02. 마르코프 결정 프로세스

김호철

Contents

---

1 마르코프 프로세스

1.1 소개

- 마르코프 결정 프로세스는 강화학습에서 환경을 설명한다.
- 환경은 완전 관측 가능(fully observable)한 상황이다.
- 현재의 상태가 프로세스를 완전히 표현한다.
- 거의 모든 강화학습문제는 마르코프 결정 프로세스(MDP)이다.

1.2 마르코프 속성

- 미래는 현재 시점에 과거와는 독립적이다.
- 상태 S_t는 다음 경우에만 마르코프이다.

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, ..., S_t] \tag{1}$$

- 상태는 히스토리로부터 관련된 모든 정보를 가져온다.
- 상태를 알면 히스토리는 버려진다.
- 그러므로 상태는 미래에 대하여 충분한 통계량이 된다.

1.3 상태 전이 행렬

- 마르코프 상태 s에서 다음 상태 s'로의 상태 전이 확률을 다음과 같이 정의

$$P = \begin{array}{c} \\ \text{from(s)} \end{array} \begin{array}{c} \begin{array}{ccc} & \text{to(s')} & \end{array} \\ \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \end{array}$$

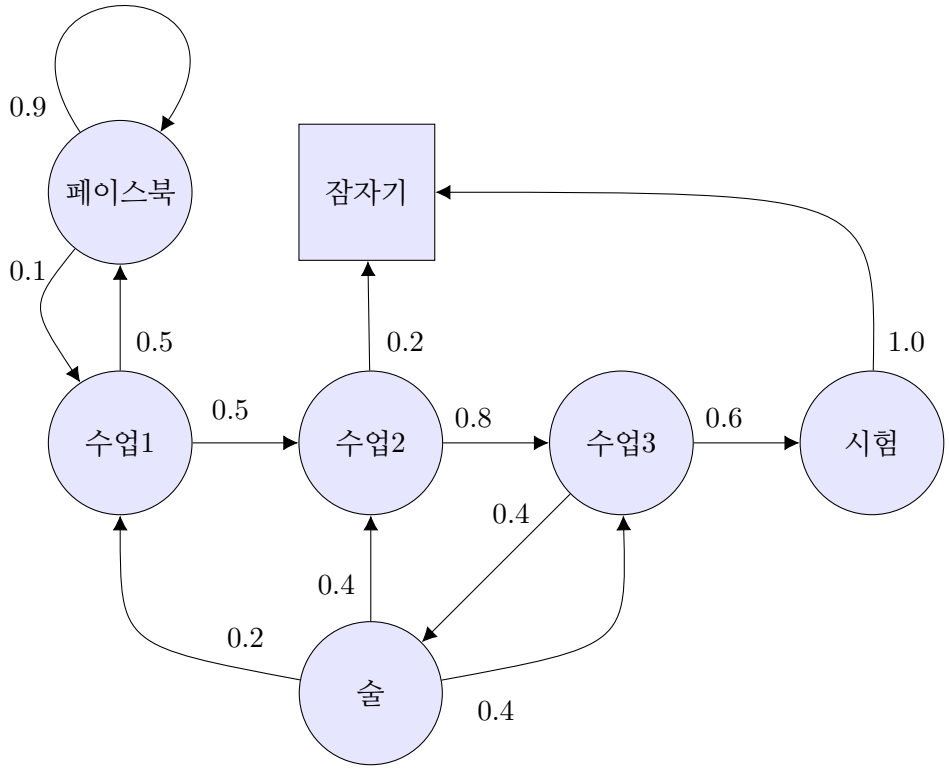
- 행렬의 각 열의 합은 1이 된다.

1.4 마르코프 체인

- 마르코프 프로세스는 무기억(memoryless-어떤 경로를 거쳐서 왔건) 랜덤 프로세스(샘플링 가능)
- 즉, 마르코프 속성을 갖는 랜덤 상태 S_1, S_2, ... 의 순서
- 마르코프 프로세스(혹은 마르코프 체인)은 튜플 $< S, P >$ 이다.
- S는 상태들의 (유한) 집합
- P는 상태 전이 확률 행렬이다.

$$P_{ss'} = \mathbb{P}[S_{t+1} = s'|S_t = s] \tag{2}$$

- 학생의 마르코프 체인 예제



- S_1을 수업1에서 시작하는 학생 마르코프 체인 샘플링 에피소드
  - 수업1 → 수업2→ 수업3 → 시험→ 잠자기
  - 수업1 → 페이스북 → 페이스북 → 수업1 → 수업2 → 잠자기
  - 수업1 → 수업2 → 수업3 → 술 → 수업2 → 수업3 → 시험 → 잠자기
  - 수업1 → 페이스북 → 페이스북 → 수업1 → 수업2 → 수업3 → 술 → 수업1 → 페이스북 → 페이스북 → 페이스북 → 수업1 → 수업2 → 수업3 → 술 → 수업2 → 잠자기

- 학생 마르코프 체인 전이 행렬

$$P = \begin{array}{r} \\ \text{수업1} \\ \text{수업2} \\ \text{수업3} \\ \text{시험} \\ \text{술} \\ \text{페이스북} \\ \text{잠자기} \end{array} \begin{array}{c} \begin{array}{ccccccc} \text{수업1} & \text{수업2} & \text{수업3} & \text{시험} & \text{술} & \text{페이스북} & \text{잠자기} \end{array} \\ \begin{bmatrix} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ & & & & & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{bmatrix} \end{array}$$

2 마르코프 보상 프로세스(Markov Reward Process)

2.1 마르코프 보상 프로세스

- 마르코프 보상 프로세스는 value(미래 보상들의 합)들을 가지는 마르코프 체인이다.

정의

마르코프 보상 프로세스는 튜플 $< S, P, R, \gamma >$ 이다.

