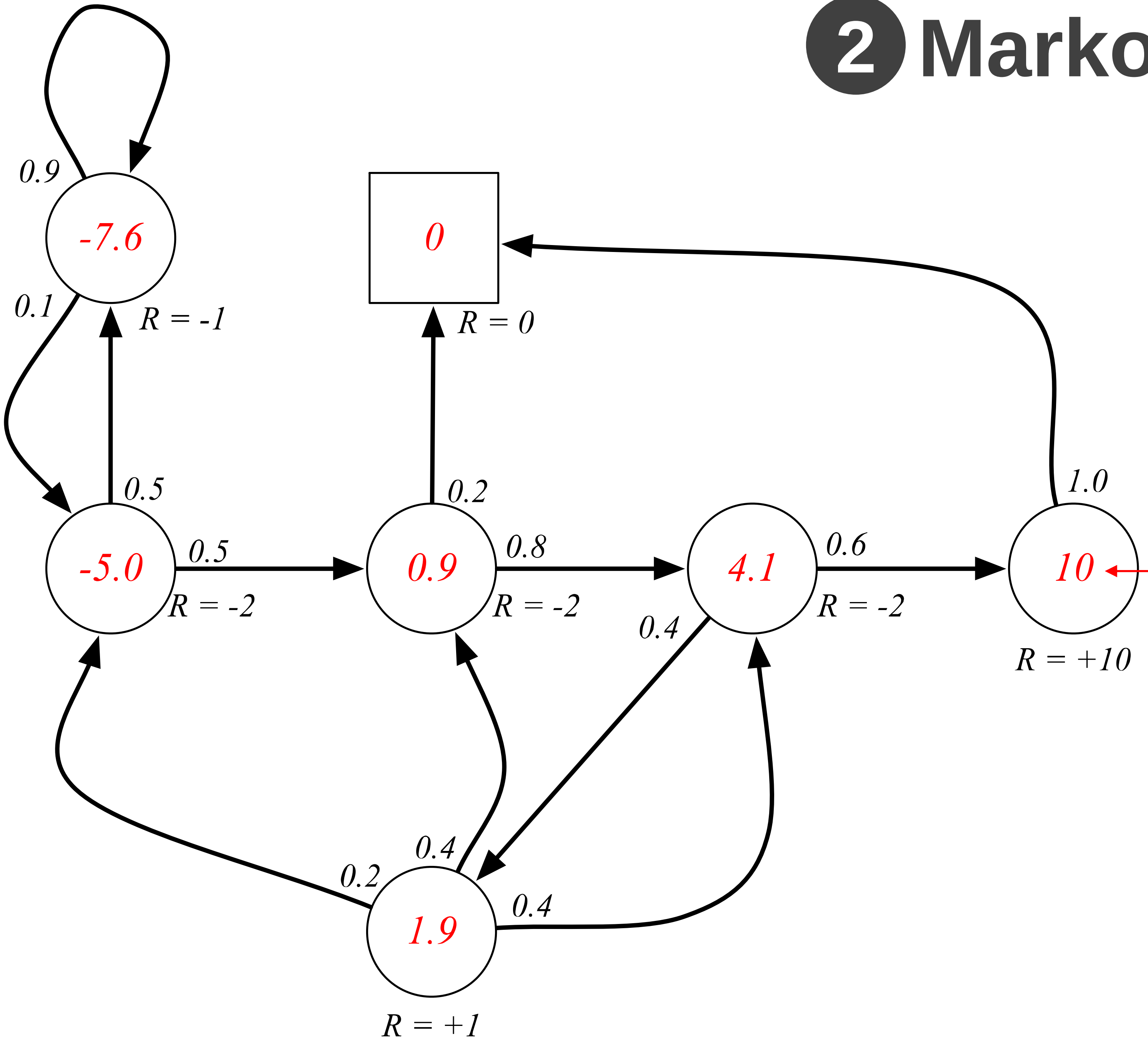
$v(s)$ for $\gamma = 0$

State value function $v(s)$ is the expected total reward (into the future)

$$v(s) = E[G_t | S = s_t]$$

Example from David Silver, UCL, 2015

❷ Markov Reward Process



Components:

State space S,

Transition probabilities, P

Rewards, R

Discount rate, γ

$v(s)$ for $\gamma = 0.9$

State value function $v(s)$ is the expected total reward (into the future)

$$v(s) = E[G_t | S = s_t]$$

Example from David Silver, UCL, 2015

“Backup” property of state value functions

$v(s) = E[G_t | S = s_t]$ where $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots$

