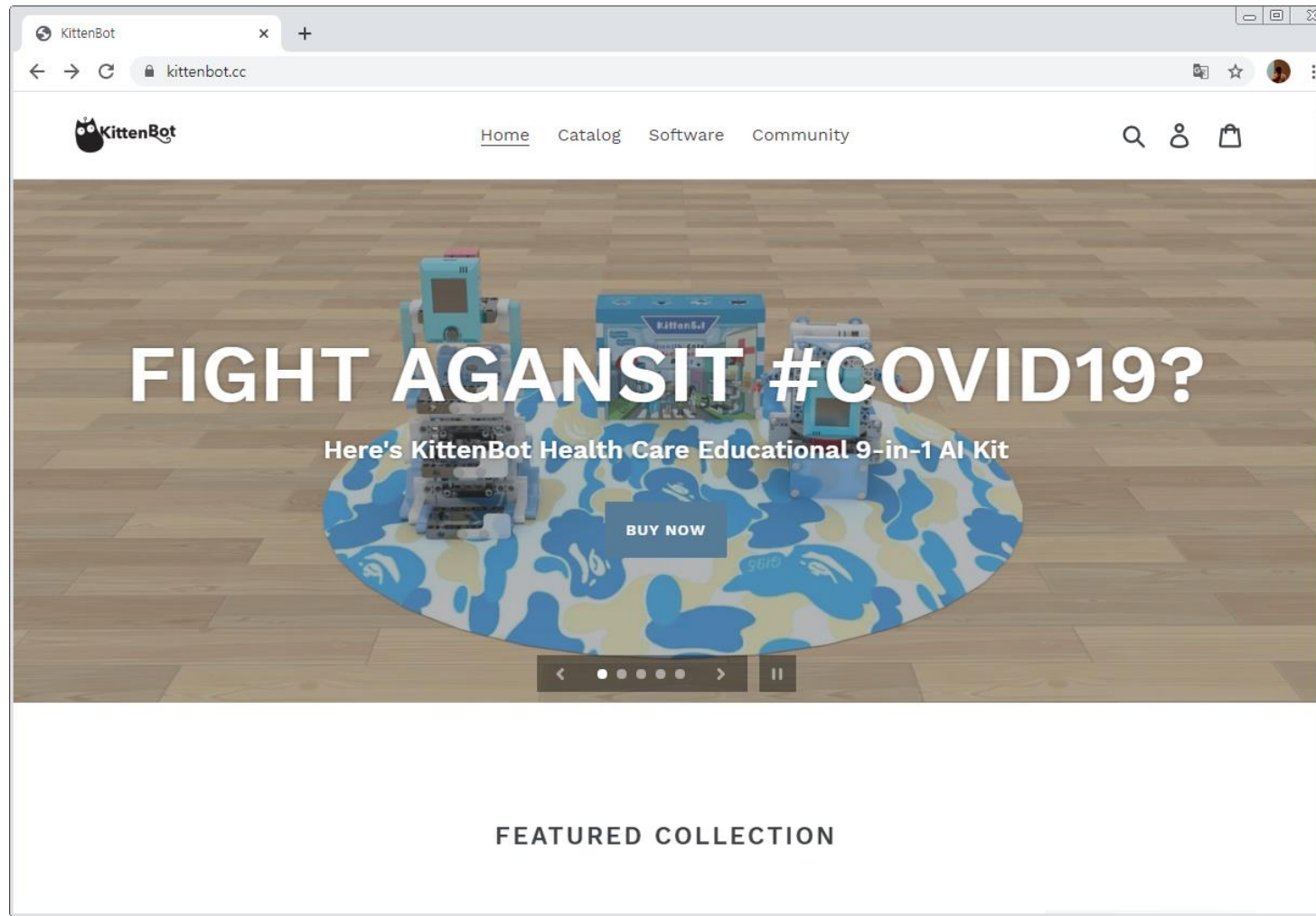


햄스터 · 햄스터S로 배우는 인공지능

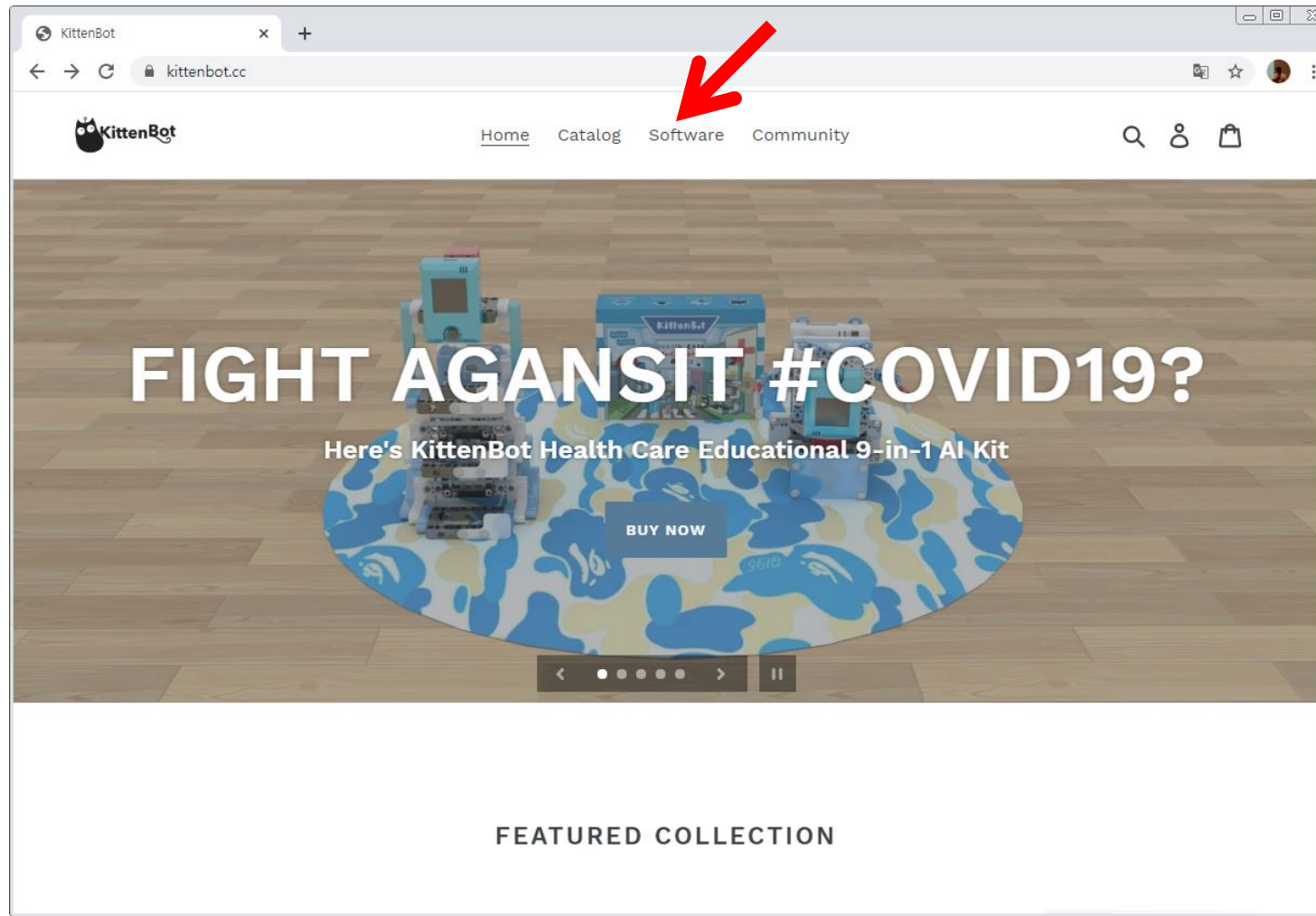
블록 코딩 #3

블록 코딩 추가

1 <https://www.kittenbot.cc>



2 Software 클릭



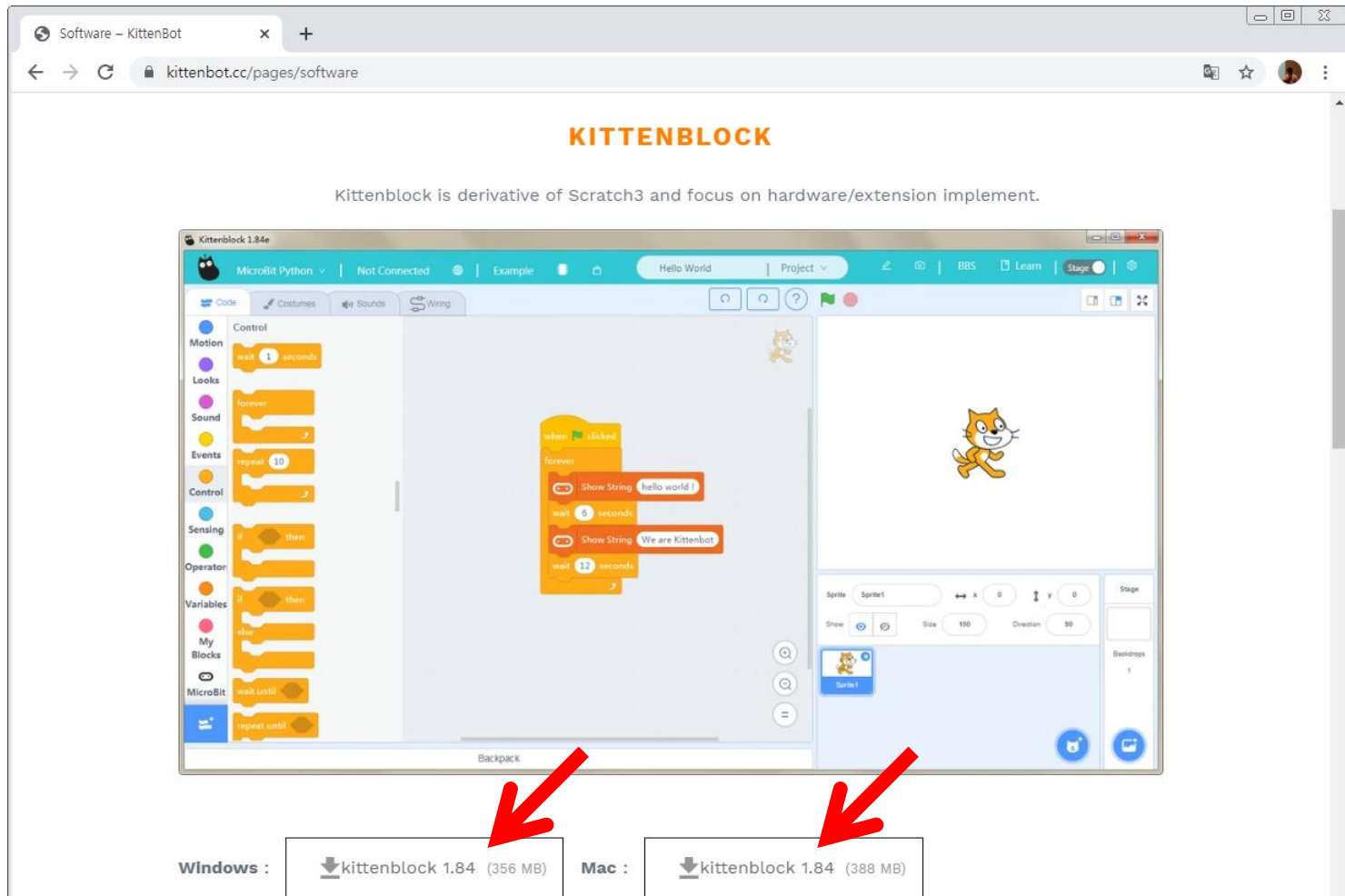
3 아래로 약간 스크롤하여 OS에 맞는 설치 파일 다운로드

Software - KittenBot

kittenbot.cc/pages/software

KITTENBLOCK

Kittenblock is derivative of Scratch3 and focus on hardware/extension implement.



The screenshot shows the Kittenblock website. At the top, there's a browser tab 'Software - KittenBot' and the URL 'kittenbot.cc/pages/software'. The main heading is 'KITTENBLOCK' in orange. Below it, a subtitle says 'Kittenblock is derivative of Scratch3 and focus on hardware/extension implement.' The central part of the page features a large image of the Kittenblock 1.84e software interface. This interface includes a top menu bar with 'MicroBit Python', 'Not