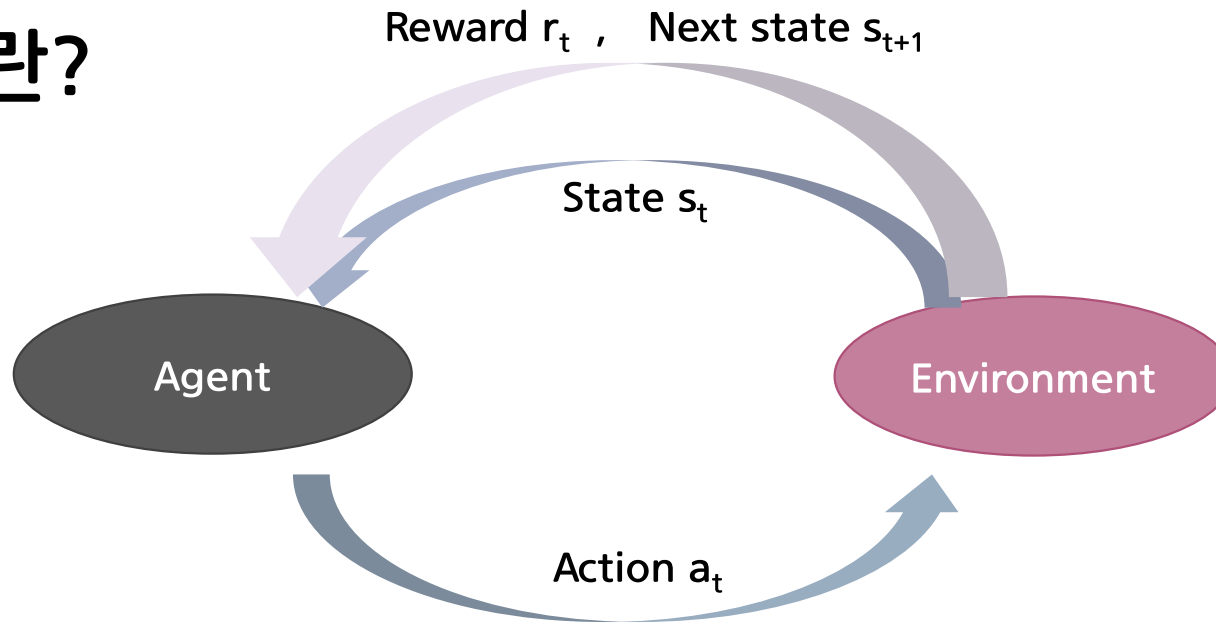


강화학습과 Q-Learning

201810808 정민지

강화학습이란?



Agent: 주변 상태에 따라서 어떤 action을 취할지 판단을 내림

Environment: agent에게 state와 reward를 부여함

Reward: 주어진 state에서 취한 action으로부터 할당되는 보상

〔 Goal: Agent의 보상을 최대화 할 수 있는 방법을 학습 〕

마코프 결정 과정(MDP, Markov Decision Process)

강화학습 문제를 풀기 위하여 차용한 수학적 모델

마코프 성질: 현재 상태에서의 다음 상태로의 상태 전이는 오로지 현재 상태에만 영향을 받으며, 이전의 어떠한 상태에도 영향을 받지 않는다.

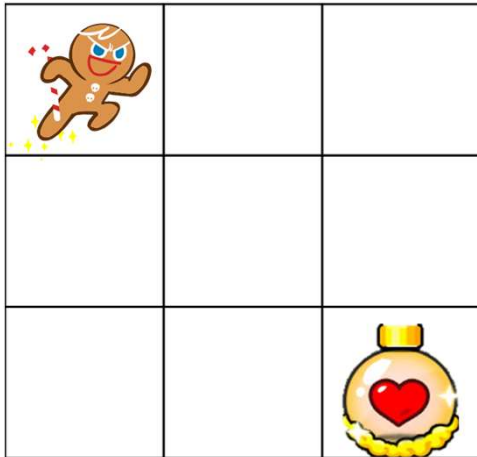
Ex) 동전 던지기, 부루마블

MDP formalism

- 상태^{state} S : 에이전트가 관찰할 수 있는 환경의 상태 s_i 들의 집합
- 행동^{action} A : 에이전트가 환경에 취할 수 있는 모든 행동 a_i 들의 집합
- 전이 확률^{transition probability} P : 상태 s 에서 행동 a 를 수행했을 때 상태 s' 로 옮겨 갈 확률
- 보상^{reward} R : 상태 s 에서 행동 a 를 수행하고 상태 s' 로 옮겨가며 환경에서 받는 보상
- 감가율^{discount rate} γ : 미래에 받을 보상에 대한 신뢰도. 0에서 1 사이의 수