$$= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots | S = s_t]$$

$$= E[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} \dots) | S = s_t]$$

$$= E[R_{t+1} + \gamma G_{t+1} | S = s_t]$$

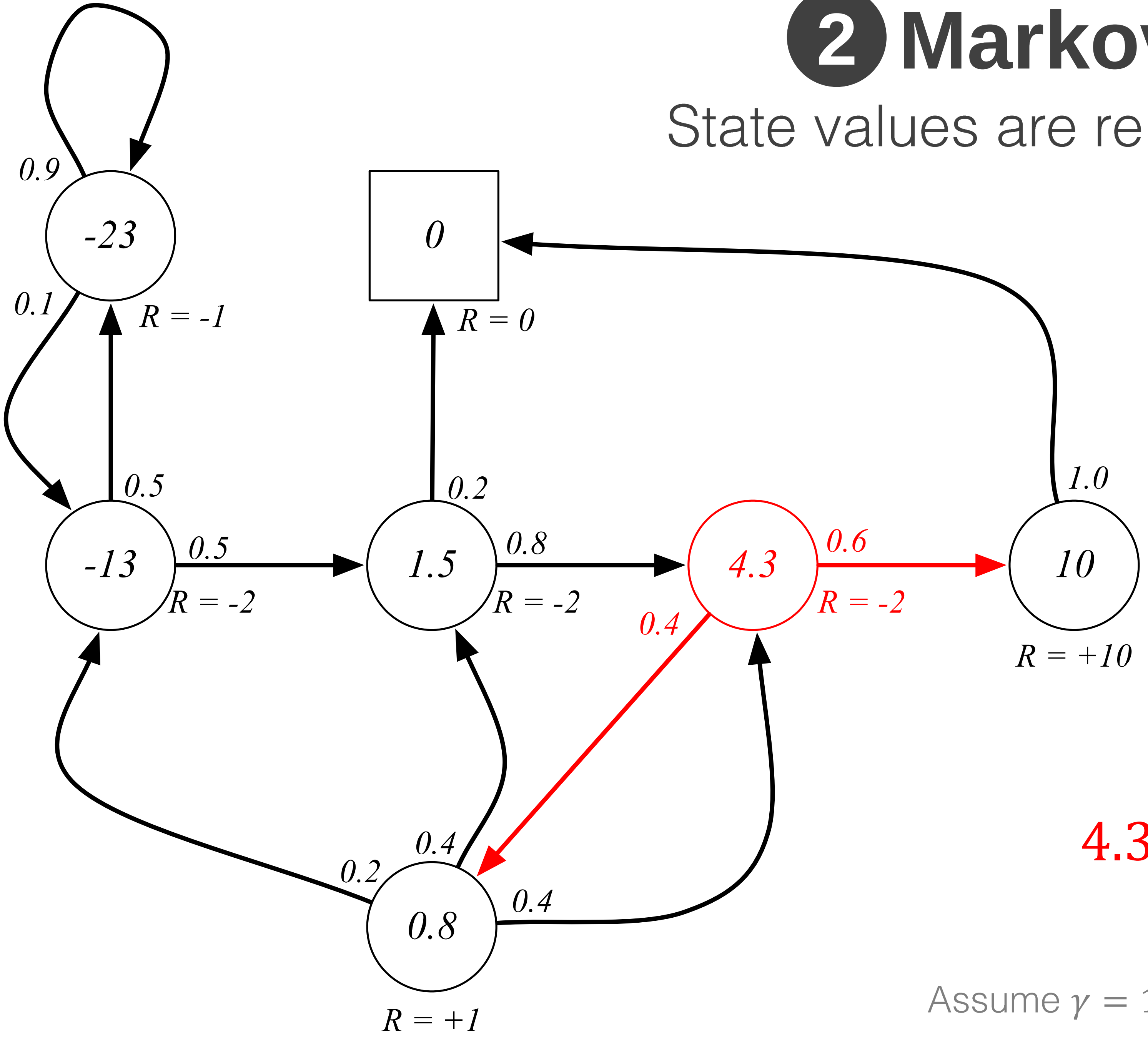
$$= E[R_{t+1} + \gamma v(s_{t+1}) | S = s_t]$$

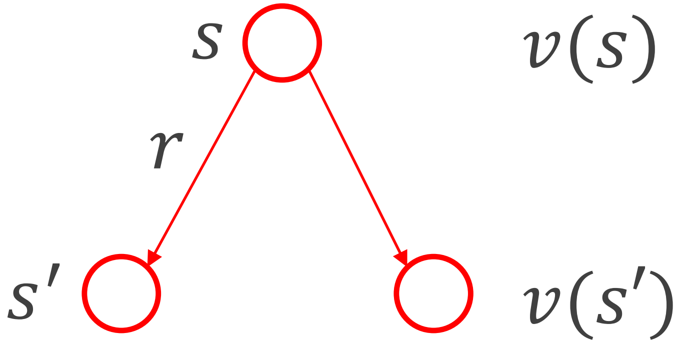
This recursive relationship is a version of the **Bellman Equation**