Connected', 'Example', and 'Hello World'. It has a left sidebar with categories like Control, Motion, Looks, Sound, Events, Control, Sensing, Operator, Variables, My Blocks, and MicroBit. The main workspace shows a Scratch-style script with a 'when clicked' event, a 'say' block, and two 'show string' blocks. The right sidebar shows a 'Sprite' section with 'Sprite1' and a 'Stage' section. Below the software interface image, there are two download buttons. The first button is labeled 'Windows :' and the second 'Mac :'. Both buttons have a download icon and the text 'kittenblock 1.84 (356 MB)' for Windows and 'kittenblock 1.84 (388 MB)' for Mac. Two red arrows point from the bottom of the software interface image to these two download buttons.

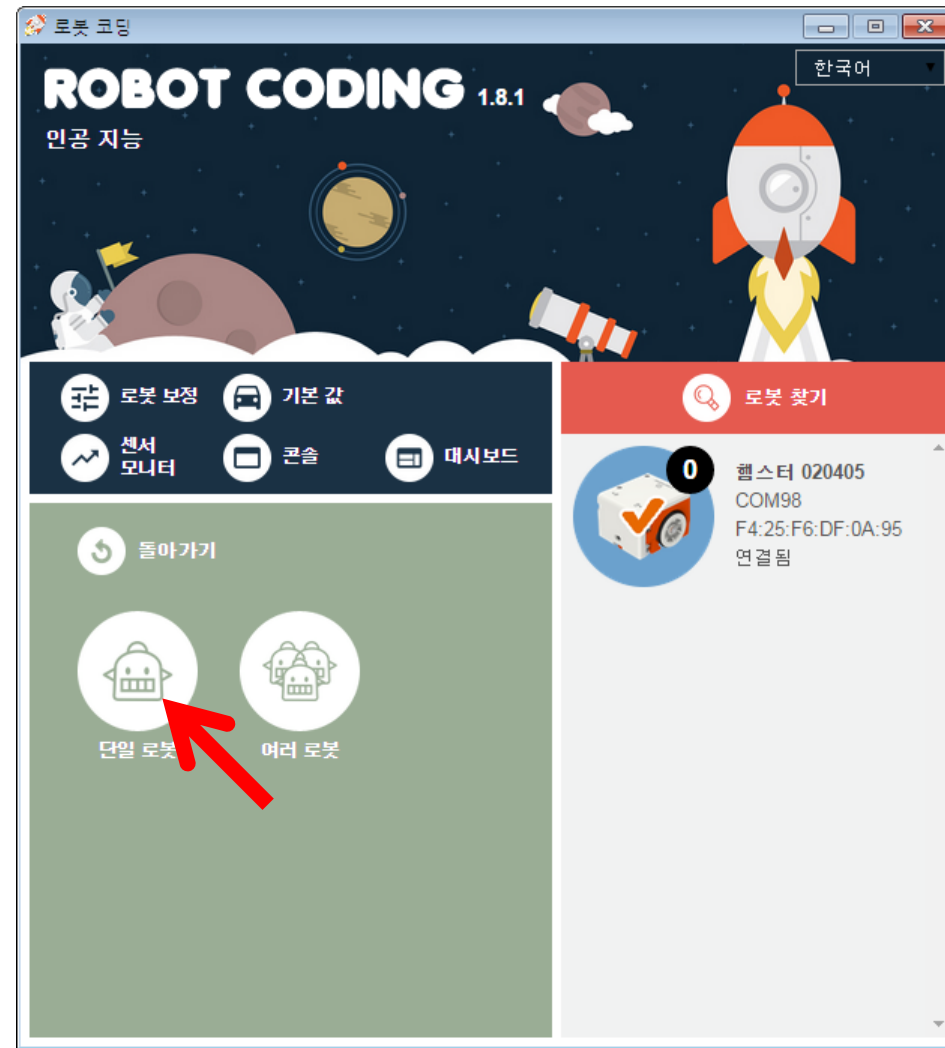
Windows : kittenblock 1.84 (356 MB)

Mac : kittenblock 1.84 (388 MB)