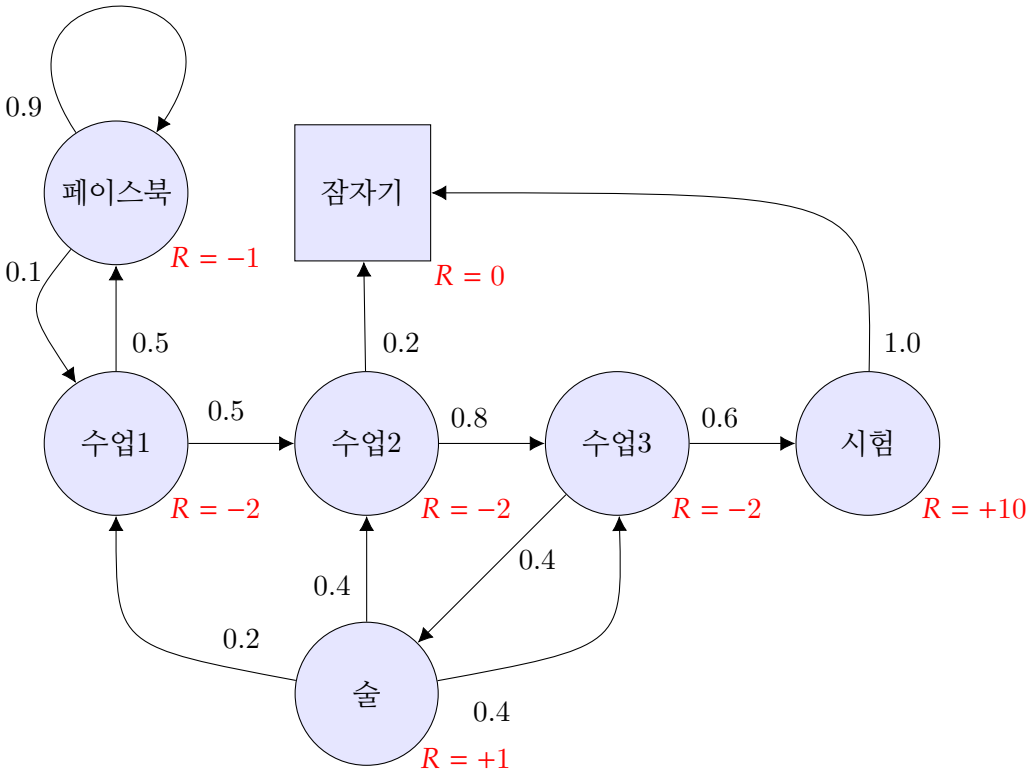
S는 상태를 나타내는 유한 집합이다.

P는 상태 전이 확률 행렬이다. $P_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$

R은 보상 함수이다. $R_s = \mathbb{E}[R_{t+1} | S_t = s]$

γ는 할인율(discount factor)이다. $\gamma \in [0, 1]$

- 학생의 마르코프 보상 프로세스 예제



2.2 리턴(Return)

정의

리턴 G_t는 어떤 에피소드에서 t 시점에 할인된 보상들의 합이다.

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

- γ가 0에 가까울수록 근시적평가, 1에 가까울수록 원시적 평가
- 할인(Discount)을 하는 이유

1. 수학적으로 편리해서
2. 순환에 의한 무한 수익 방지
3. 미래에 대한 불확실성은 완전히 표현 되지 않을 수 있다
4. 보상이 재정적인 경우 즉각적인 보상이 지연된 보상보다 더 많은 이자를 얻을 수 있다
5. 동물/인간 행동은 즉각적인 보상을 선호함
6. 모든 시퀀스가 종료 되는 경우 가끔 γ=1 이 가능한 경우가 있음

2.3 가치 함수(Value Function)

- 가치 함수 $v(s)$는 리턴의 기대값으로, 상태 s에서 장기(long-term) 가치를 제공

정의

MRP의 상태 가치 함수 $v(s)$는 상태 s에서 시작되는 기대 리턴이다.

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

- 학생 MRP 리턴의 예제
  - 학생 MRP에서 리턴을 샘플링하고, γ는 $\frac{1}{2}$이고 "수업1"에서 시작

$$G_1 = R_2 + \gamma R_3 + ... + \gamma^{T-2} R_T \tag{3}$$

에피소드1 : 수업1 → 수업2 → 수업3 → 시험 → 잠자기

$$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8} = -2.25$$

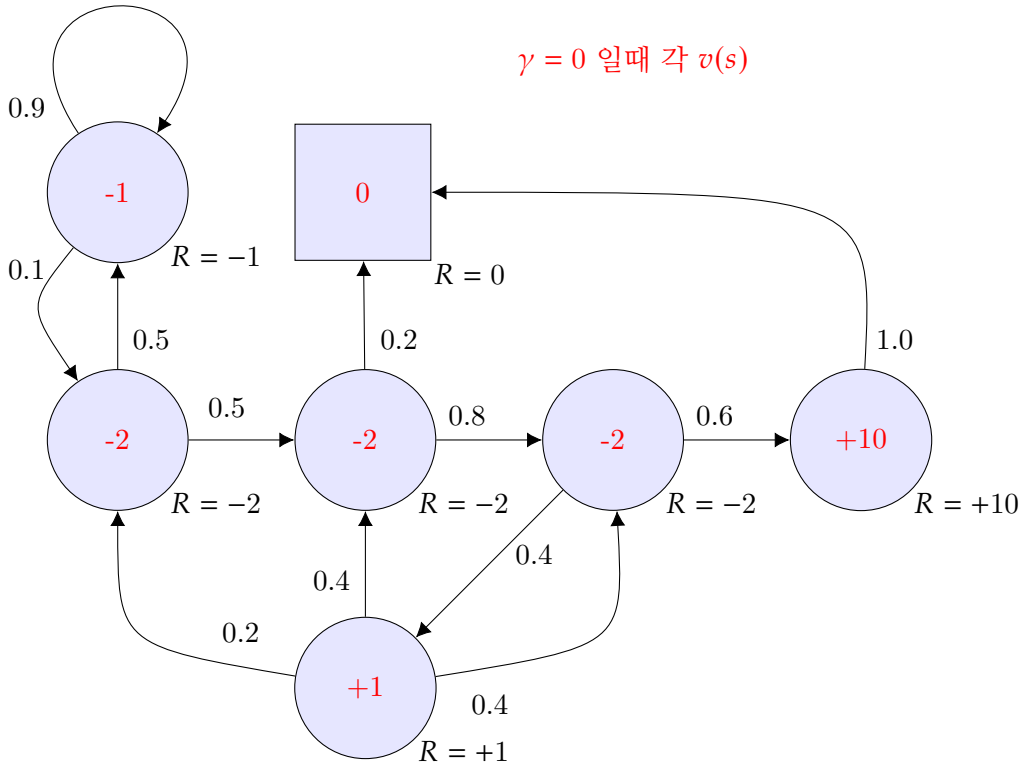
에피소드2 : 수업1 → 페이스북 → 페이스북 → 수업1 → 수업2 → 잠자기

