Introduction to Reinforcement Learning

2025. 1st semester

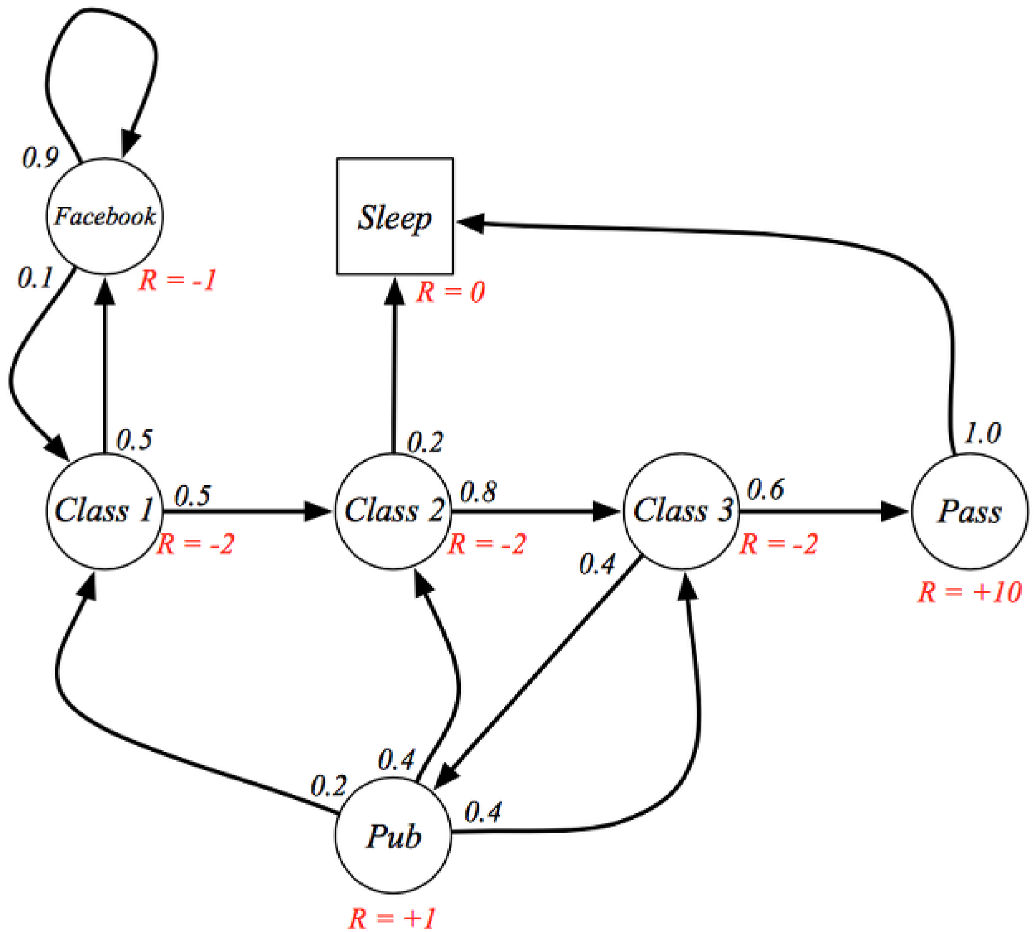
Markov Reward Process (MRP)

- *A Markov Reward Process is a Markov chain with values*
- *Definition*
  - *A MRP is a tuple* $< S, P, R, \gamma >$
    - *S is a finite set of states*
    - *P is a state transition matrix,*

      $$P_{ss'} = \Pr[S_{t+1} = s' \mid S_t = s]$$
    - *R is a reward function,* $\boxed{R_s = \mathrm{E}[R_{t+1} \mid S_t = s]}$
    - γ *is a discount factor,* $\gamma \in [0,1]$
  - *Episode*
    - *A sequence of states until the agent-environment interaction breaks*
    - *Each episode ends in a special state called the 'terminal state'*
    - *Sometimes called "trials"*

MRP

- ***Example: Student MRP***



Return

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

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

  - The discount factor $\gamma \in [0,1]$ has influence on the present value of future reward
  - The value of receiving reward R after k+1 time-steps is $\gamma^k R$
  - This values immediate reward above delayed reward
  - γ close to 0 leads to "myotic" evaluation
  - γ close to 1 leads to "far-sighted" evaluation

Why

- *Most Markov rewards 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 behavior 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*