1 인공 지능 클릭



2 단일 로봇 클릭



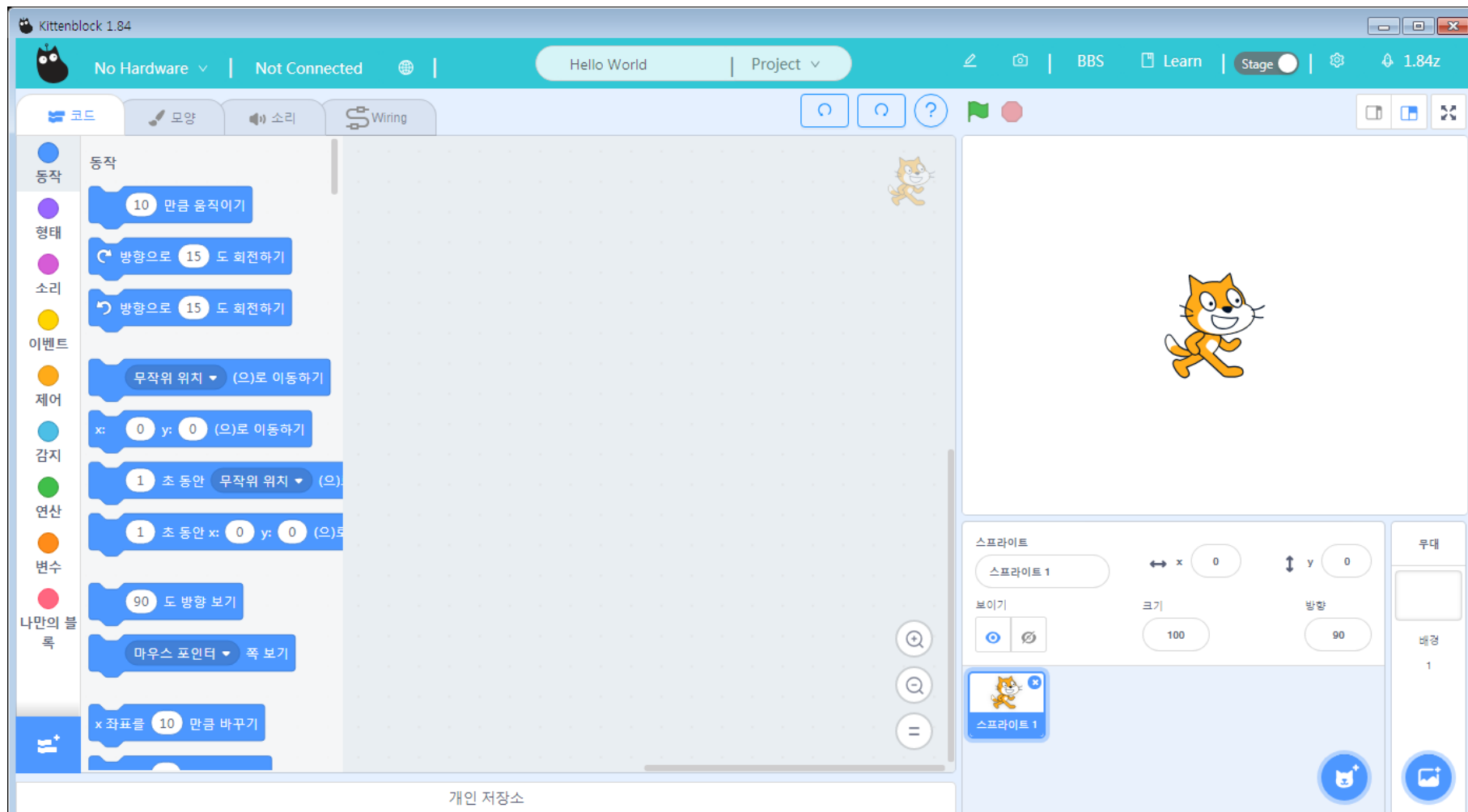
3 키튼블록 클릭



4 키튼블록 연결 프로그램 실행됨



5 설치된 키튼블록 실행



6 확장 클릭

Kittenblock 1.84

No Hardware | Not Connected | Hello World | Project

코드 | 모양 | 소리 | Wiring

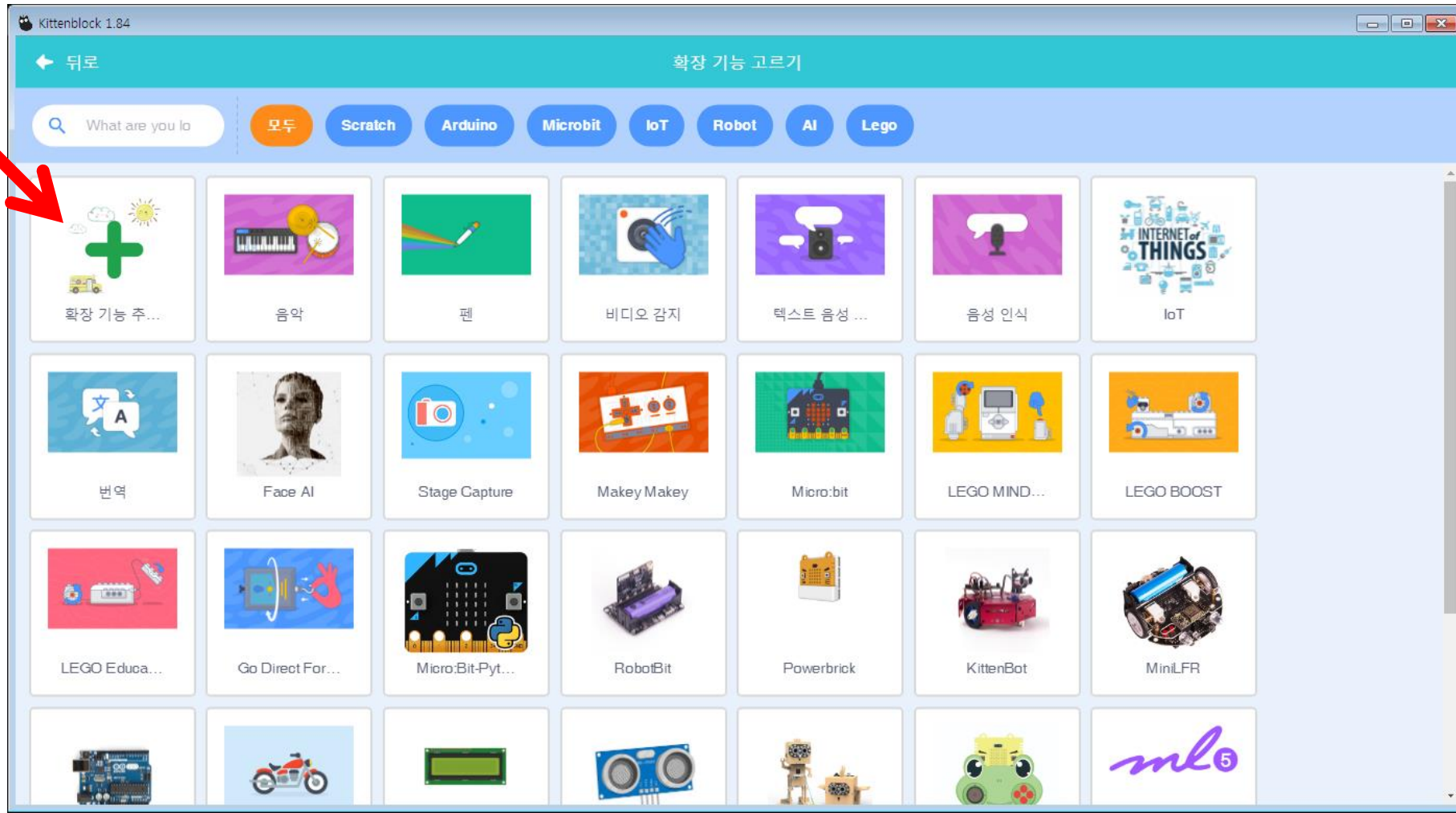
동작
형태
소리
이벤트
제어
감지
연산
변수
나만의 블록

10 만큼 움직이기
방향으로 15 도 회전하기
방향으로 15 도 회전하기
무작위 위치 (으)로 이동하기
x: 0 y: 0 (으)로 이동하기
1 초 동안 무작위 위치 (으)로 이동하기
1 초 동안 x: 0 y: 0 (으)로 이동하기
90 도 방향 보기
마우스 포인터 쪽 보기
x 좌표를 10 만큼 바꾸기