Example from David Silver, UCL, 2015

2 Markov Reward Process

State values are related to neighboring states





possible states we could transition to from s

$$v(s) = E[R_s + \gamma v(s')|s]$$

$$v(s) = R_s + \gamma \sum_{s'} P_{ss'} v(s')$$

$$4.3 = -2 + 0.6 \times 10 + 0.4 \times 0.8$$

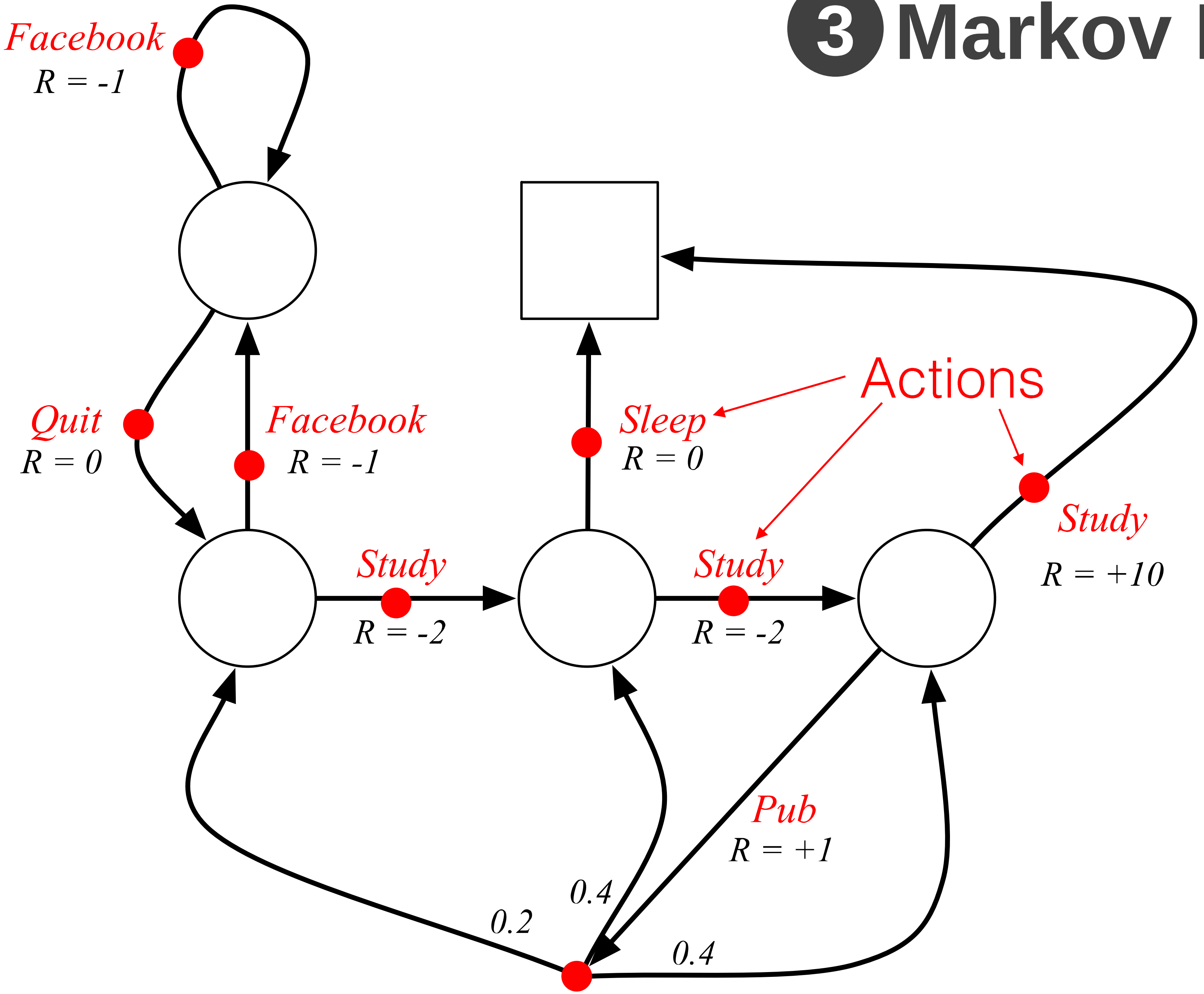
Notation: $s = s_t$ and $s' = s_{t+1}$

$R_s = E[R_{t+1}|S_t = s]$

Assume $\gamma = 1$

Example from David Silver, UCL, 2015

3 Markov Decision Process



Components:

State space S,

Transition probabilities, P

Rewards, R

Discount rate, γ

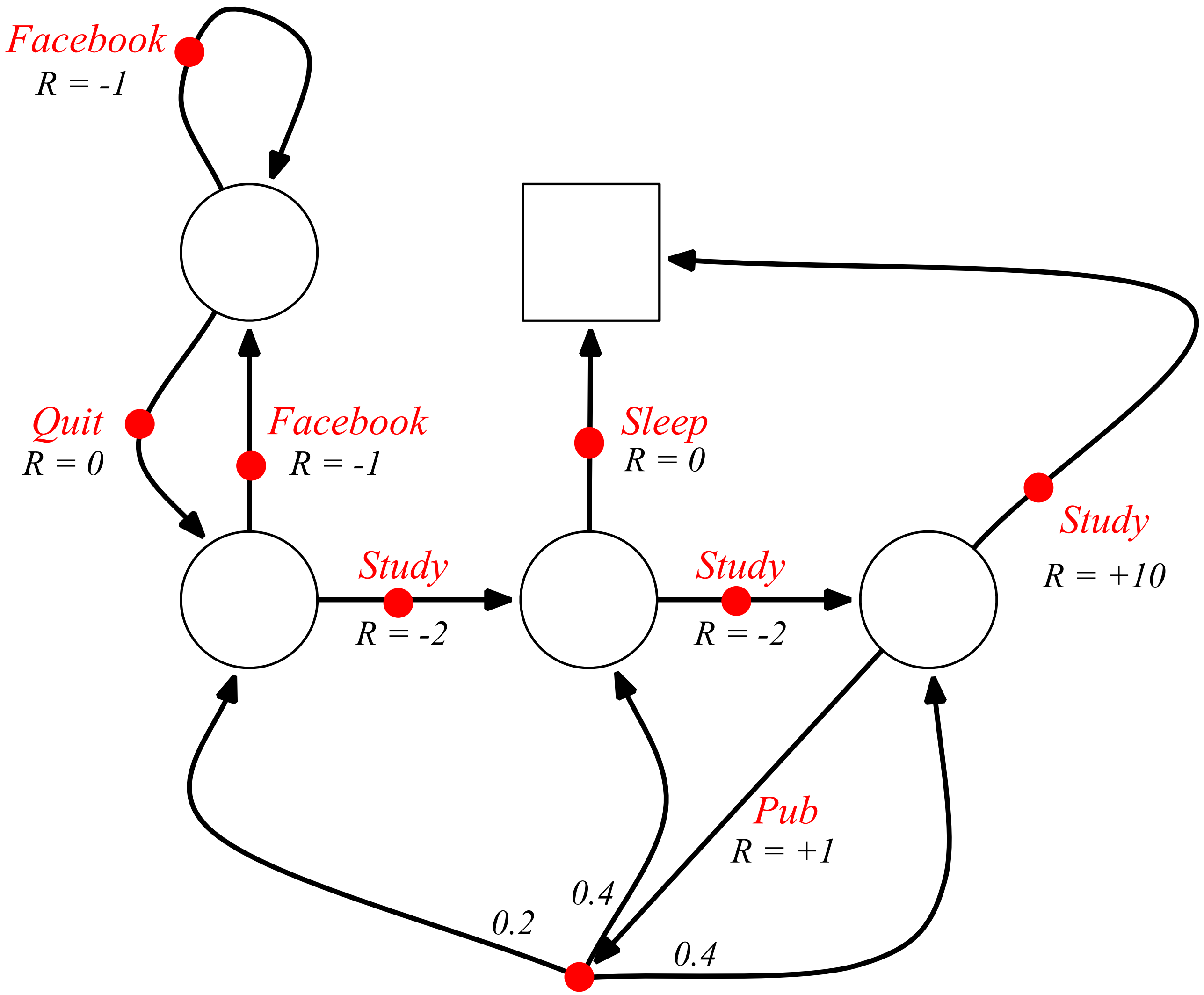
Actions, A

Adds interaction with the environment

An agent in a state chooses an action, the environment (the MDP) provides a reward and the next state