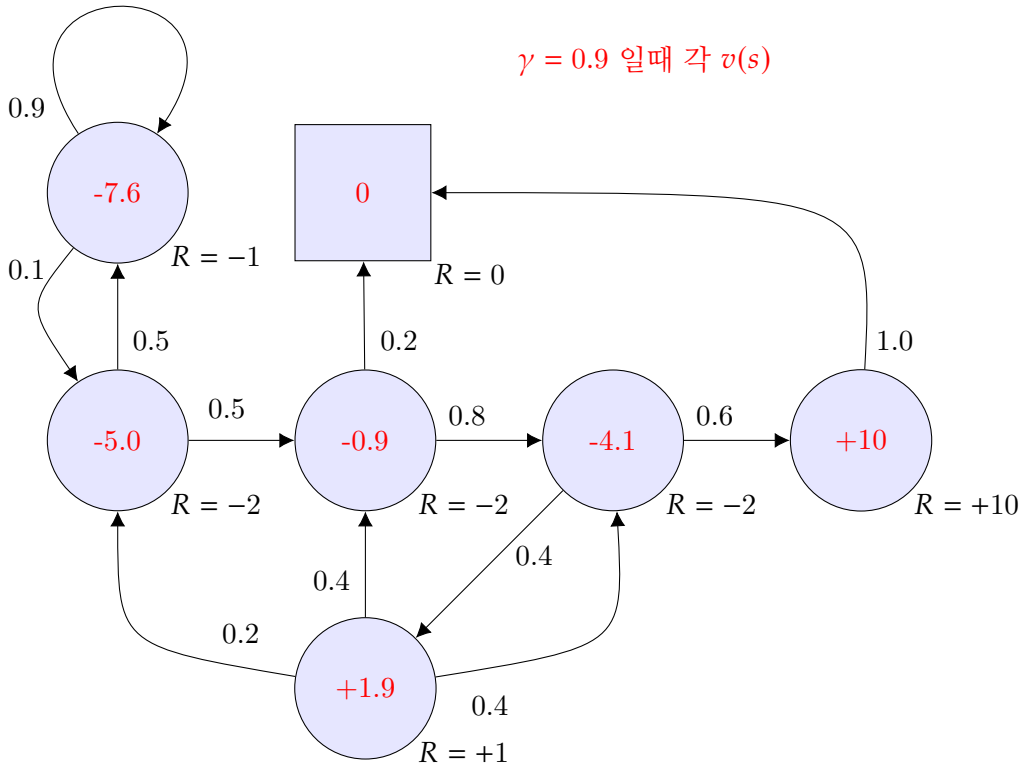
$$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} = -3.125$$

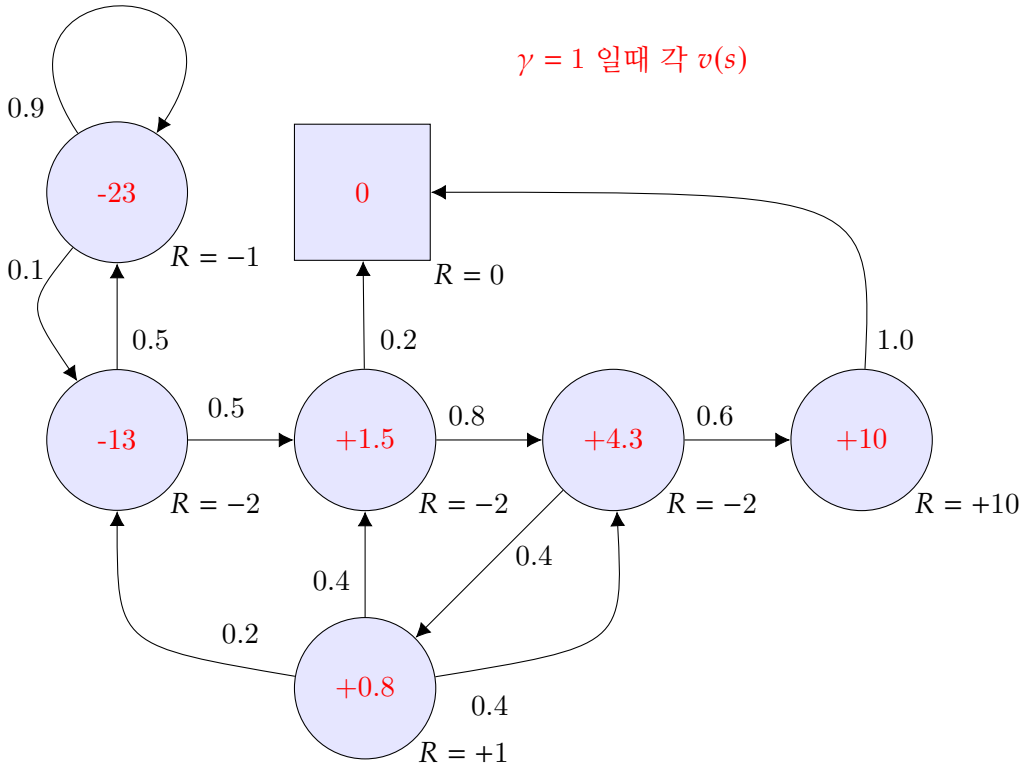
에피소드3 : 수업1 → 수업2 → 수업3 → 술 → 수업2 → 수업3 → 시험 → 잠자기

$$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} - 2 * \frac{1}{32} + 10 * \frac{1}{64} = -3.41 \tag{4}$$