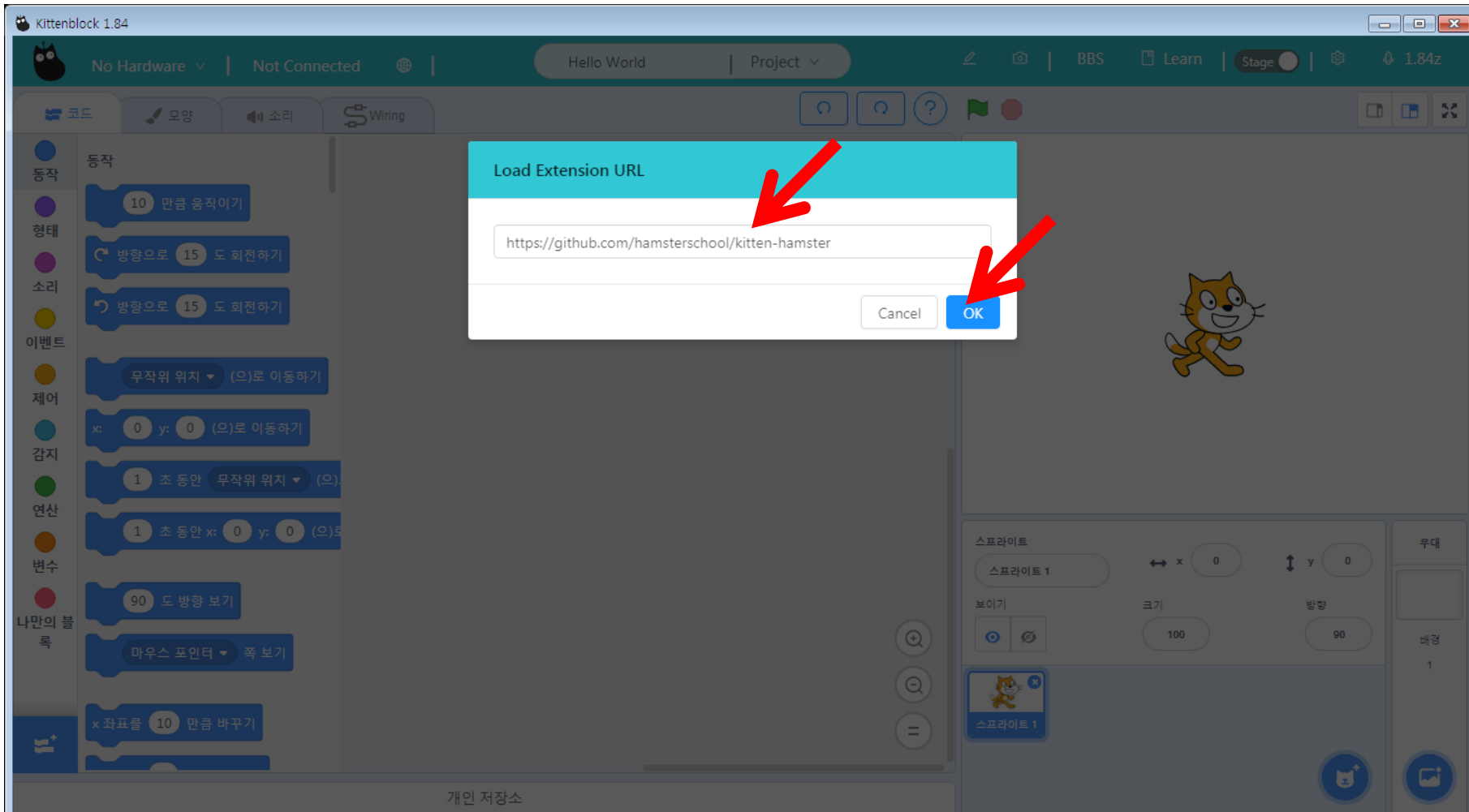
스프라이트
스프라이트 1
보이기
크기 100
방향 90
무대
배경 1

개인 저장소

7 확장 기능 추가 클릭



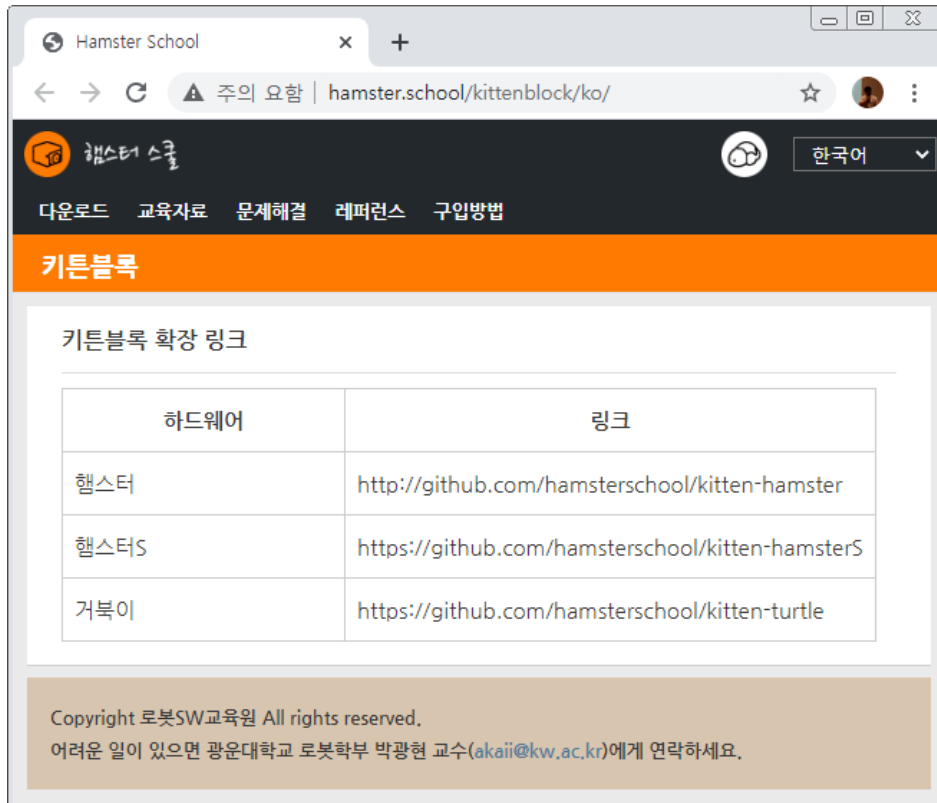
8 URL 입력 → OK 클릭



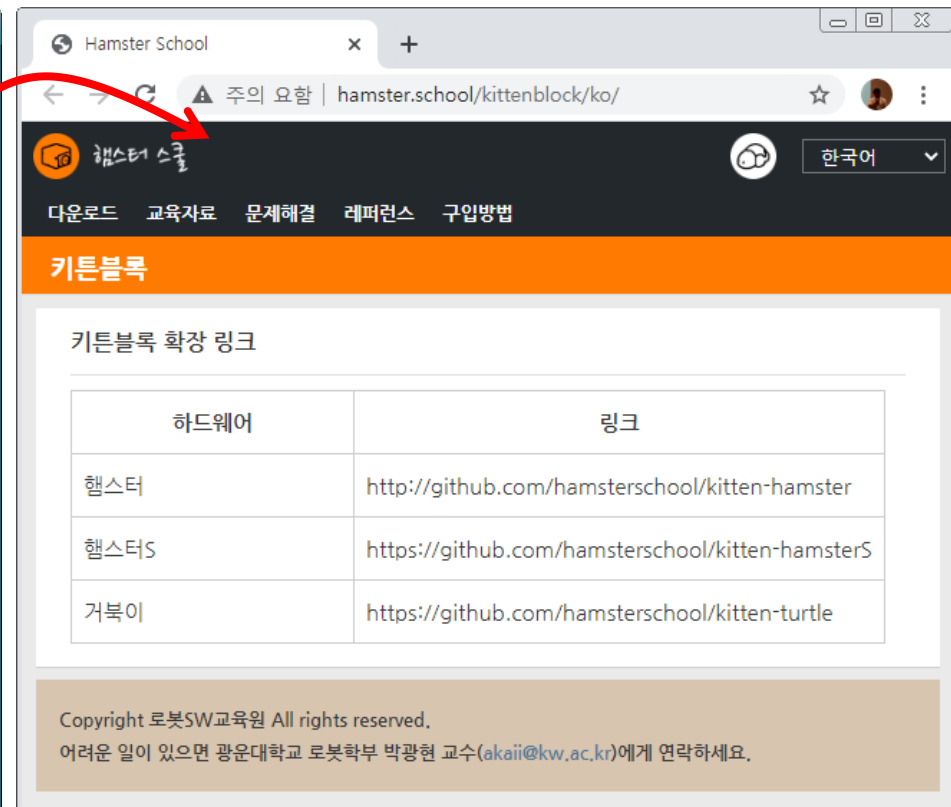
- URL 리스트

하드웨어	링크 주소
햄스터	http://github.com/hamsterschool/kitten-hamster
햄스터S	https://github.com/hamsterschool/kitten-hamsterS
거북이	https://github.com/hamsterschool/kitten-turtle

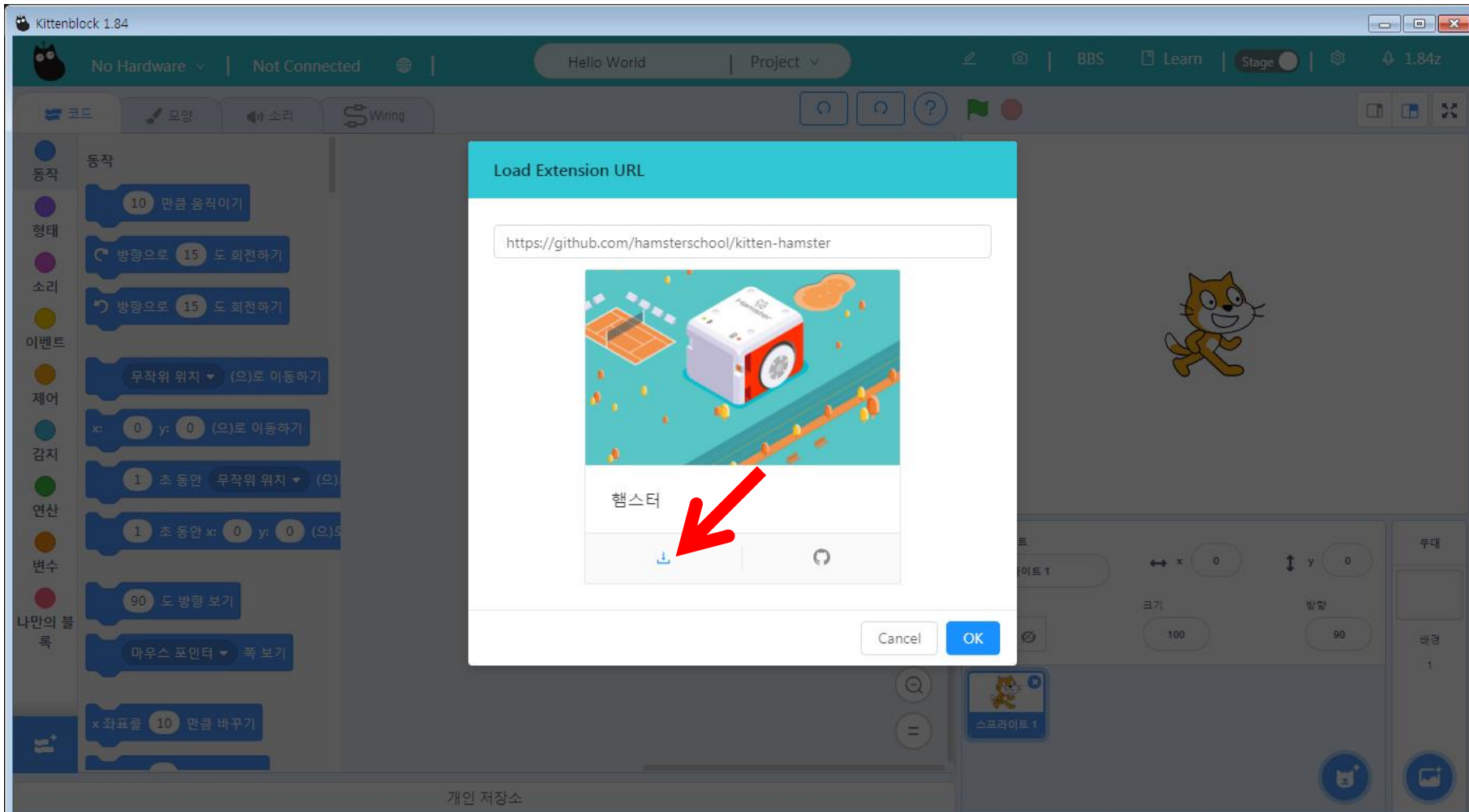
- 또는
<http://hamster.school/kittenblock>