Example from David Silver, UCL, 2015

3 Markov Decision Process



Policy (how we choose actions)

(can be stochastic or deterministic)

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

State value function

(expected return from state s, and following policy π)

$$v_\pi(s) = E[G_t|s]$$

$$v_\pi(s) = E[R_s^a + \gamma v_\pi(s')|s]$$

Action value function

(expected return from state s, taking action a, and following policy π)

$$q_\pi(s,a) = E[G_t|s,a]$$

$$q_\pi(s,a) = E[R_s^a + \gamma q_\pi(s',a')|s,a]$$

$R_s^a = E[r_{t+1}|S_t = s, A_t = a]$

Example from David Silver, UCL, 2015

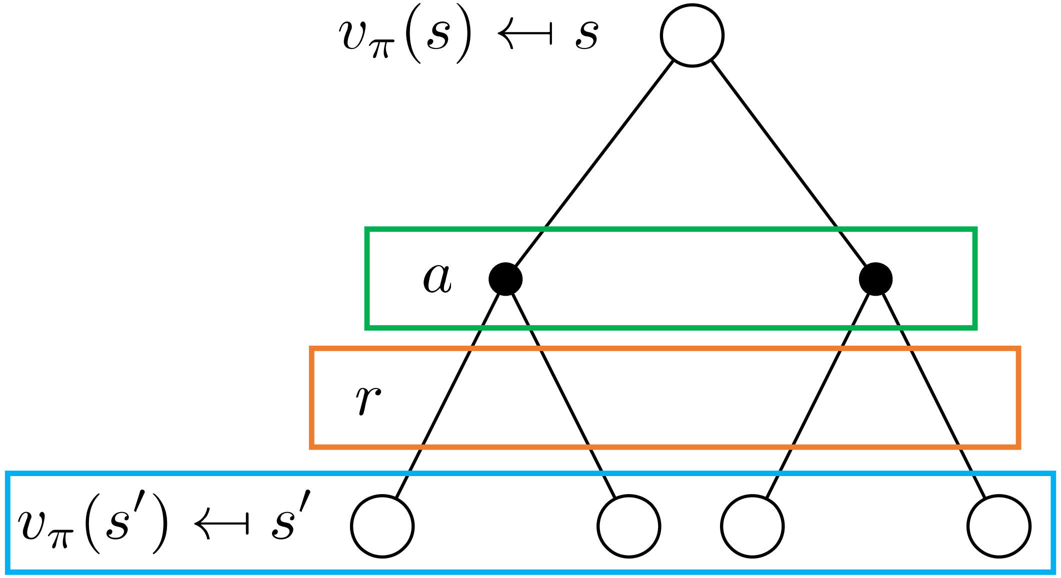
Bellman Expectation Equations for the state value function

(expected return from state s, and following policy π)

$$v_\pi(s) = E[G_t|s]$$

$$v_\pi(s) = E[R_s^a + \gamma v_\pi(s')|s]$$

$$R_s^a = E[R_{t+1}|S_t = s, A_t = a]$$



Expectation over the possible actions

Expectation over the rewards

(based on state and choice of action)

Expectation over the next possible states

$$v_\pi(s) = \underbrace{\sum_a \pi(a|s)}\left(\underbrace{R_s^a} + \gamma \underbrace{\sum_{s'} P_{ss'}^a v_\pi(s')}\right)$$

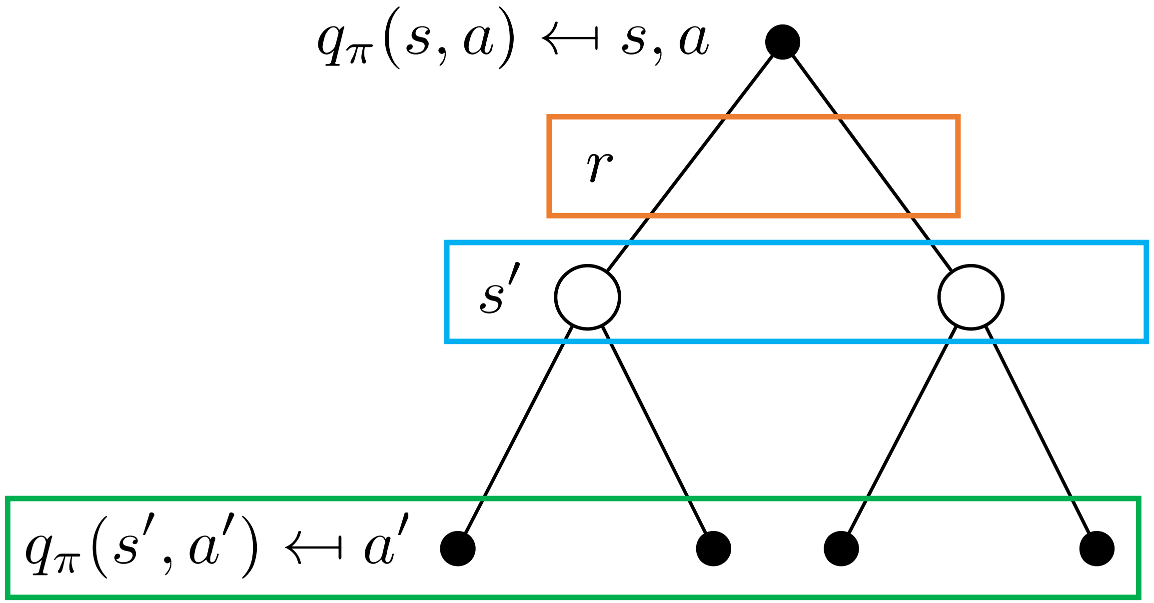
Bellman Expectation Equations for the **action value** function