State 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*

$$\boxed{v(s) = \mathrm{E}\left[G_t \mid S_t = s\right]}$$

  - *Remember that*

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

  - *and*

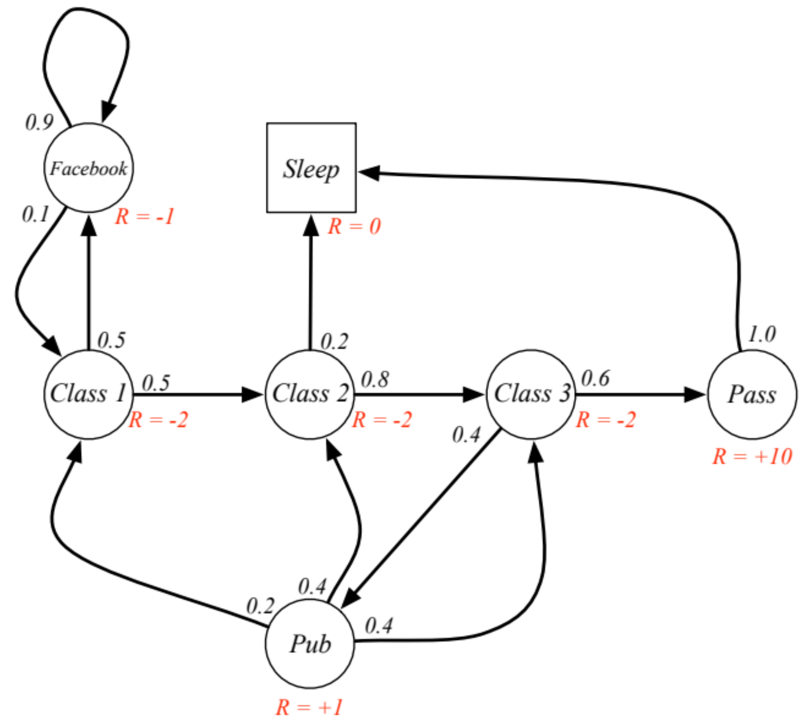
$$\boxed{R_s = \mathrm{E}[R_{t+1} \mid S_t = s]}$$

Student MRP Returns

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

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

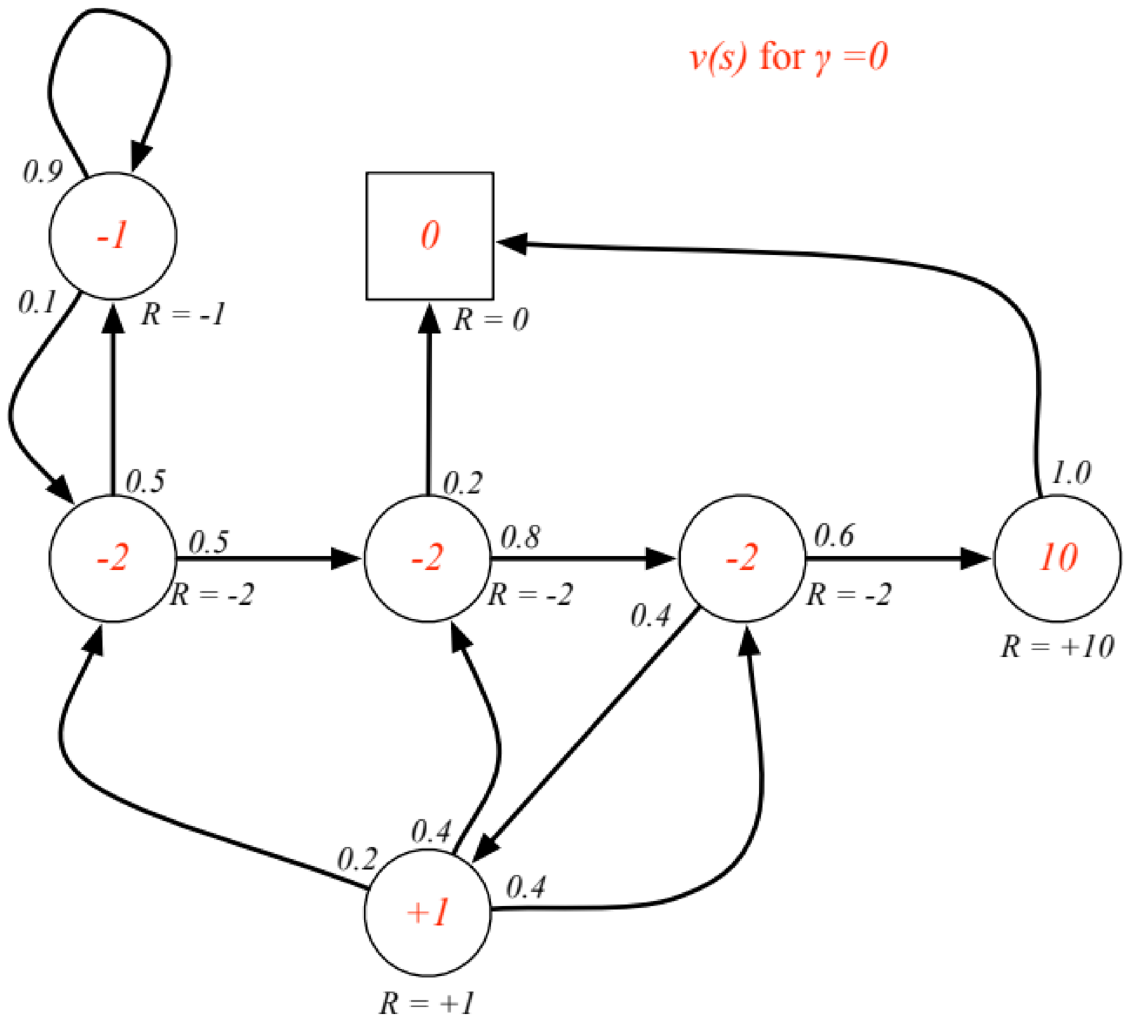
  - ***State value = Average return values of multiple episodes***