  - 학생 MRP에서 상태-가치(State-Value) 함수


γ = 0 일때 각 v(s)
0.9
-1
R = −1
0
R = 0
0.1
0.5
0.2
1.0
-2
0.5
-2
0.8
-2
0.6
+10
R = −2
R = −2
R = −2
R = +10
0.4
0.4
0.2
+1
0.4
R = +1


γ = 0.9 일때 각 v(s)
0.9
-7.6
R = −1
0
R = 0
0.1
0.5
0.2
1.0
-5.0
0.5
-0.9
0.8
-4.1
0.6
+10
R = −2
R = −2
R = −2
R = +10
0.4
0.4
0.2
+1.9
0.4
R = +1



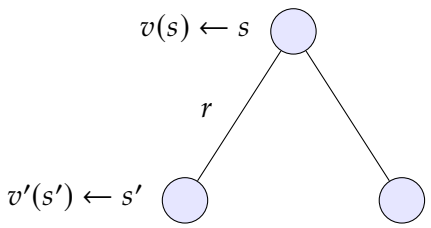
2.4 벨만 방정식

- 가치 함수는 두개의 부분으로 나뉘어 진다.
  - 즉각적인 보상 R_{t+1}
  - 다음 상태들의 할인된 가치 $\gamma v(S_{t+1})$

$$\begin{aligned} v(S) &= \mathbb{E}[G_t|S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots |S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots)|S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1})|S_t = s] \end{aligned} \tag{5}$$

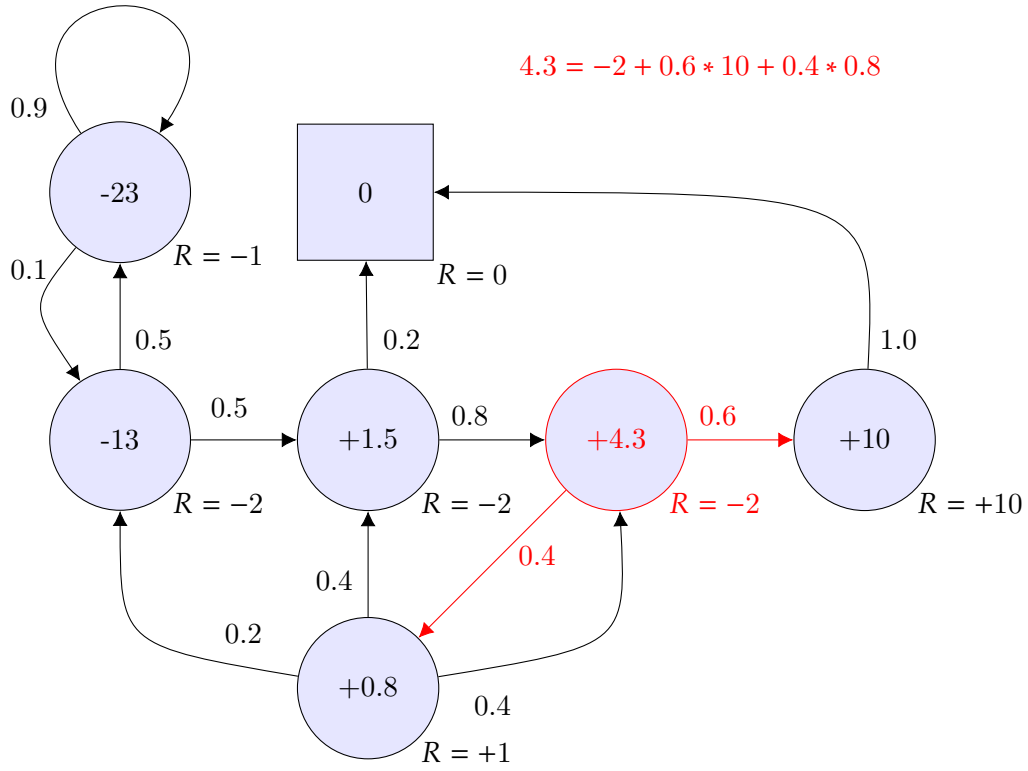
- MRP에 대한 벨만 방정식

$$v(S) = \mathbb{E}[R_{t+1} + \gamma v(S_{t+1})|S_t = s] \tag{6}$$



$$v(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} v(s') \tag{7}$$

- 학생 MRP에 대한 상태 가치(State Value) 함수



- 벨만 방정식의 행렬식

$$v = R + \gamma P v \tag{8}$$

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

- 벨만 방정식은 선형 방정식으로 다음과 같이 직접 풀 수 있다.

$$\begin{aligned} v &= R + \gamma P v \\ (I - \gamma P) v &= R \\ v &= (I - \gamma P)^{-1} R \end{aligned} \tag{9}$$

- 계산 복잡도는 n개 상태에서 $O(n^3)$
- 작은 MRP는 손으로 직접 푸는 것이 가능
- 큰 MRP는 많은 반복적(Iterative) 방법들(DP, MC, TD 학습)들을 사용

3 마르코프 결정 프로세스

3.1 마르코프 결정 프로세스(Markov Decision Process)

- MDP는 의사 결정(Decision)이 포함된 MRP 이다.
- 모든 상태들이 마르코프인 환경이다.

정의

마르코프 결정 프로세스는 튜플$< S, A, P, R, \gamma >$이다.

- S는 상태들의 유한 집합니다.

- A는 액션들의 유한 집합이다.

- P는 상태 전이 확률이다.