(expected return from state s, taking action a, then following policy π)

$$q_\pi(s,a) = E[G_t|s,a]$$

$$q_\pi(s,a) = E[R_s^a + \gamma q_\pi(s',a')|s,a]$$

$$R_s^a = E[R_{t+1}|S_t = s, A_t = a]$$



Expectation over the rewards

(based on state and choice of action)

Expectation over the next possible states

Expectation over the possible actions

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

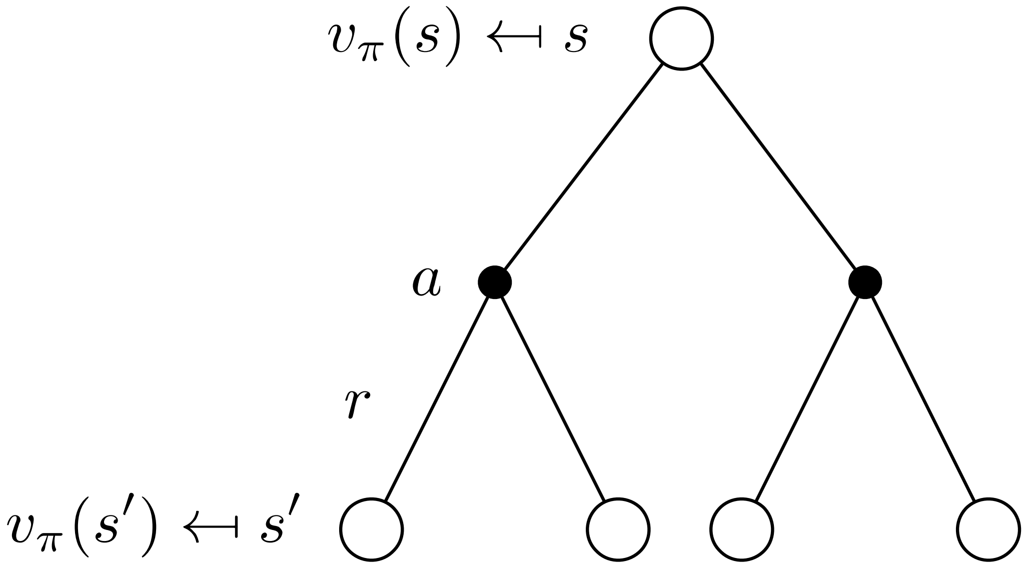
Bellman Expectation Equations

State value function

(expected return from state s, and following policy π)

$$v_\pi(s) = E[G_t|s]$$

$$v_\pi(s) = E[R_s^a + \gamma v_\pi(s')|s]$$



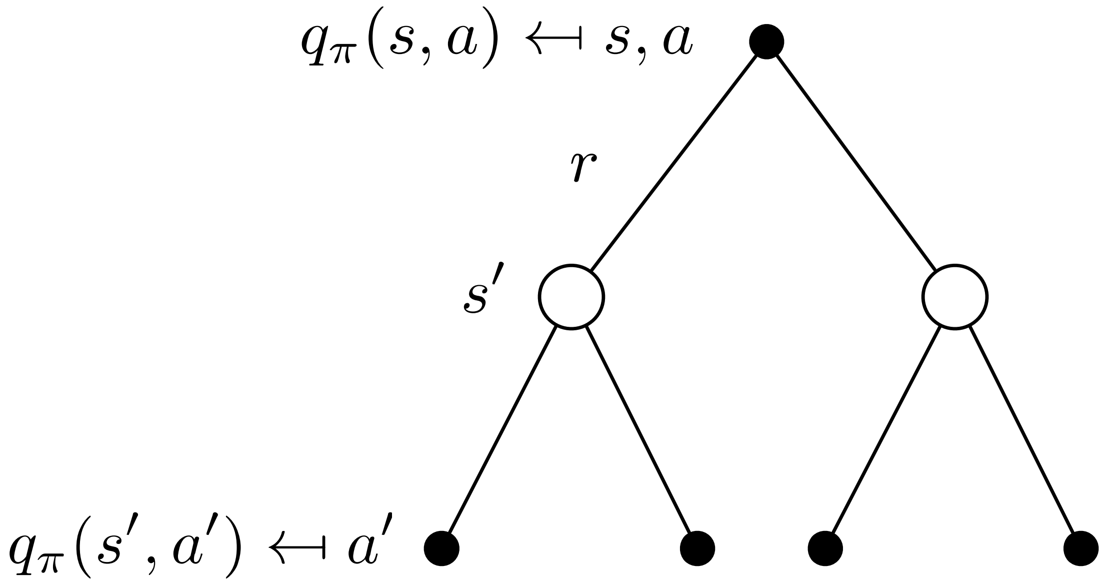
$$v_\pi(s) = \sum_a \pi(a|s)\left(R_s^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s')\right)$$

Action value function

(expected return from state s, taking action a, then following policy π)