- 또는 물음표 클릭

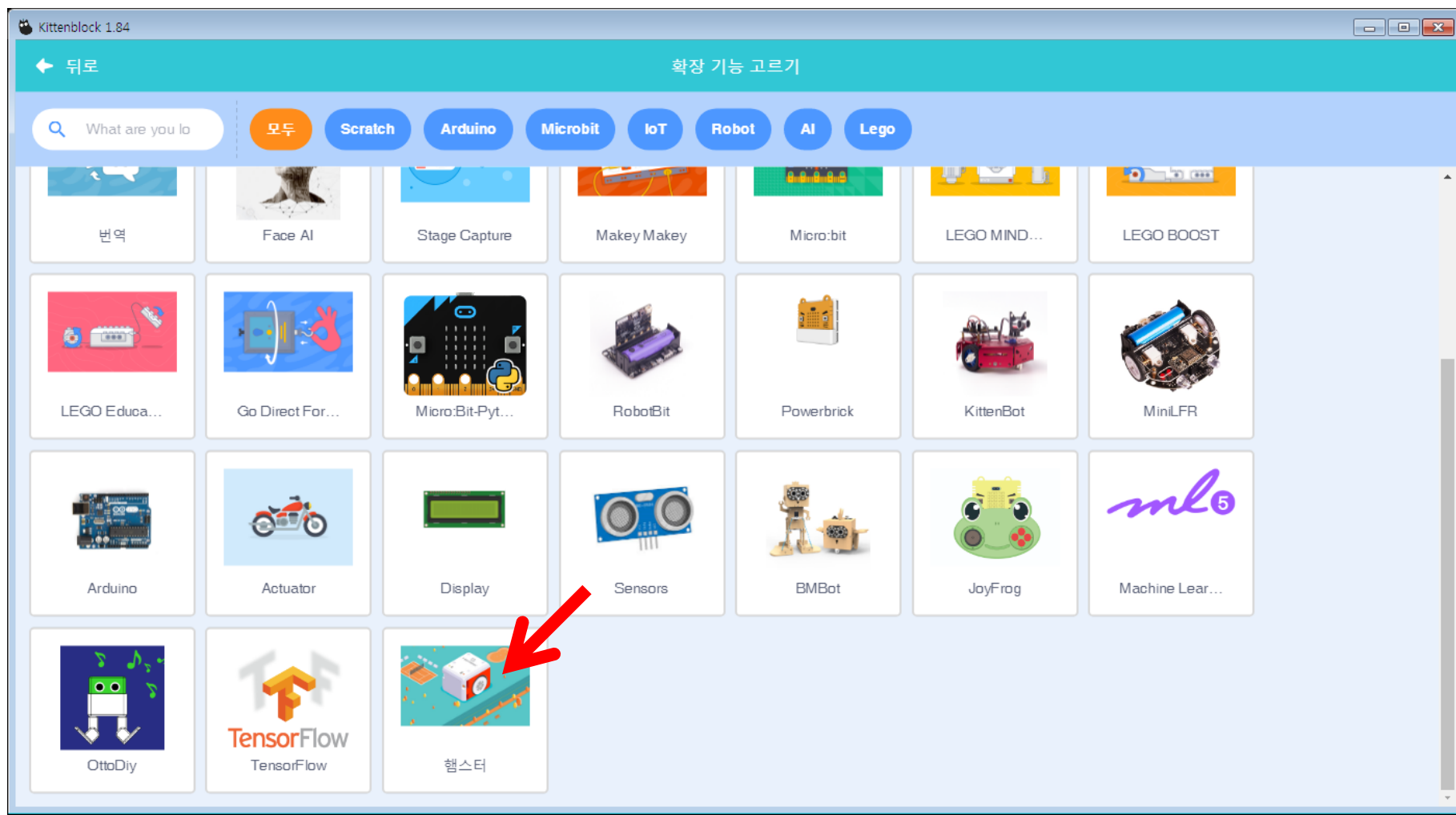


9 다운로드 버튼 클릭

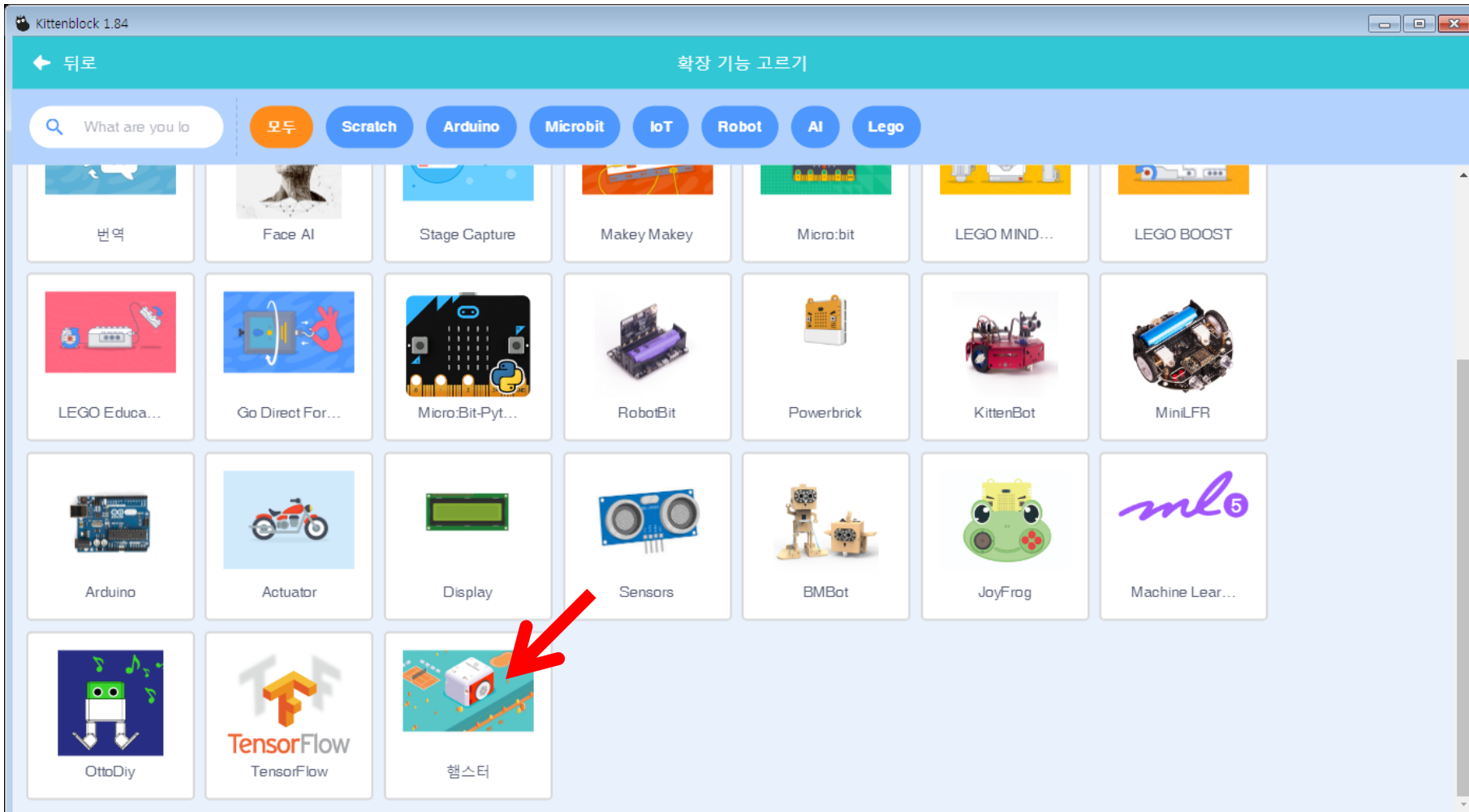


추가된 이후에는 다시 URL 입력하여 다운로드할 필요 없습니다.

10 키튼블록 새로 실행됨 → 확장 기능에 햄스터 추가되어 있음



11 햄스터 클릭



12 햄스터 블록 추가됨

The screenshot displays the Kittenblock 1.84 application window. The top bar includes the title 'Kittenblock 1.84', status indicators for 'No Hardware' and 'Not Connected', a 'Hello World' button, and a 'Project' dropdown. The main interface is divided into a left sidebar, a central workspace, and a right sidebar.

Left Sidebar: A vertical menu with colored circles representing different block categories: 동작 (Action), 형태 (Shape), 소리 (Sound), 이벤트 (Event), 제어 (Control), 감지 (Sensing), 연산 (Math), 변수 (Variables), 나만의 블록 (My Blocks), and 햄스터 (Hamster). The '햄스터' category is currently selected and highlighted in blue.