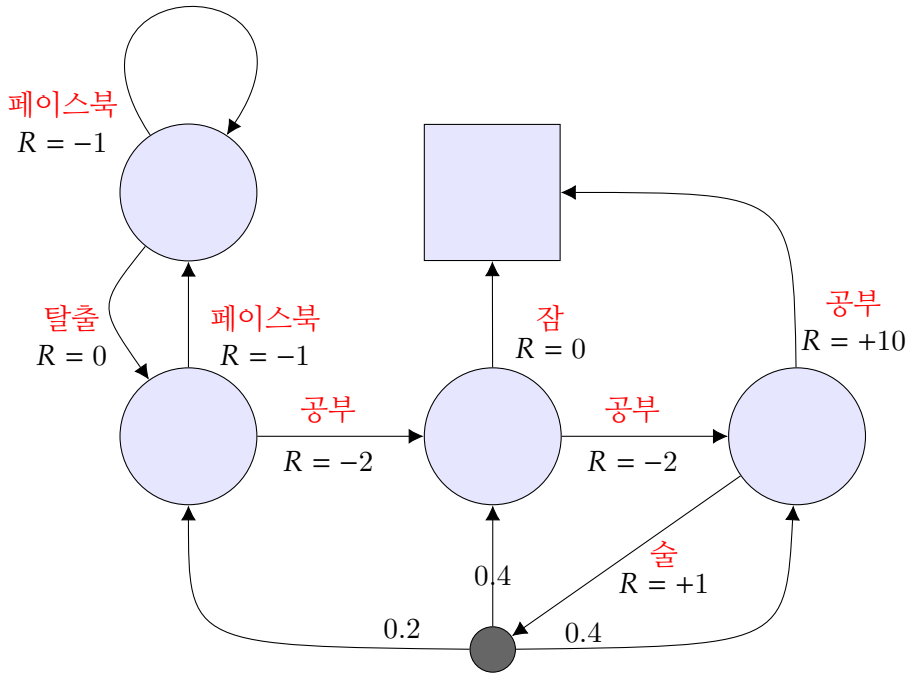
$$P^{a}_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

- R은 보상 함수이다.

$$R^{a}_{s} = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

- γ는 할인율이다.$\gamma \in [0, 1]$

- 학생의 MDP



3.2 정책(Policies)

정의

정책 π 는 주어진 각 상태들에서 발생할 수 있는 액션에 대한 분포이다.

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

- 정책은 에이전트의 동작을 완전히 정의한다.
- MDP 정책은 히스토리가 아니라, 현재 상태에 종속적이다.
- 즉, 정책은 고정적이다(시간에 독립적).

3.3 가치 함수(Value Function)

정의

MDP의 상태-가치(State-Value) 함수 $v_\pi(s)$ 는 상태 s에서 정책 π를 따랐을때 예상되는 리턴이다.

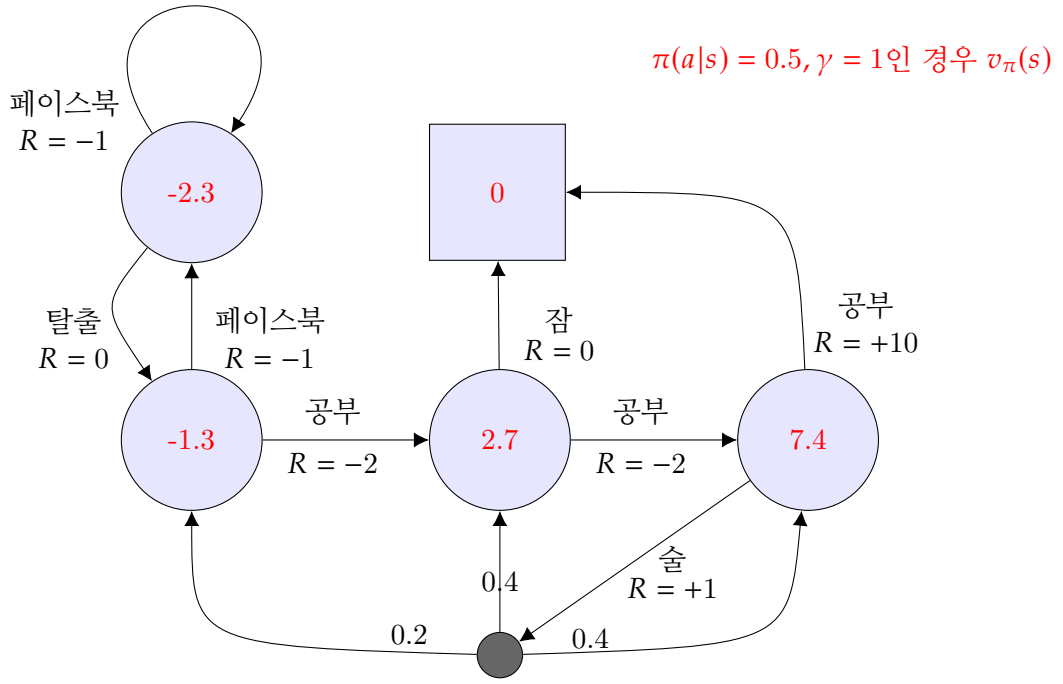
$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$$

정의

액션-가치(Action-Value) 함수 $q_\pi(s,a)$ 는, 상태 s에서 액션 a를 취하고 정책 π를 따랐을때 예상되는 리턴이다.

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

- 학생 MDP 상태 가치 함수



3.4 벨만 기대 방정식(Bellman Expectation Equation)

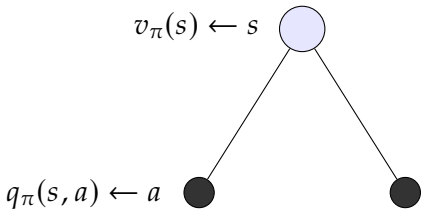
- 상태 가치 함수는 즉각적인 보상과 다음 상태의 할인된 가치의 합으로 다시 생각해 볼 수 있다.

$$v_\pi(s) = \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \tag{10}$$

- 액션 가치 함수도 유사하게 생각할 수 있다.

