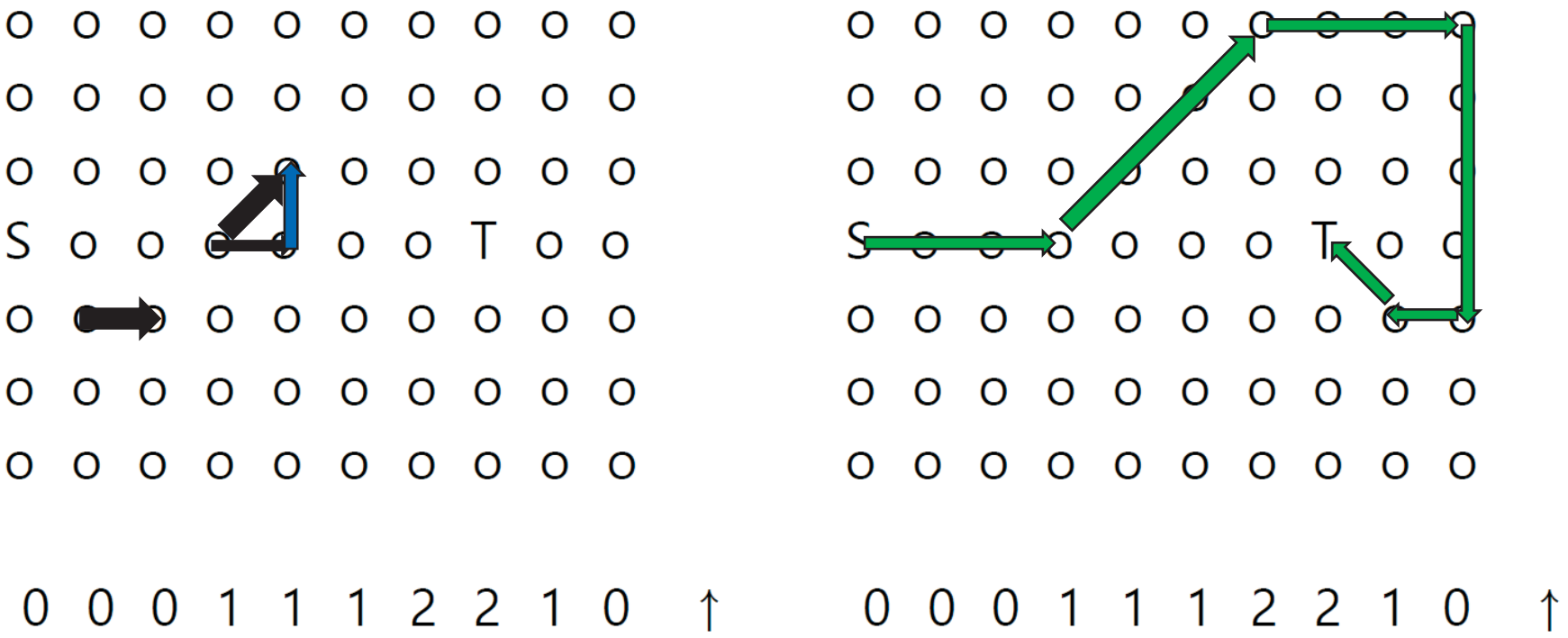
실습 예제 – SARSA

- WindyGridWorld Enviornment
- 수행 가능한 Action : 상하좌우 움직이기
- 특정 Column에서 위쪽 방향으로 바람이 불고 있음.



실습 예제 – SARSA

- WindyGridWorld Enviornment
  - 수행 가능한 Action : 상하좌우 움직이기
  - 특정 Column에서 위쪽 방향으로 바람이 불고 있음.
  - 바람이 불어 World 바깥으로 나가지는 않음.
    - 즉, 가장자리에서 바람의 영향으로 더 이상 밖으로 나가지 않음.