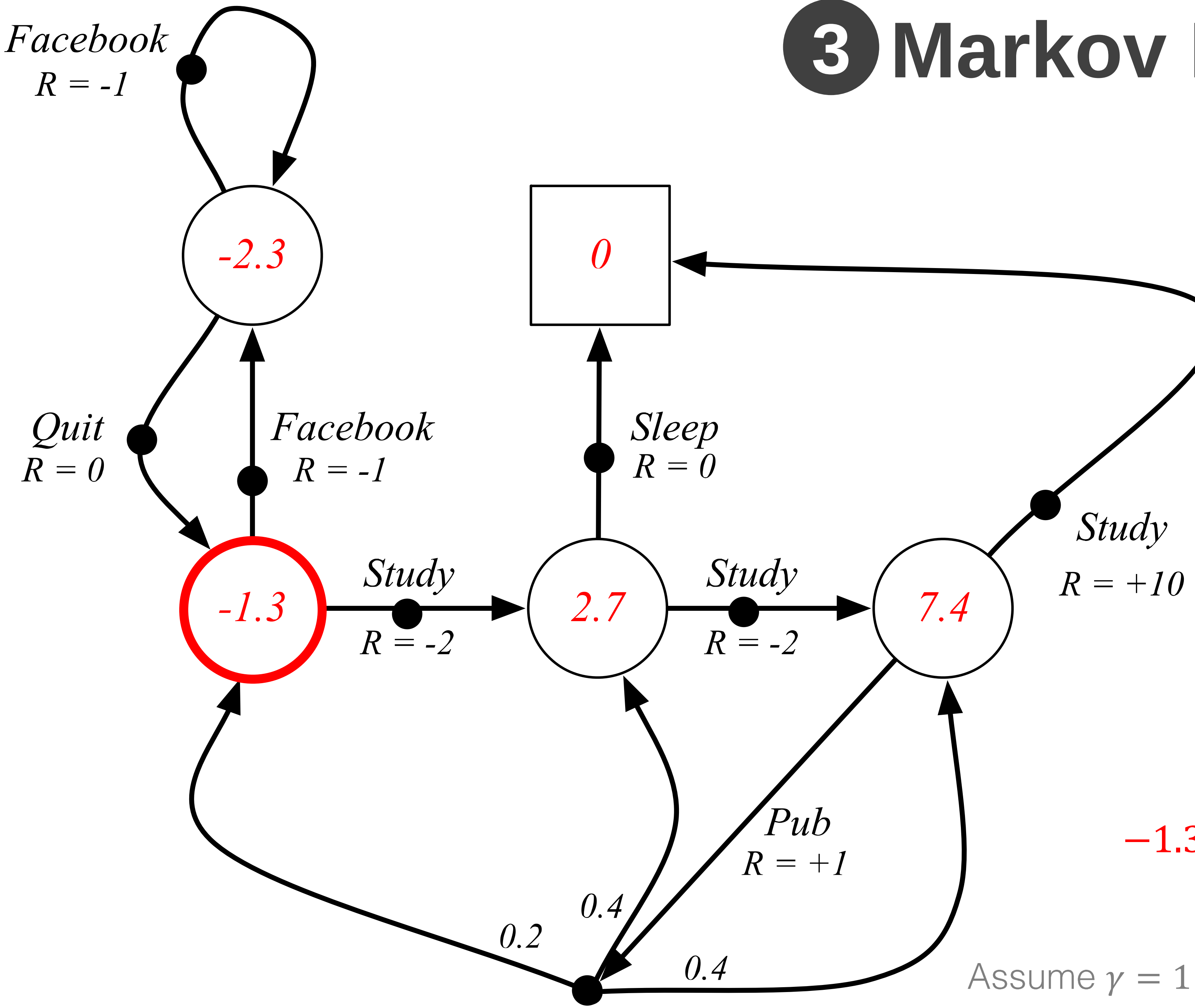
$$q_\pi(s,a) = E[G_t|s,a]$$

$$q_\pi(s,a) = E[R_s^a + \gamma q_\pi(s',a')|s,a]$$



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

❸ Markov Decision Process



State value function, $v_\pi(s)$

Assumptions:

$$\pi(a|s) = \frac{1}{\text{\# of reachable states}}$$

(randomly select an action)

$$\gamma = 1$$

(no discounting)