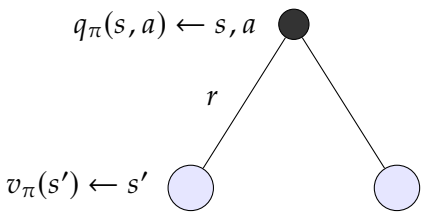
$$q_\pi(s,a) = \mathbb{E}[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{11}$$

- V^π 벨만 기대 방정식



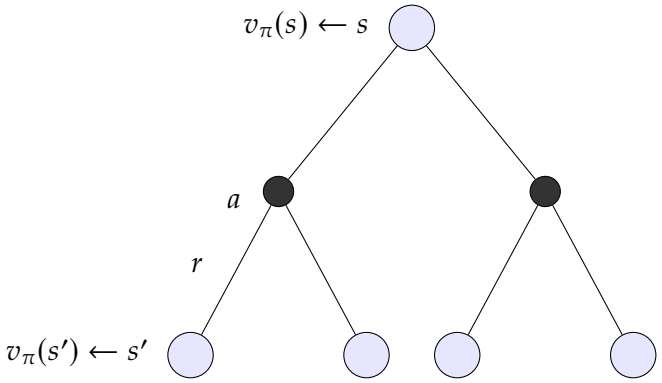
$$v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s,a) \tag{12}$$

- Q^π 벨만 기대 방정식



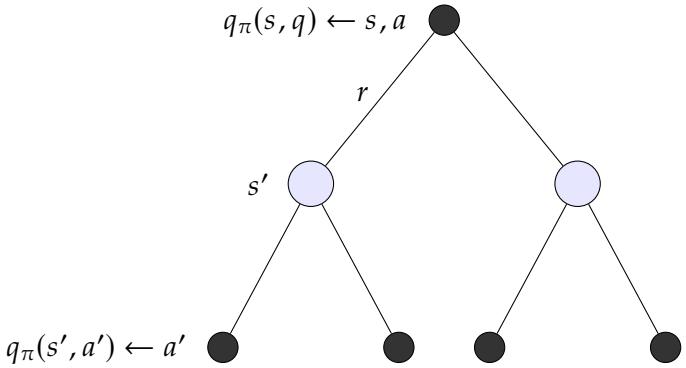
$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \tag{13}$$

- v_π 벨만 기대 방정식 2



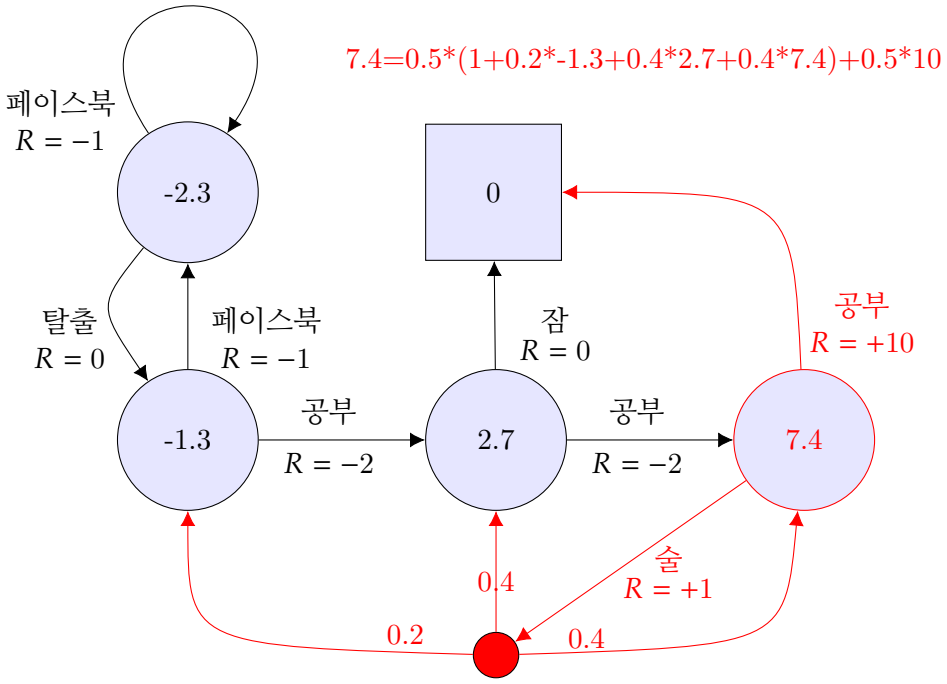
$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \Big(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \Big) \tag{14}$$

- q_π 벨만 기대 방정식 2



$$q_\pi(s,a) = R_s^a + \gamma \sum_{a \in A} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s',a') \tag{15}$$

- 학생 MDP 상태 가치 함수 예



3.5 최적 가치 함수(Optimal Value Function)

정의

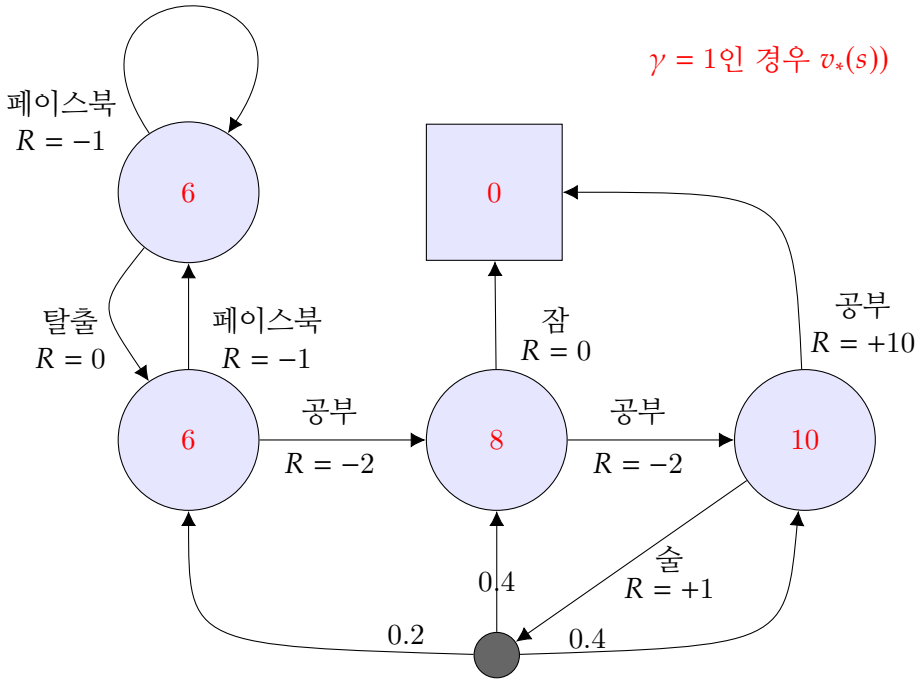
최적 상태 가치 함수 $v_*(s)$는 모든 정책에 대하여 최대 가치 함수이다.