Central Workspace: A large grid area for building code. It contains several green 'Hamster' blocks, each featuring a small hamster icon. The blocks include:

- 말판 앞으로 한 칸 이동하기 (Move forward one space)
- 말판 왼쪽으로 한 번 돌기 (Turn left once)
- 앞으로 1 초 이동하기 (Move forward 1 second)
- 뒤로 1 초 이동하기 (Move backward 1 second)
- 왼쪽으로 1 초 돌기 (Turn left 1 second)
- 왼쪽 바퀴 10 오른쪽 바퀴 10 (Set left wheel to 10, right wheel to 10)
- 왼쪽 바퀴 30 오른쪽 바퀴 30 (Set left wheel to 30, right wheel to 30)
- 왼쪽으로 바퀴 10 만큼 바꾸기 (Change wheel by 10)
- 왼쪽으로 바퀴 30 (으)로 정하기 (Set wheel to 30)
- 검은색 선을 왼쪽 바닥 (Draw black line to the left bottom)

Right Sidebar: Contains settings for the selected sprite. It includes a '스프라이트' (Sprite) section with a dropdown for '스프라이트 1', a '보이기' (Visibility) section with '보이기' (Show) and '숨이기' (Hide) buttons, a '크기' (Size) section with a value of 100, and a '방향' (Direction) section with a value of 90. Below these are buttons for '무대' (Stage) and '배경' (Background).

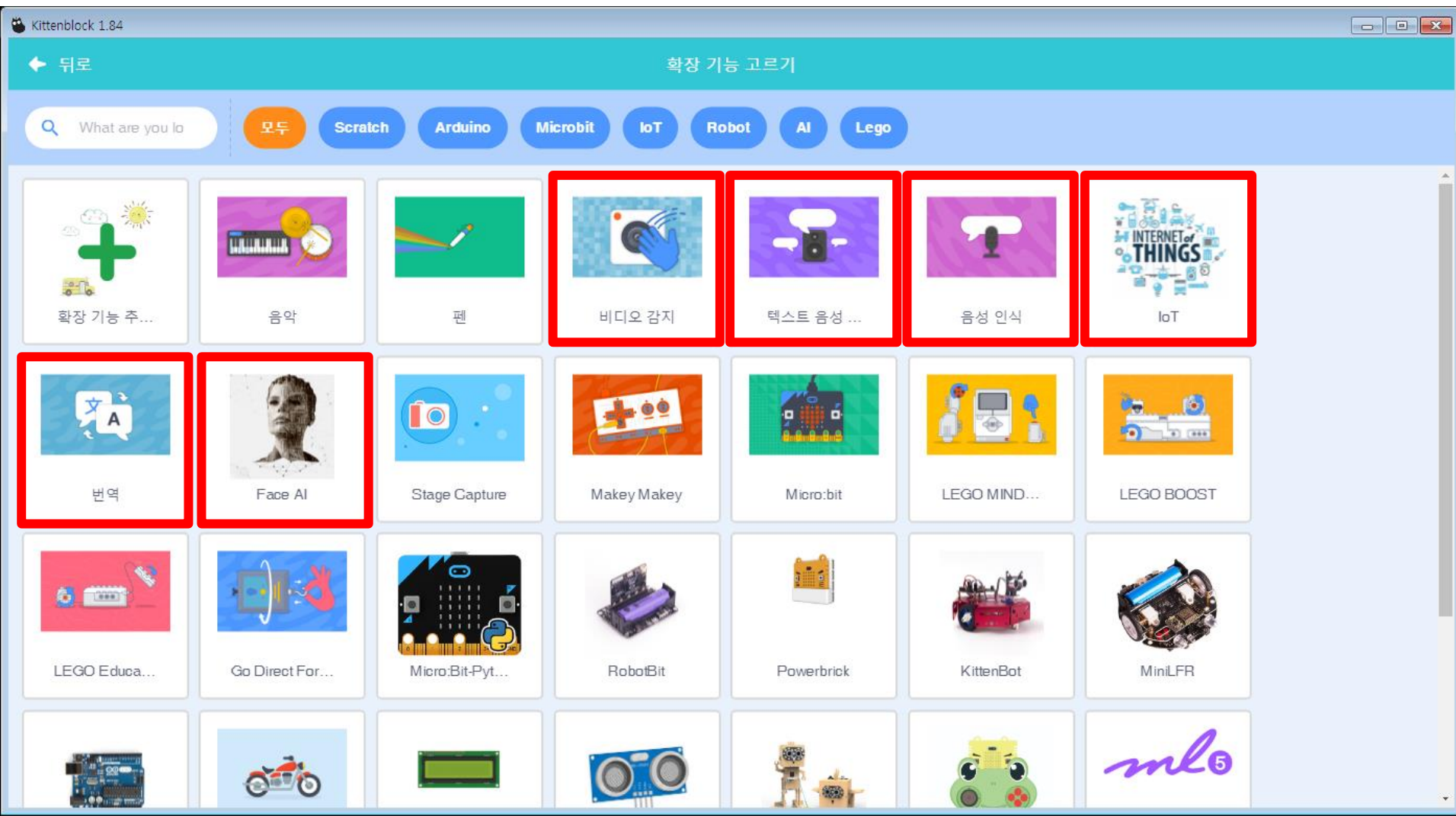
At the bottom of the window, there is a '개인 저장소' (Personal Storage) button.

아래 두 가지 중 어느 것을 먼저 해도 상관 없습니다.

- 로봇 코딩에서 키튼블록 연결 프로그램 실행
- 키튼블록에서 확장 기능 추가 → 햄스터 추가

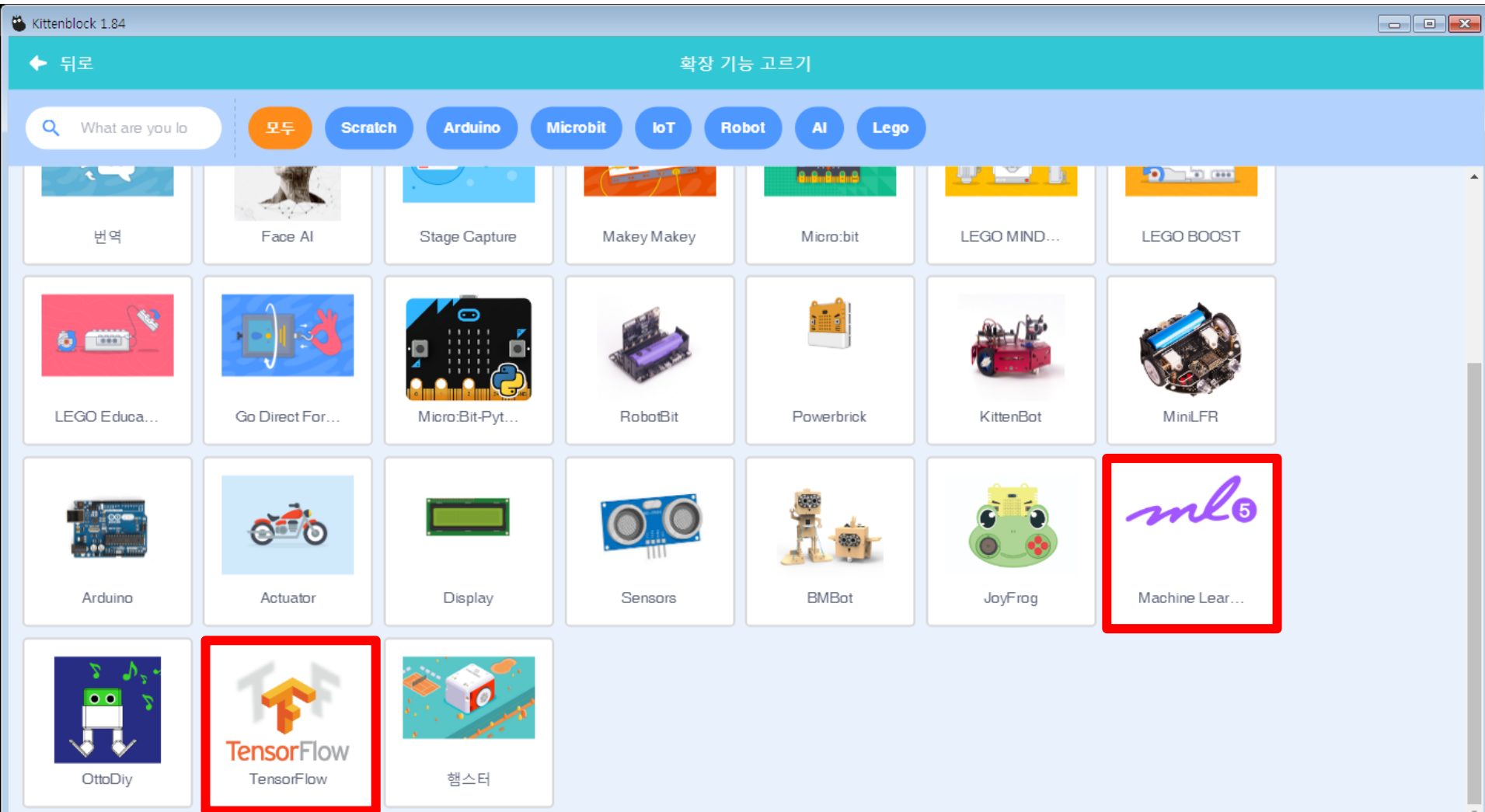
어떤 기능이 있을까요?