Bellman Expectation Equation

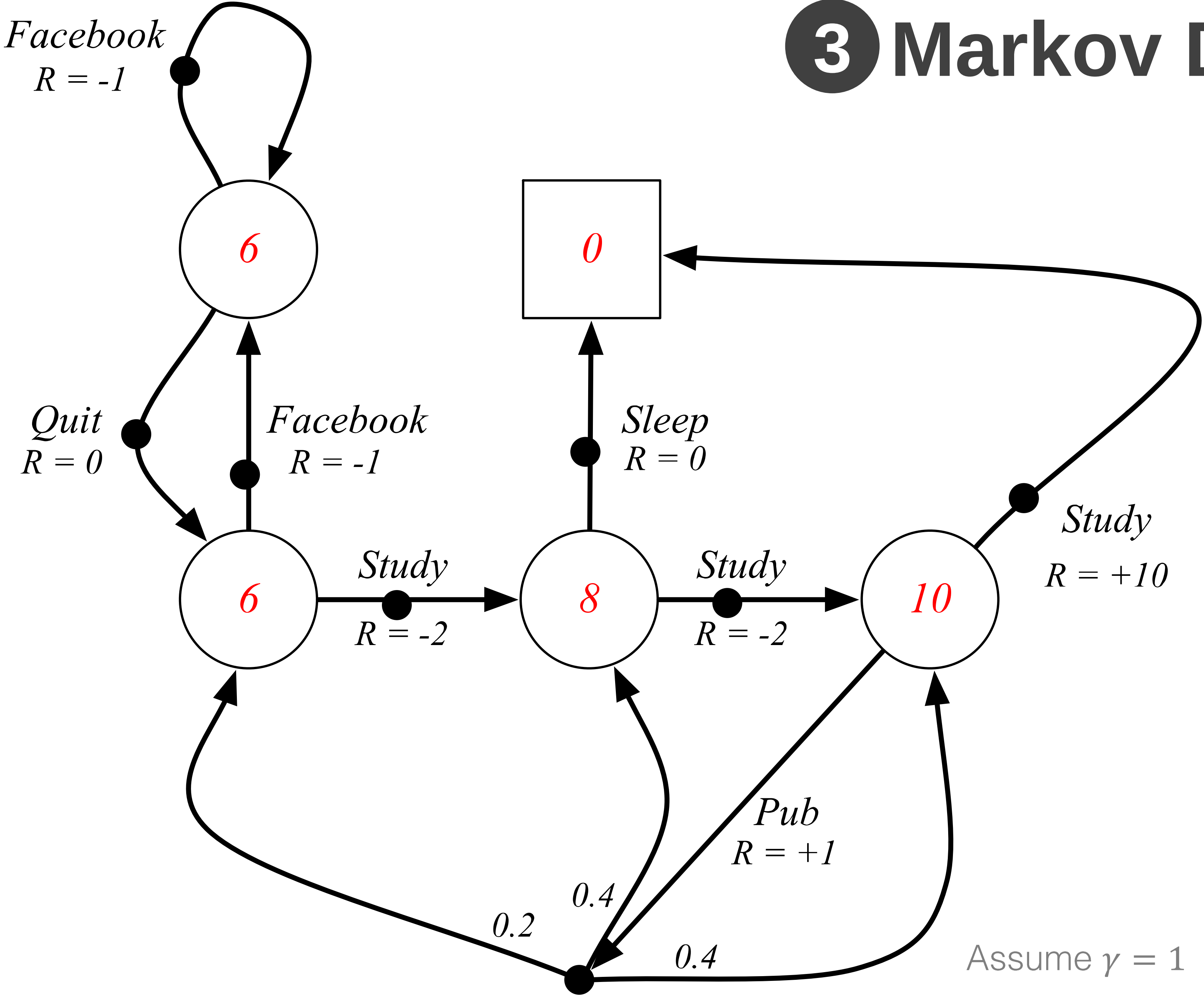
$$v(s) = E[R_s^a + \gamma v(s')|s]$$

$$-1.3 \approx 0.5 \times (-1 - 2.3) + 0.5 \times (-2 + 2.7)$$

Assume $\gamma = 1$

Example from David Silver, UCL, 2015

3 Markov Decision Process



Optimal **state-value** function, $v_*(s)$

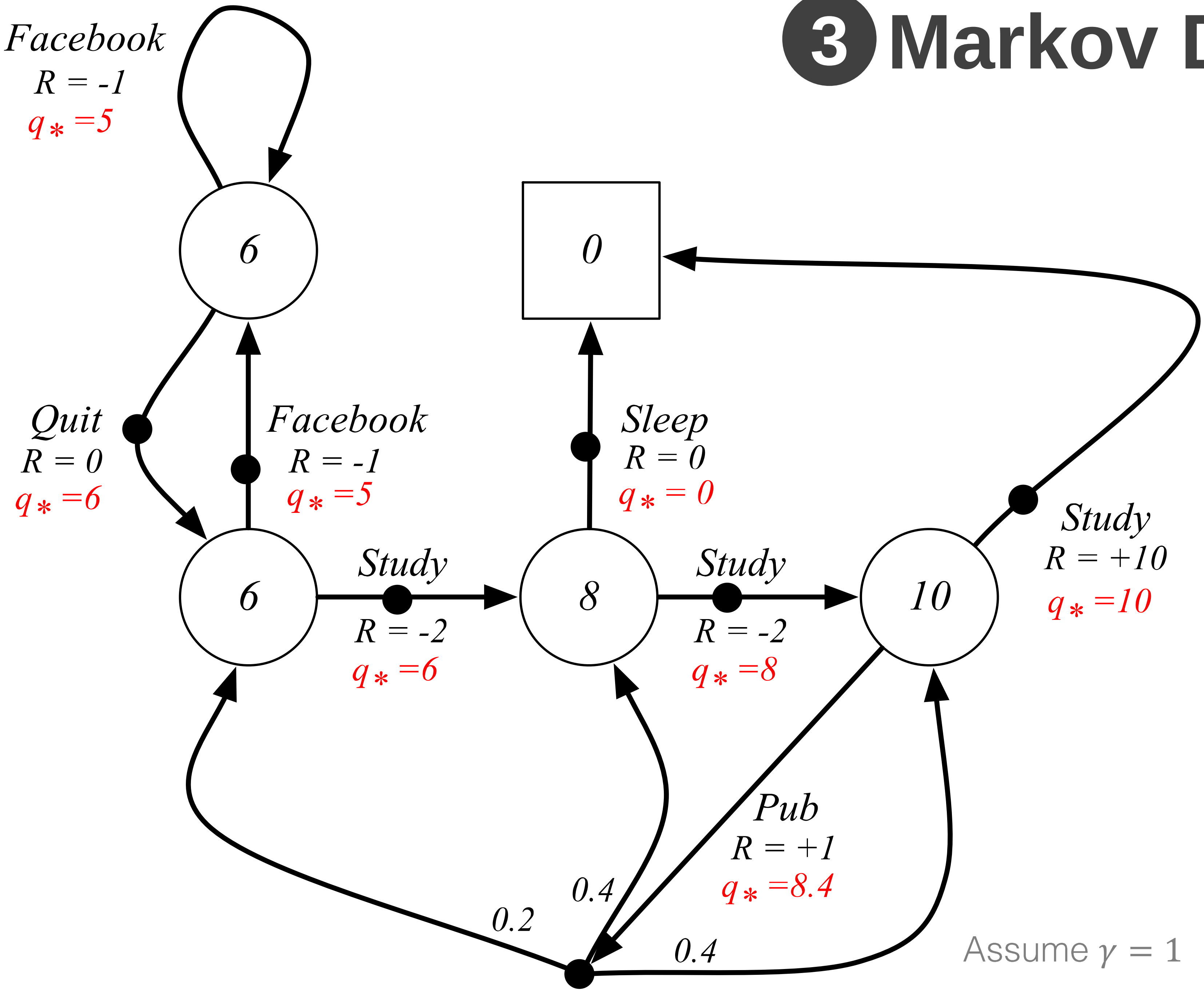
Maximum value function over all policies