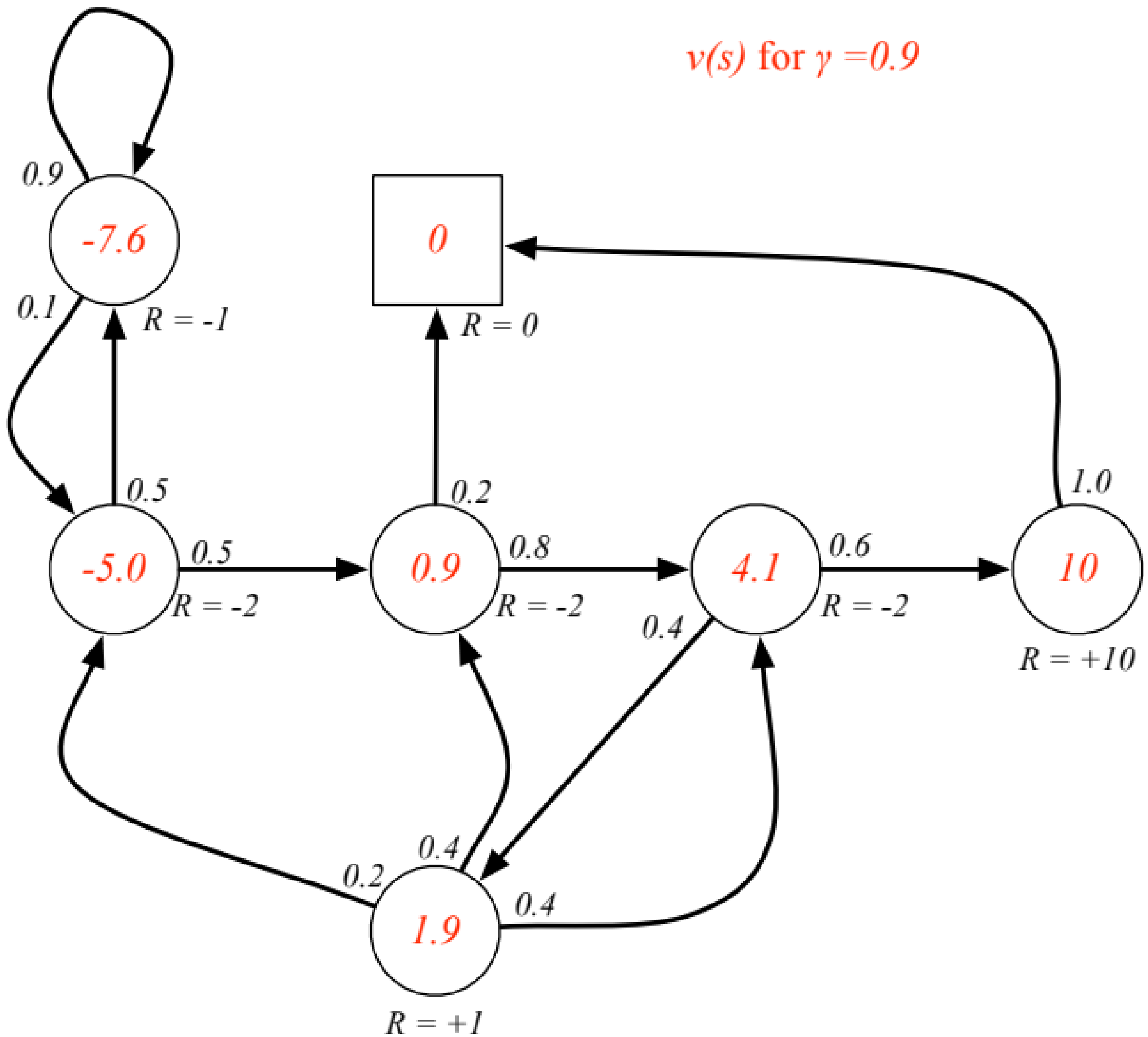
C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8} \quad = \quad -2.25$
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \quad = \quad -3.125$
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4}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