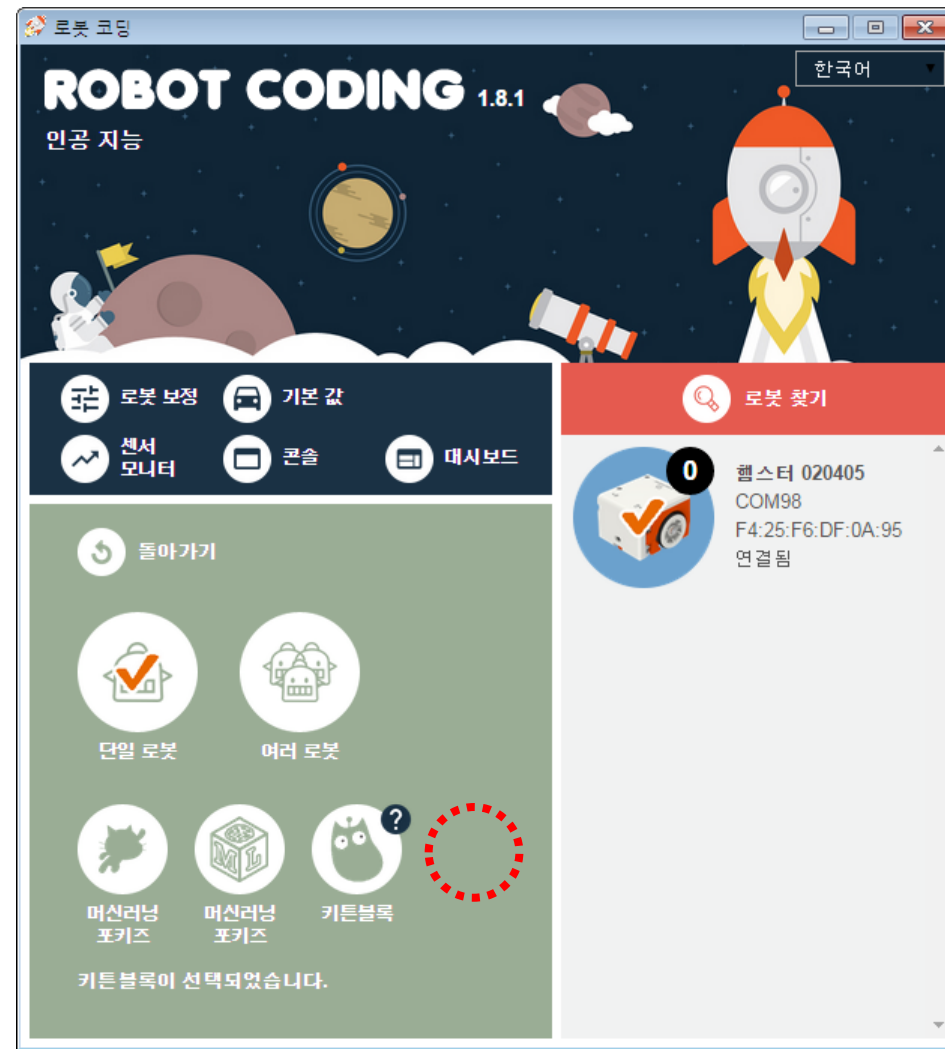
22



어떤 기능이 있을까요?

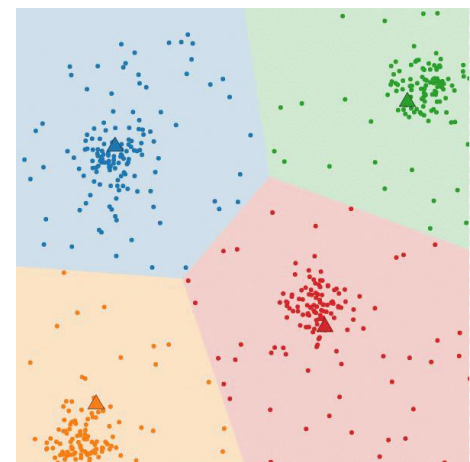
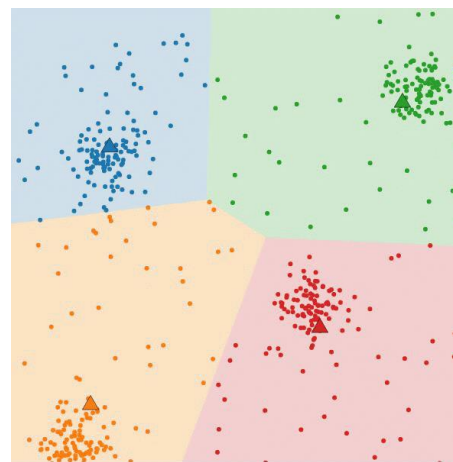
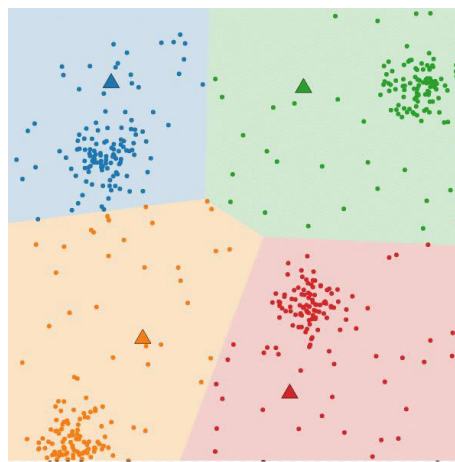
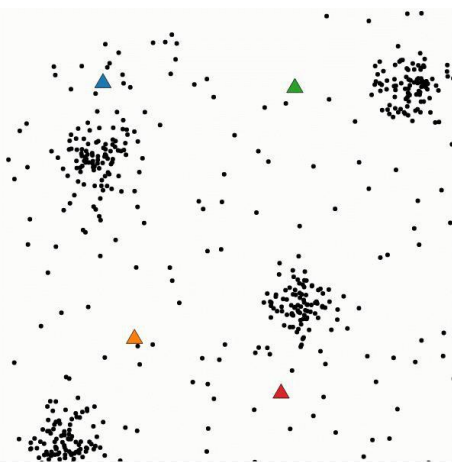
23






선생님을 위한 보충 설명

- 1 초기화 : k개의 중심 설정
- 2 각 데이터를 가장 가까운 중심점의 클러스터에 할당
- 3 각 클러스터에 속한 데이터들의 평균을 새로운 중심점으로 갱신

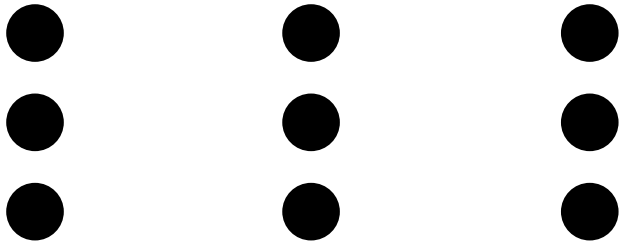


- 1 초기화 : k개의 중심 설정
 - 2 각 데이터를 가장 가까운 중심점의 클러스터에 할당 ←
 - 3 각 클러스터에 속한 데이터들의 평균을 새로운 중심점으로 갱신
- 

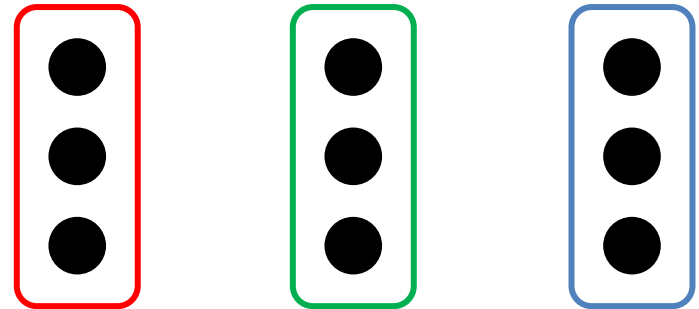
종료 조건

- 중심점에 변화가 없을 때까지
- 각 데이터의 소속 클러스터가 바뀌지 않을 때까지
- 지정된 횟수만큼

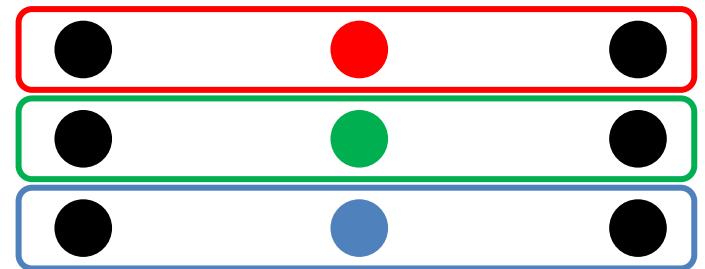
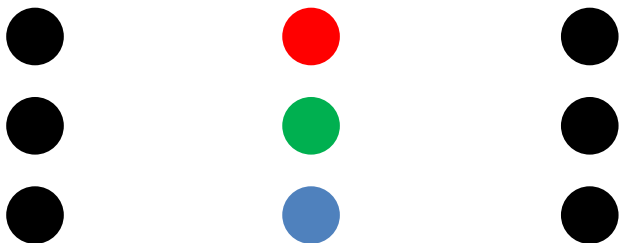
k=3