MDP Example: grid-world

- 쿠키가 물약을 먹을 수 있게 도와주자

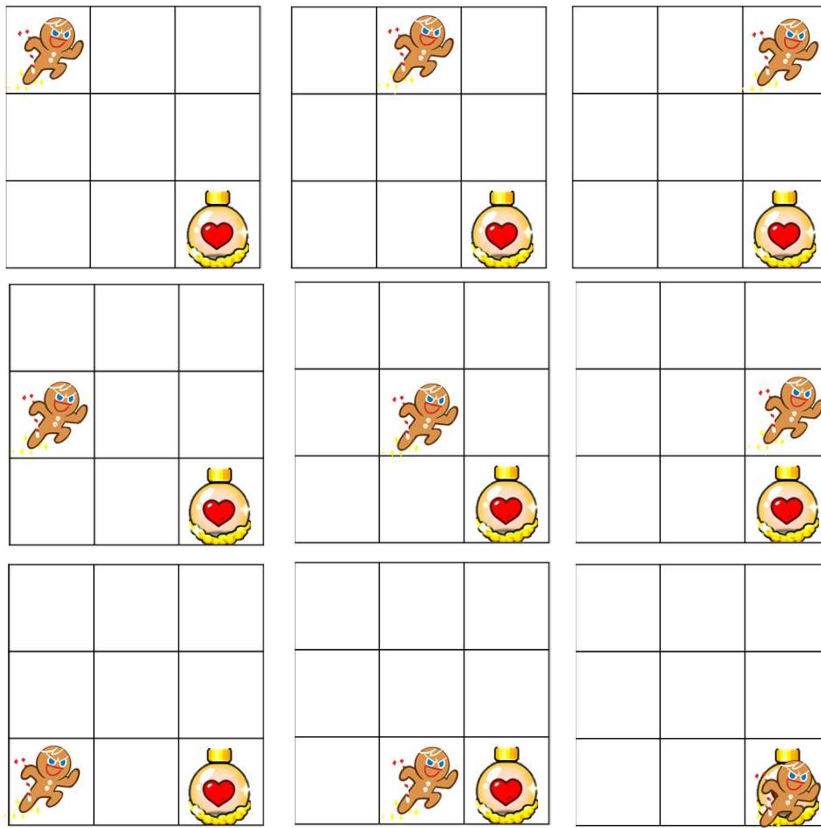


Agent: 주변 상태에 따라서 어떤 action을 취할지 판단을 내림
Environment: agent에게 state와 reward를 부여함
Reward: 주어진 state에서 취한 action으로부터 할당되는 보상

- Objective: 최소한으로 움직여서 물약이 있는 곳(terminal state)로 가기

MDP Example: grid-world

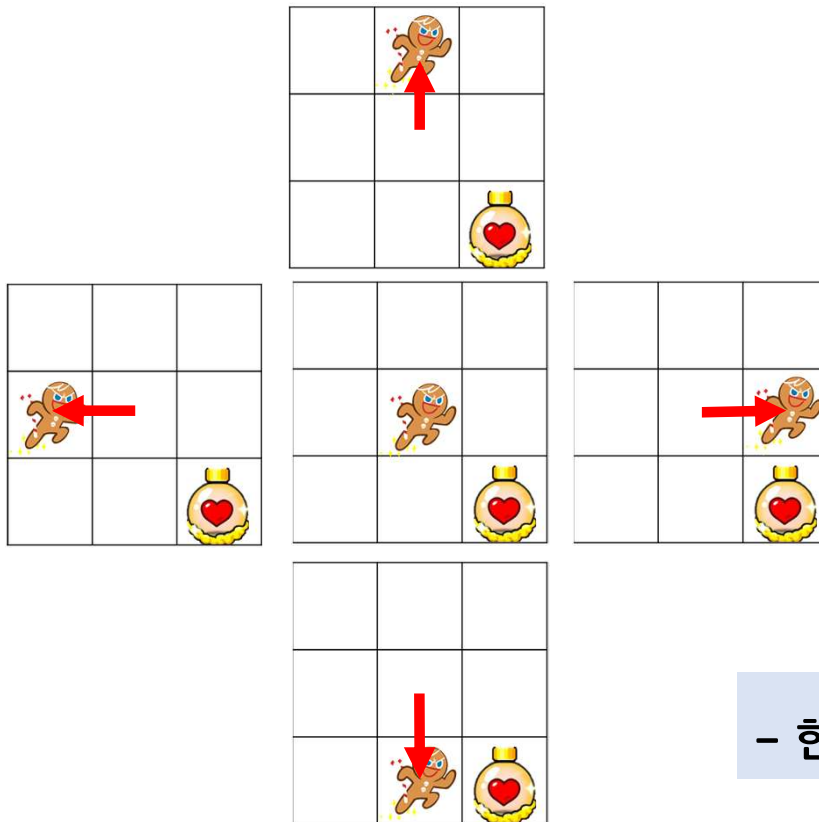
상태^{state} S: 에이전트가 관찰할 수 있는 환경의 상태 s_i 들의 집합