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

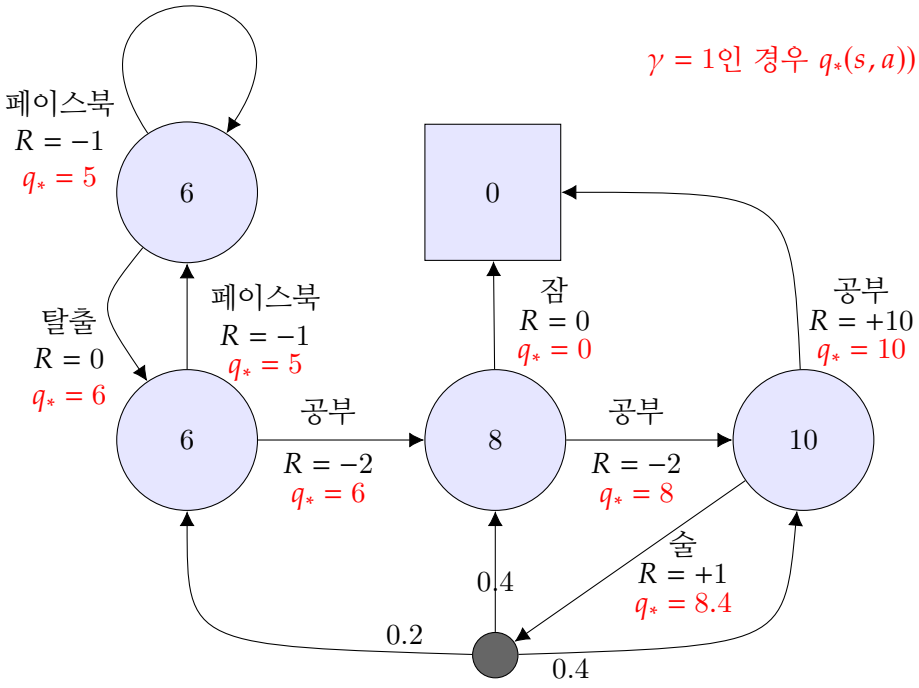
최적 액션 가치 함수 $q_*(s,a)$는 모든 정책에 대하여 최대 액션 가치 함수이다.

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

- 최적 가치 함수는 MDP에서 가능한 최상의 성능을 나타낸다.
- 최적 가치 함수를 알면 MDP문제는 해결된 것이다.
- 학생 MDP 최적 가치 함수



- 학생 MDP 최적 액션 가치 함수



- 최적 정책(Optimal Policy)
  - 정책에 대해, 각각의 순위를 정의하면,

$$\pi \geq \pi' \; if \; v_\pi(s) \geq v_{\pi'}(s), \forall s \tag{16}$$

정리(Theorem)

어떤 마르코프 결정 프로세스에서,

- 모든 다른 정책들 보다 더 좋거나 동일한 최적 정책 π_*는 존재한다.

  $\pi_* \geq \pi, \forall \pi$

- 모든 최적 정책들은 최적 가치 함수를 달성한다. $v_{\pi_*(s)} = v_*(s)$

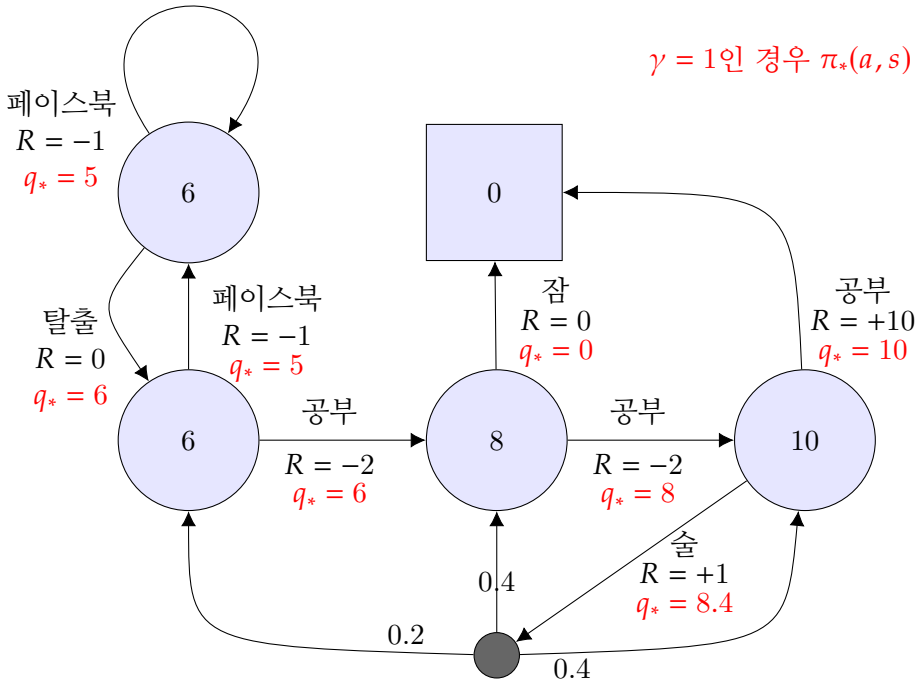
- 모든 최적 정책들은 최적 액션 가치 함수를 달성한다. $q_{\pi_*(s,a)} = q_*(s,a)$

- 최적 정책 찾기
  - 최적 정책은 $q_*(s,a)$를 최대화하여 찾을 수 있다.

$$\pi_*(a|s) = \begin{cases} 1, & \text{if } a = \underset{a \in A}{\text{argmax}}\ q_*(s,a) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

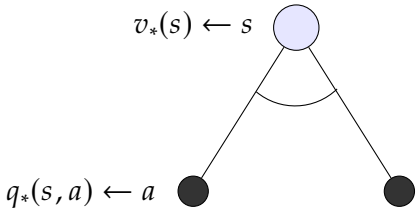
  - 모든 MDP에 대해 항상 결정론적 최적 정책이 있다.
  - $q_*(s,a)$를 알면 즉시 최적의 정책을 알게 된다.