State Value function

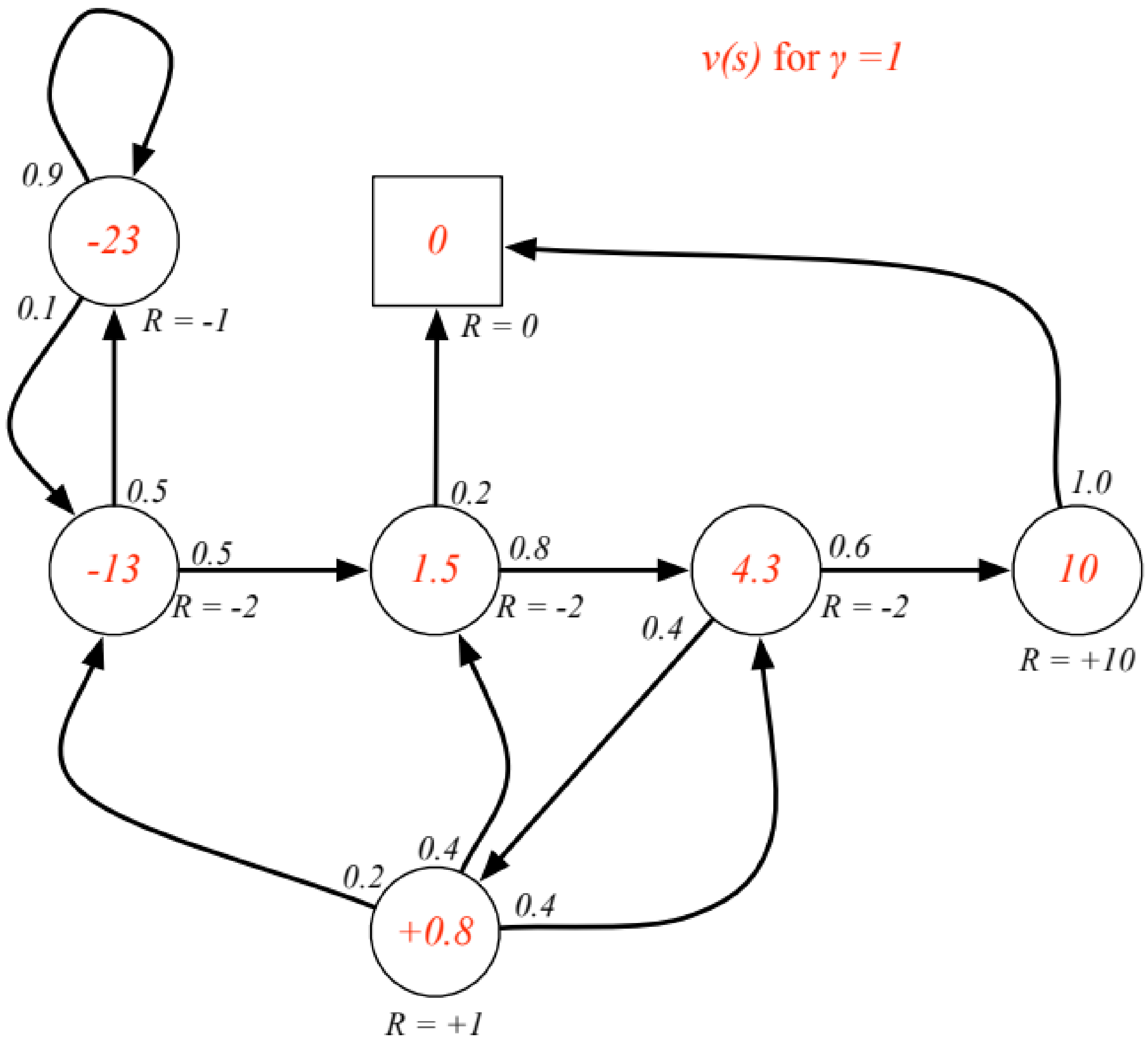
- *Note that v(s) is expected value*
- *Ex)*