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

Assume $\gamma = 1$

Example from David Silver, UCL, 2015

3 Markov Decision Process



Assume $\gamma = 1$

Optimal **state-value** function, $v_*(s)$

Maximum value function over all policies

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

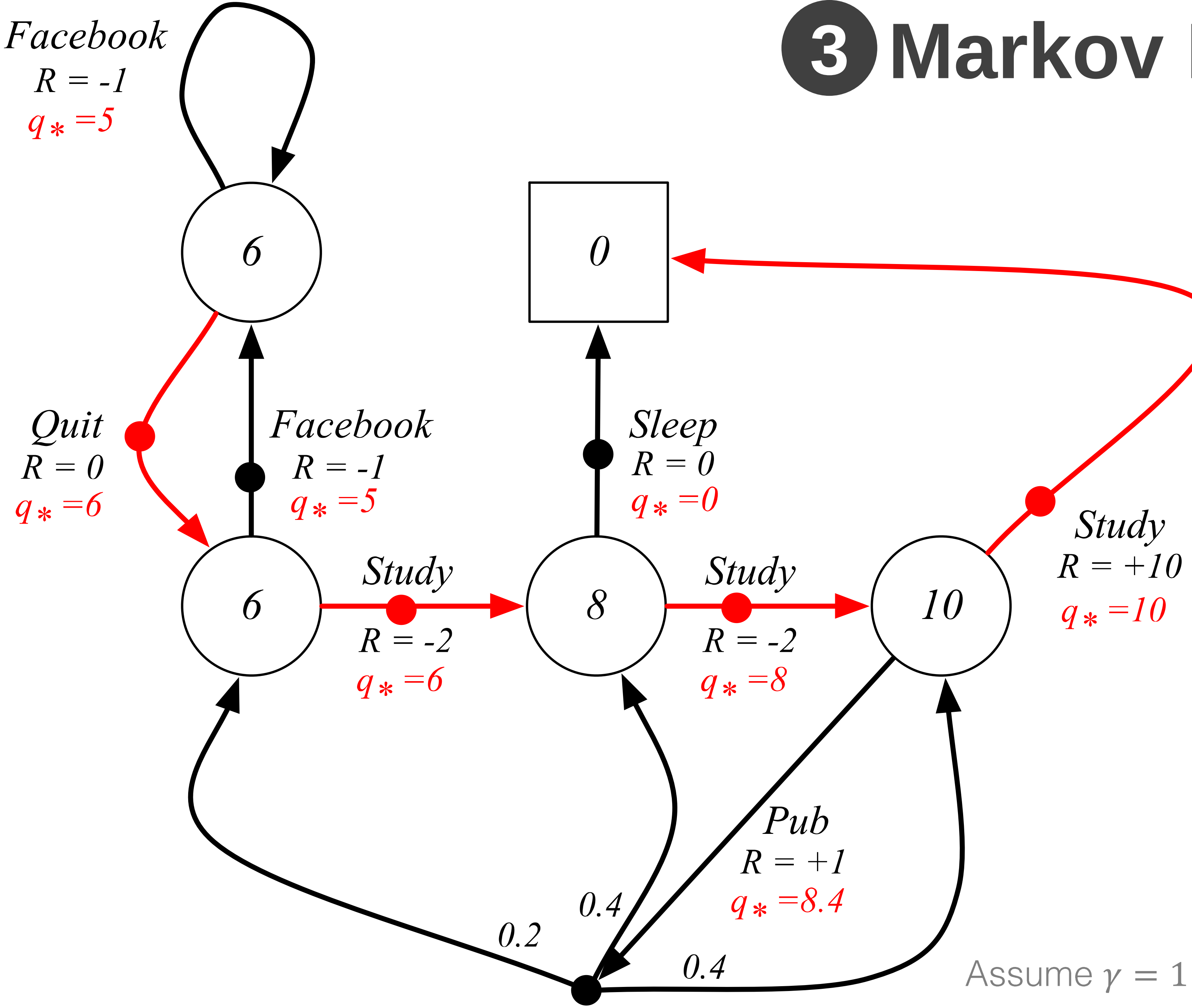
Optimal **action-value** function, $q_*(s,a)$

Maximum value function over all policies

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

Example from David Silver, UCL, 2015

3 Markov Decision Process



Optimal **policy**, $\pi_*(s)$

Which action to take at each moment

$$\pi_*(s) = \arg\max_a q_*(s, a)$$

Once we have the optimal value functions, we've "solved" the MDP!

Assume $\gamma = 1$

Example from David Silver, UCL, 2015

Learning strategy

	Model-based (planning)	Model-free
Knowledge of **Environment**		Reinforcement Learning
No knowledge Must learn from experience		**Monte Carlo Learning**
Perfect knowledge Known MDP	**Dynamic Programming** Policy iteration Value iteration	**Next class**: 1. How to **compute optimal policies** for known MDPs? 2. How to extend this to the case **without full knowledge** of MDPs