  - Reward : 매 Timestep마다 -1씩 주어짐.
  - S에서 시작하여 T까지 가는 경로 찾기.
  - Optimal Path : 최단경로

실습 예제 – SARSA

- Step 1. Make ϵ-greedy policy from given q, ϵ, nA

- Input : q(action-value function), ϵ, number of actions(nA)
- Output : A policy function $\pi(\cdot|s)$
  - Return **a function** that
    - takes the observation s(current state) as an argument, and
    - returns a list of probabilities
    - $\{\pi(a|s)\}_{a\in\mathcal{A}} = [\pi(0|s), \pi(1|s), \dots, \pi((nA-1)|s)]$

- Function을 return해야 하므로, 다음과 같이 작성해 주시면 됩니다.

실습 예제 – SARSA

- Step 1. Make ϵ-greedy policy from given q, ϵ, nA

```
def make_epsilon_greedy_policy(Q, epsilon, nA):
    def policy_fn(observation):
        
        
        
    return policy_fn
```

실습 예제 – SARSA

- Step 2. SARSA Algorithm

- Input :
  - env : OpenAI Enviornment
  - num_episodes : Number of Episodes
  - γ : Discount Factor
  - α : Hyperparameter for step-size
  - ϵ : Hyperparameter for epsilon-greedy policy

- Output : $q(s, a)$ table ***q*** and related statistics ***stats***.
  - q : updated q(s,a)
  - Stats : save each episode's total reward and length of episode.

실습 예제 – SARSA

- Step 2. SARSA Algorithm

```
def sarsa(env, num_episodes, discount_factor=1.0, alpha=0.5, epsilon=0.1):
    # The final action-value function.
    ## 각자 사용할 Q table을 정의해 줍시다.
    # 예시 : A nested dictionary that maps state -> (action -> action-value).
    # 혹은 simply 2d numpy array

    Q = defaultdict(lambda: np.zeros(env.action_space.n))


    # Keeps track of useful statistics
    ## stats 변수를 정의하고, 여기에 매 episode 길이와 episode reward를 기록합시다.

    # The policy we're following
    ## epsilon-greedy policy from Q로부터 policy를 얻습니다.

    # 각 episode에 대..
        ## Initialize가 필요합니다.
        ## 1. enviroement를 reset해주고 initial state를 얻습니다.
        ## 2. 이를 앞서 정의한 policy에 넣어 probability vector를 얻고,
        ## 3. 위에서 얻은 probability vector에 기반하여 action을 sampling합니다. REFER : np.random.choice
        # 매 time step에 대..
            # a. 주어진 action으로 step을 진행하고,
            # b. 위의 initialize한 과정과 비슷하게 앞서 정의한 policy에 넣어 probability vector를 얻은 뒤
            # c.  next_action 및 next_state를 얻습니다.
            # d. stats 변수에 episode reward와 episode length에 관련된 사항들을 기록합시다.
            # e. TD error를 계산하고, SARSA update 식을 토대로 Q를 update해 줍시다.
            # f. 만약 현재 enviornment가 terminal이라면, for문을 break해 주고,
            # g. 아니라면 action과 state를 update해줍시다.

    #return Q, stats
    return None
```

Define q = np.zeros(num_State,num_Action)
Numpy 2d array 등 편한 방식을 이용합시다.

매 episode의 reward와 length를 저장할 stats 변수를 정의합시다.

앞서 정의한 ϵ-greedy policy 함수에 Q, ϵ, nA를 넣어 policy function을 만듭시다. **Q가 update됨에 따라, policy도 같이 update되므로**, 맨 처음에 한 번 정의하는 것으로 충분합니다.