State Value function



State Value function



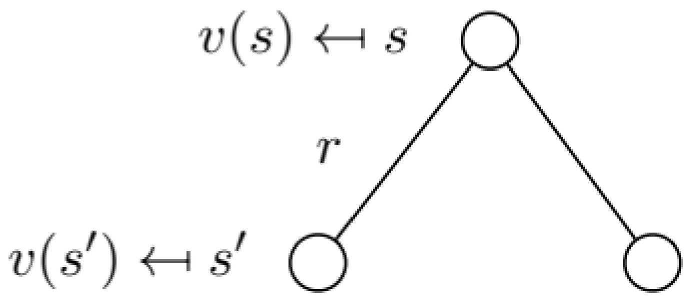
Bellman Equation for MRPs

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

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

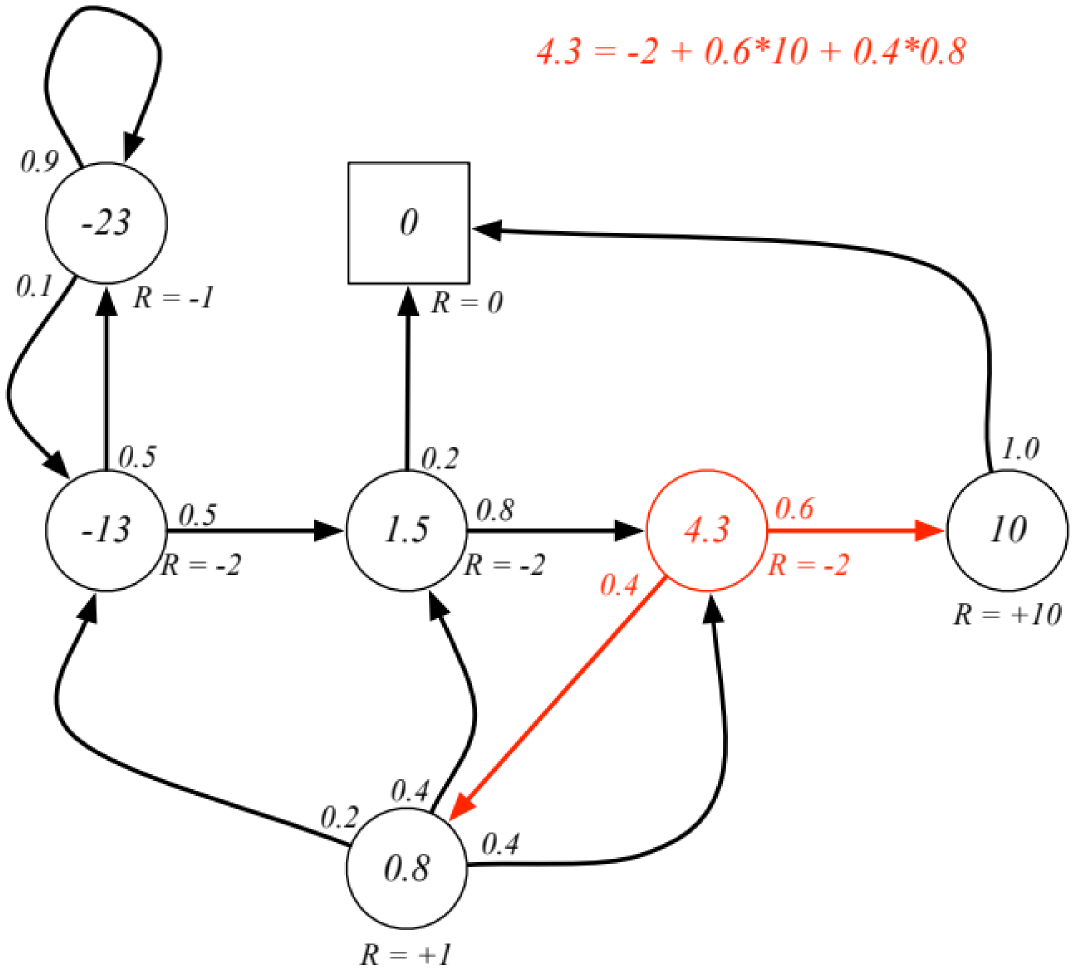
Bellman Equation for MRPs

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



$$v(s) = R_s + \gamma \sum_{s' \in S} p_{ss'} v(s')$$

State Value function



Bellman equation in Matrix form

- *The Bellman equation can be expressed concisely using matrices,*

$$v = R + \gamma P v$$

  - *Where v is a column vector with one entry per state*

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_n \end{bmatrix} + \gamma \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

Solving the Bellman equation

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

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

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