- 학생 MDP 최적 정책(Optimal Policy)



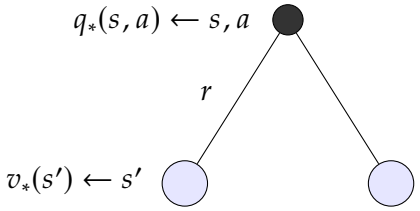
3.6 벨만 최적 방정식

- v_* 벨만 최적 방정식 : 최적 가치 함수는 벨만 최적 방정식과 재귀적으로 연관되어 있다.



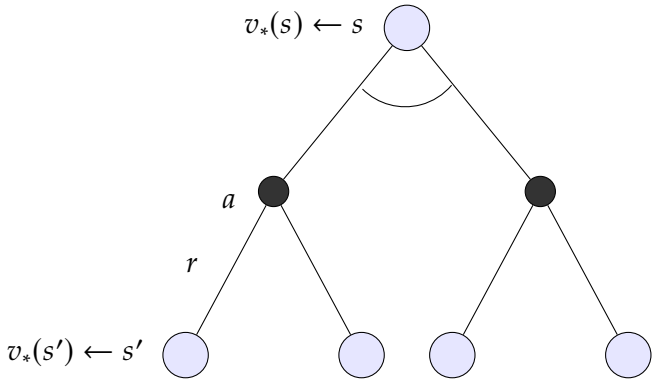
$$v_*(s) = \max_a \ q_*(s,a) \tag{18}$$

- Q_* 벨만 최적 방정식



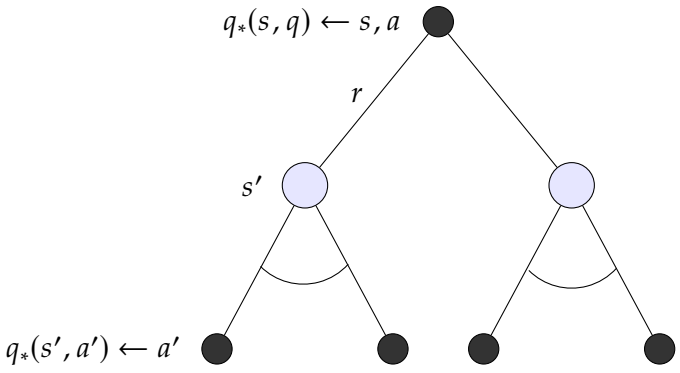
$$q_*(s,a) = R_s^a + \gamma \max_{s' \in S} \ P_{ss'}^a \ v_*(s') \tag{19}$$

- V_* 벨만 최적 방정식 2



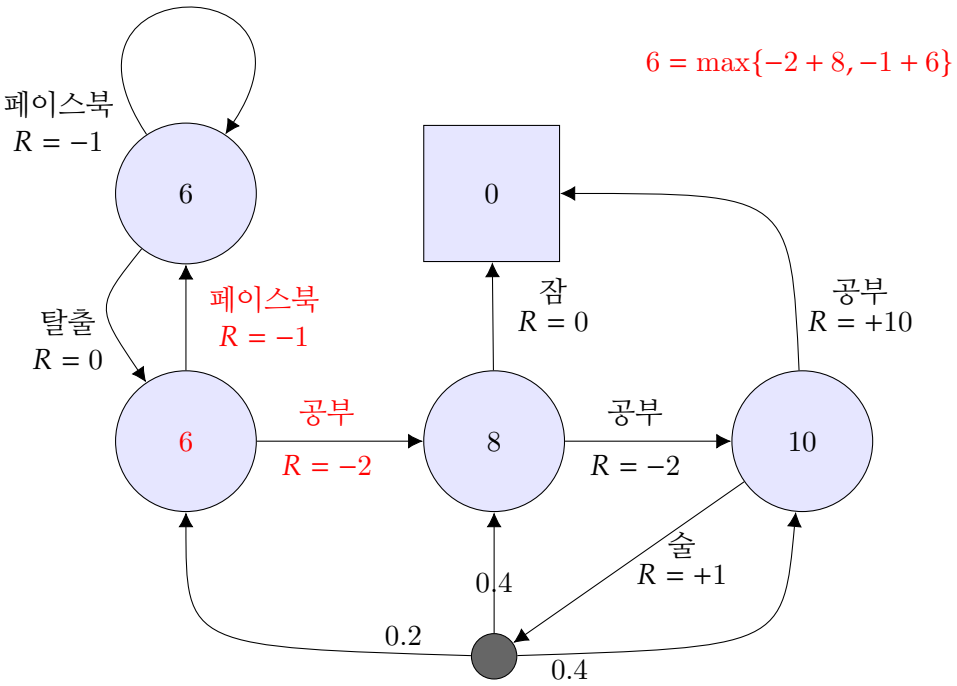
$$v_*(s) = \max_a R_s^a \ + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s') \tag{20}$$

- Q_* 벨만 최적 방정식 2



$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} q_*(s',a') \tag{21}$$

- 학생 MDP 벨만 최적 방정식



- 벨만 최적 방정식 풀기
  - 벨만 최적 방정식은 비선형적이다.
  - closed form solution 없음
  - 많은 반복적 솔루션 방법
  - Value Iteration, Policy Iteration, Q-learning, Sarsa