- 여기서 State는 각 grid에서 쿠키의 위치
- Terminal state를 포함

MDP Example: grid-world

행동 action A: 에이전트가 환경에 취할 수 있는 모든 행동 a_i 들의 집합

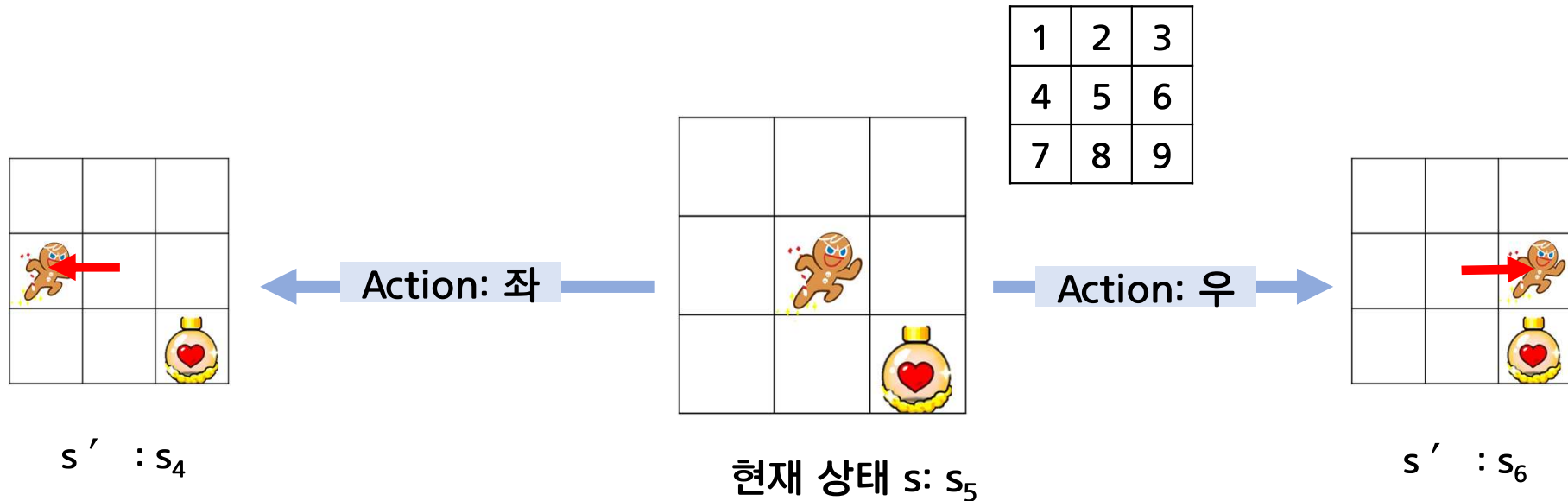


• 상, 하, 좌, 우

- 현재 상태에서 어떻게 행동할 지 정하는 정책(Policy): π

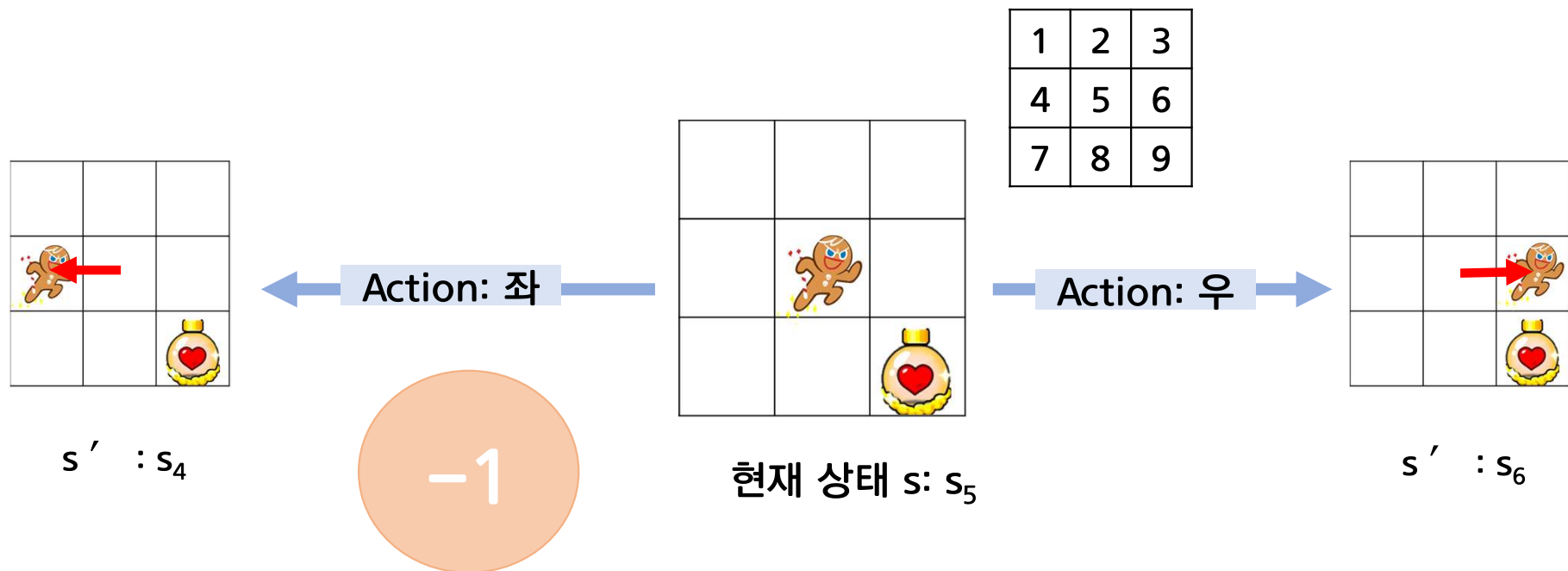
MDP Example: grid-world

전이 확률 transition probability P : 상태 s 에서 행동 a 를 수행했을 때 상태 s' 로 옮겨 갈 확률: $P(s, s')$



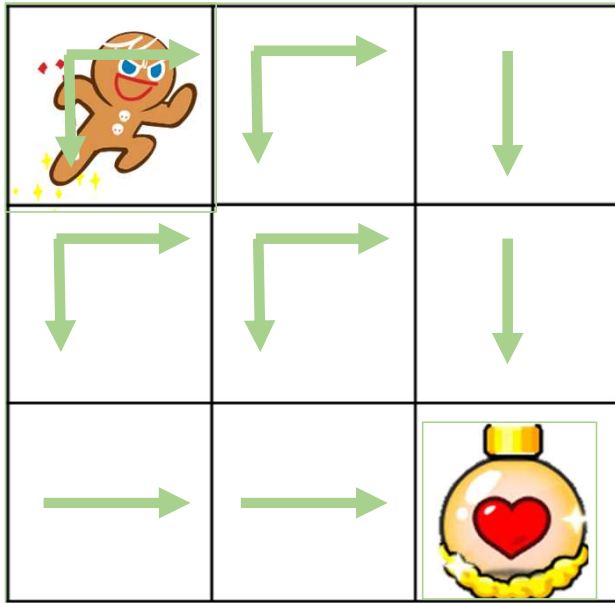
MDP Example: grid-world

보상reward R : 상태 s 에서 행동 a 를 수행하고 상태 s' 로 옮겨가며 환경에서 받는 보상



→ Reward를 최대로 만드는 최적의 정책 π^* 를 찾자

MDP Example: grid-world



→ Optimal policy π^*

Value function

미래 보상들의 합: 누적보상

$$R_t = r_{t+1} + r_{t+2} + \dots + r_T$$



S_t 에서 취한 액션 a_t 로 인하여 상태 S_{t+1} 에서 받은 보상

미래에 대한 불확실성 존재

즉각적인 보상만을 고려하여 행동을 선택할 수도 있음

Value function

- 감가율^{discount rate} γ : 미래에 받을 보상에 대한 신뢰도. 0에서 1 사이의 수

감가율 0: 미래의 보상 고려하지 않음

감가율 1: 미래의 보상을 매우 신뢰함

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{T-t-1} r_T$$

Value function 현재 상태 s 가 얼마나 좋은가?