Episode에 대한 for문

Timestep에 대한 for문

실습 예제 – SARSA

- Step 2’. Expected SARSA Algorithm

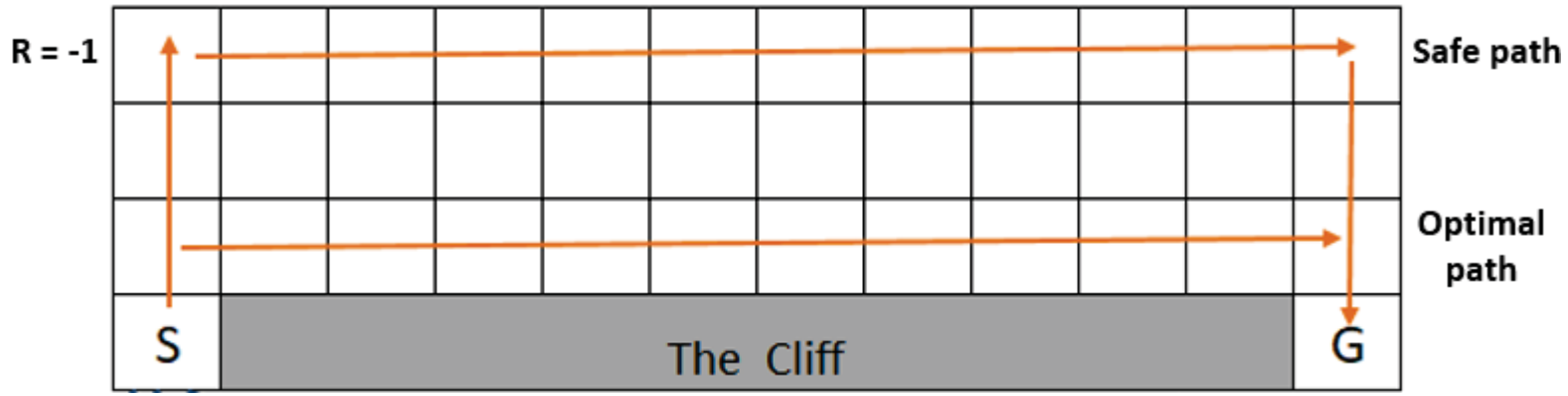
- Timestep 내 for문의 update 식에서 Next action에 대해서 계산했던 식을 일부 수정하는 것으로 충분합니다.

- Policy에서 얻어졌던 $[\pi_q(a|s)]_a$ 벡터와 $[q(s_{t+1}, a)]_a$ 벡터를 dot product하면,

- 평균($E_\pi[q(s_{t+1}, a_{t+1})|s_{t+1}\ given]$)이 얻어집니다.

실습 예제 – Q-Learning

- CliffWalking Enviornment
  - 수행 가능한 Action : 상하좌우 움직이기
  - 하단의 Cliff 영역에 도달하는 경우를 제외하면 매 이동마다 -1의 reward
  - 하단의 Cliff 영역에 도달하면 -100의 reward를 얻음과 함께 초기 위치로 이동
  - G 지점에 도착하면 Terminate



실습 예제 – Q-Learning

- (Step 1). Make ϵ-greedy policy from given q, ϵ, nA

```
def make_epsilon_greedy_policy(Q, epsilon, nA):
    def policy_fn(observation):

    return policy_fn
```

- **아까 SARSA 시간에 만든 것 활용!**

실습 예제 – Q-Learning

- Step 2. Q-Learning Algorithm

- Input :
  - env : OpenAI Enviornment
  - num_episodes : Number of Episodes
  - γ : Discount Factor
  - α : Hyperparameter for step-size
  - ϵ : Hyperparameter for epsilon-greedy policy

- Output : $q(s, a)$ table ***q*** and related statistics ***stats***.
  - q : updated q(s,a)
  - Stats : save each episode's total reward and length of episode.

실습 예제 – Q-Learning

- Step 2. Q-Learning Algorithm

```python
def q_learning(env, num_episodes, discount_factor=1.0, alpha=0.5, epsilon=0.1):
    # The final action-value function.
    ## 각자 사용할 Q table을 정의해 줍시다.
    # 예시 : A nested dictionary that maps state -> (action -> action-value).
    # 혹은 simply 2d numpy array

    # Keeps track of useful statistics
    ## stats 변수를 정의하고, 여기에 매 episode 길이와 episode reward를 기록합시다.

    # The policy we're following
    ## epsilon-greedy policy from Q로부터 policy를 얻습니다.

    # 매 Episode마다 ..
        ## Initialize가 필요합니다.
        ## 1. enviroement를 reset해주고 initial state를 얻습니다.

        # 매 time step마다..
            # a. 위에서 정의한 policy에 현재 state를 넣어 probability vector를 얻은 뒤
            # b. probability vector를 이용하여 다음 action을 sampling하고,
            # c. 이로부터 next_state를 얻습니다.
            # d. stats 변수에 episode reward와 episode length에 관련된 사항들을 기록합시다.
            # e. TD error를 계산하고, Q-Learning 업데이트 식을 토대로 Q를 update해 줍시다.
            # f. 만약 현재 environment가 terminal이라면, for문을 break해 주고,
            # g. 아니라면 action과 state를 update해줍시다.

    #return Q, stats
    return None
```

Define q = np.zeros(num_State,num_Action)
Numpy 2d array 등 편한 방식을 이용합시다.

매 episode의 reward와 length를 저장할 stats 변수를 정의합시다.

앞서 정의한 ϵ-greedy policy 함수에 Q, ϵ, nA를 넣어 policy function을 만듭시다. **Q가 update됨에 따라, policy도 같이 update되므로**, 맨 처음에 한 번 정의하는 것으로 충분합니다.

SARSA와 달라지는 part!

실습 예제 – Q-Learning

- Step 2’. **학습된 q로부터 π_*를 얻고**, 학습된 policy를 이용하여 여러 강화학습 환경을 관찰해 봅시다.
  - CliffWalking
  - WindyGridWorld
  - Copy-v0
  - HomeEscape(첫 시간!)

```
env.reset()

obs_log = []
cumulative_reward = []

for t in range(200):
    env.render()

    # your agent here
    # 이번에는 random action이 아니라 $\pi_{Q^*, epsion}$에 기반한 policy를 얻어봅시다.
    [                                                                    ]

    ## output : (next_state, reward, is_terminal, debug_info)
    [                                                                    ]
    [                                                                    ]
    [                                                                    ]

    ## if is_terminal == True, then break for loop
    if(done):
        clear_output(wait = True)
        sleep(0.1)
        print("Episode finished after {} timesteps".format(t + 1))
        sleep(0.05)
        env.render()
        break

    sleep(0.1)
    clear_output(wait = True)

env.close()
```

실습 예제 – Q-Learning

- Step 3[Advanced].
  - 비슷한 방식으로, Double Q-Learning을 구현해 봅시다.

Algorithm 1 Double Q-learning

1: Initialize Q^A,Q^B,s
2: **repeat**
3: Choose a, based on $Q^A(s,\cdot)$ and $Q^B(s,\cdot)$, observe r, s'
4: Choose (e.g. random) either UPDATE(A) or UPDATE(B)
5: **if** UPDATE(A) **then**
6: Define $a^* = \arg\max_a Q^A(s',a)$
7: $Q^A(s,a) \leftarrow Q^A(s,a) + \alpha(s,a)\left(r + \gamma Q^B(s',a^*) - Q^A(s,a)\right)$
8: **else if** UPDATE(B) **then**
9: Define $b^* = \arg\max_a Q^B(s',a)$
10: $Q^B(s,a) \leftarrow Q^B(s,a) + \alpha(s,a)(r + \gamma Q^A(s',b^*) - Q^B(s,a))$
11: **end if**
12: $s \leftarrow s'$
13: **until** end

ϵ-greedy policy
based on $q^A(s,a) + q^B(s,a)$

Randomly choose A or B

Q & A