- **가치함수** : 현재 상태에서 미래에 받을 것이라고 기대하는 보상의 합
- 임의의 상태 s , 정책 π 가 주어졌을 때 누적 보상의 기댓값

$$V^\pi(s) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t \mid s_0 = s, \pi \right]$$

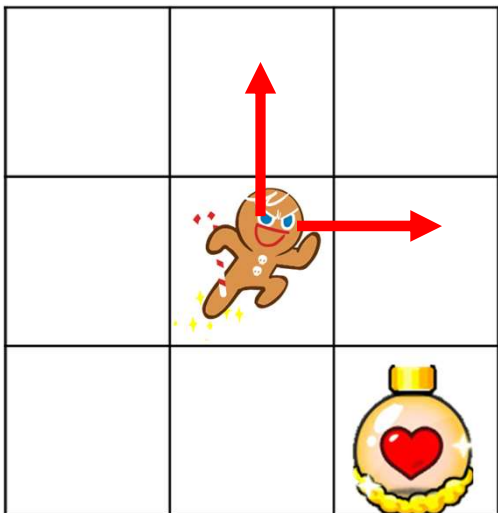
Q function

- **큐함수**: 지금 상태에서 이 행동을 선택했을 때 미래에 받을 것이라고 기대하는 보상의 합

$$Q^{\pi}(s, a) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t \mid s_0 = s, a_0 = a, \pi \right]$$

탐욕 정책(greedy policy)

지금 상태에서 큐함수가 가지는 값이 가장 높은 행동을 선택하는 것



$$\pi(s) = \operatorname{argmax}_{a'} Q(s, a') = a$$

벨만 방정식(Bellman equation)

큐함수를 찾는 가장 기본적인 방법

벨만 기대 방정식: $Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right]$

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

현재 상태와 행동에 대한 미래의 최대 보상

즉각적인 보상

다음 상태에서 얻을 수 있는 미래의 최대 보상

벨만 방정식(Bellman equation)

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

현재 상태와 행동에 대한 미래의 최대 보상

즉각적인 보상

다음 상태에서 얻을 수 있는 미래의 최대 보상

$$Q(s, a) = Q(s, a) + \alpha (r + \gamma Q(s', a') - Q(s, a))$$

점진적인 큐함수 업데이트 - 학습률 α

SARSA

ϵ - 탐욕정책

탐욕정책 - 에이전트가 다양한 방식으로 학습하지 못할 수 있음

ϵ : 수확과 탐험의 비율, 즉 무작위로 행동을 결정하는 비율

- 수확- 강화학습에서 현재의 정책을 그대로 따름
- 탐험- 현재의 정책을 무시하고 새로운 가능성을 추구함

ϵ - 탐욕정책

```
if random value >  $\epsilon$   
    argmax(a)  
else  
    random(a)
```

ϵ 값: 사용자 지정 ($0 < \epsilon < 1$)

랜덤으로 뽑은 숫자가 ϵ 보다 클 경우: 탐욕 정책으로 행동 결정

랜덤으로 뽑은 숫자가 ϵ 보다 작을 경우 : 랜덤하게 행동 결정

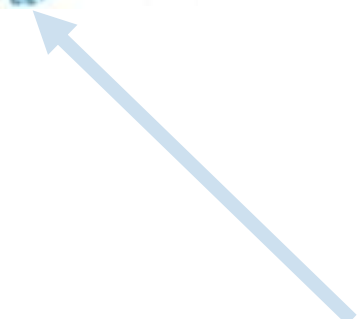
Q-Learning

SARSA의 문제점: 행동하는대로 학습을 한다.

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma Q(s', a') - Q(s, a))$$

ex) $Q(s, a)$ 가 가장 최고가 되어야하는데 탐험으로 인하여 $Q(s', a')$ 가 음의 보상을 얻게 되면 $Q(s, a)$ 가 작아져버림

Q-Learning

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$


- 행동: ϵ - 탐욕정책
- 학습: 다음 상태에서 어떤 행동을 할 때 다음 상태의 최대 큐함수를 이용하여 현재 상태의 큐함수로 업데이트

Q-Learning

- 큐러닝 구현 : 큐테이블로 표현
 - $\langle s, a, Q \rangle$ 이런식으로 상태를 행으로 가지는 테이블을 만든다.
- 간단한 강화학습 문제를 해결하기에 좋은 방법

참고자료

- <https://www.slideshare.net/WoongwonLee/ss-78783597>
- <https://horizon.kias.re.kr/14611/>
- CS231n 14강
- <https://blog.lgcns.com/1692>
- <https://m.blog.naver.com/plutonium235/221473299598>
- <https://multicore-it.com/112>
- <https://sumniya.tistory.com/3>

감사합니다