이렇게 되어야 할 것 같은데



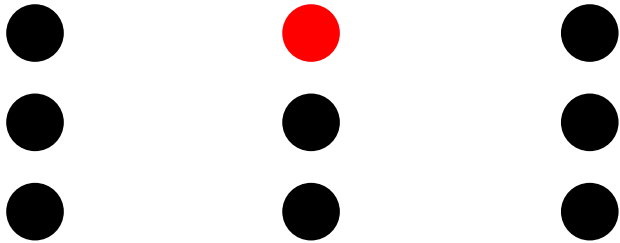
초기 중심을 이렇게 하면



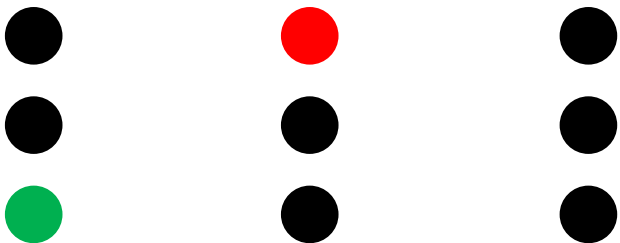
개선된 초기화 방법 #1

29

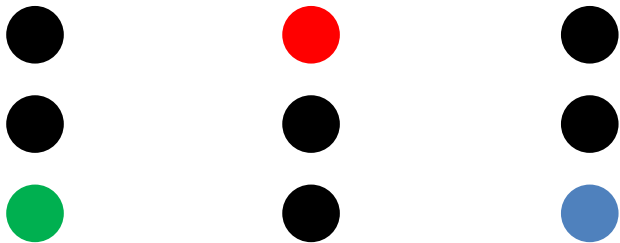
k=3



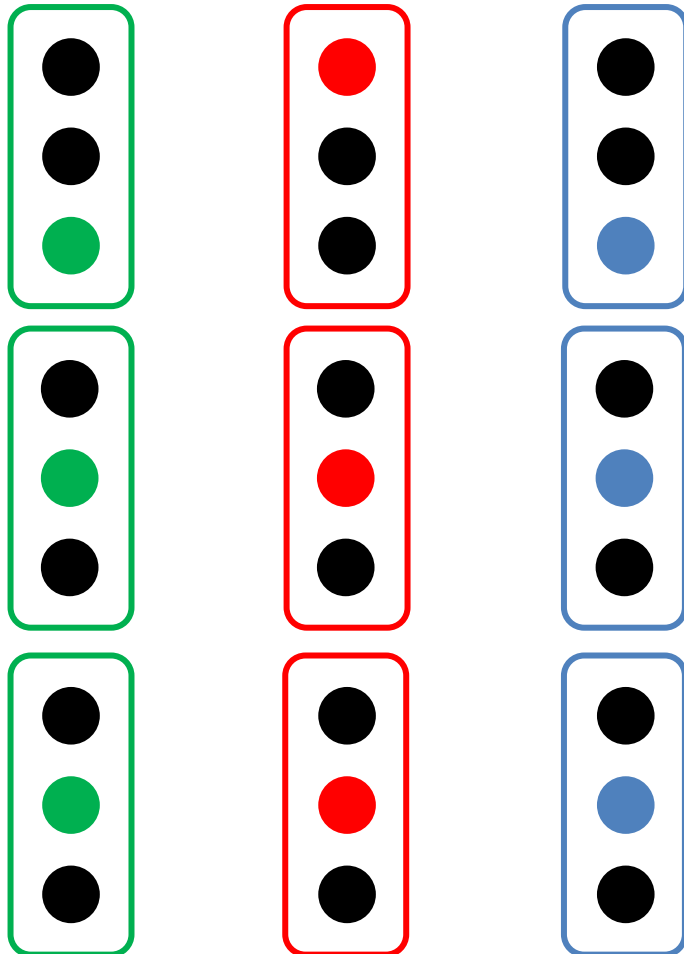
첫 번째 중심은 랜덤으로



두 번째 중심은 첫 번째 중심에서 가장 먼 것



세 번째 중심은 첫 번째와 두 번째 중심에서 가장 먼 것



가장 가까운 중심의 클러스터에 할당

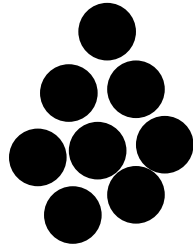
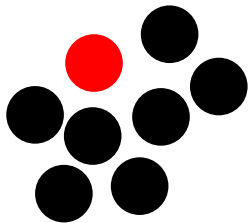
중심 위치 갱신

가장 가까운 중심의 클러스터에 할당

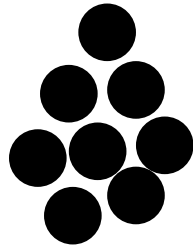
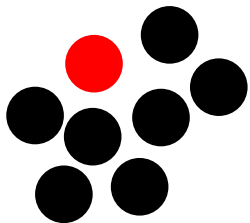
개선된 초기화 방법 #1

31

$k=2$



첫 번째 중심은 랜덤으로



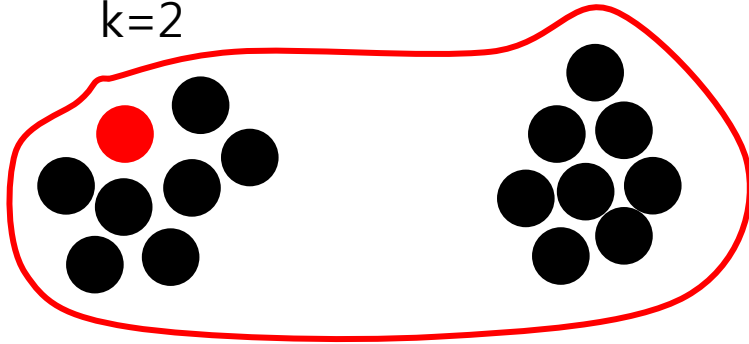
두 번째 중심은 첫 번째 중심에서 가장 먼 것



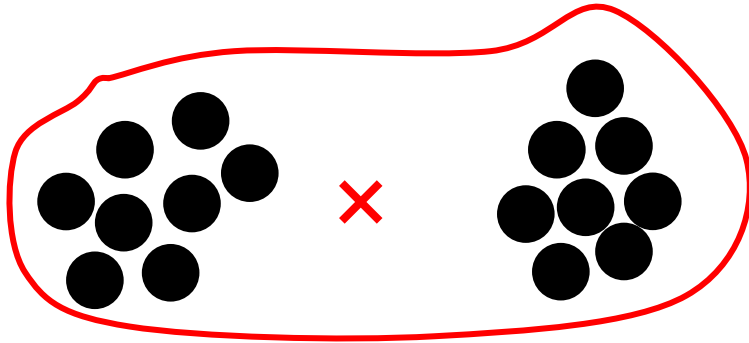
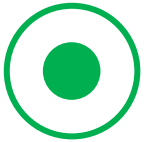
개선된 초기화 방법 #1

32

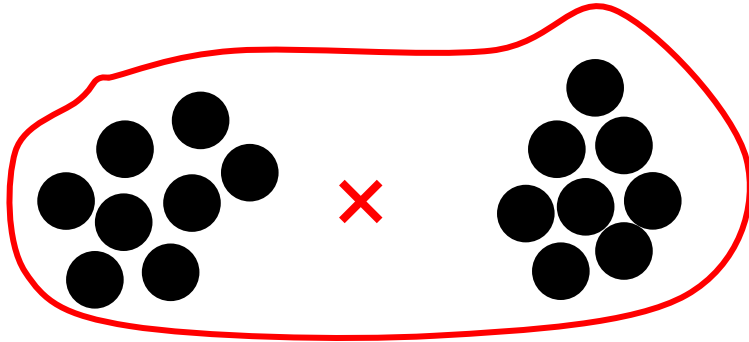
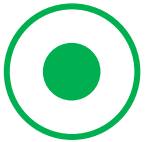
$k=2$



가장 가까운 중심의 클러스터에 할당



중심 위치 갱신



가장 가까운 중심의 클러스터에 할당

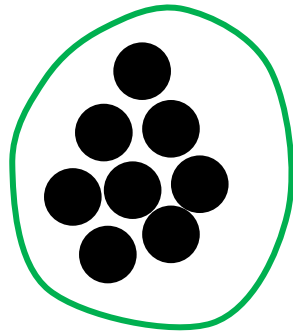
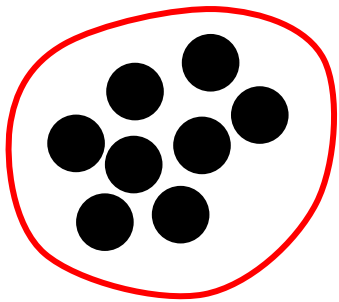


개선된 초기화 방법 #1

33

이렇게 되어야 할 것 같은데

$k=2$



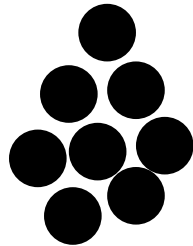
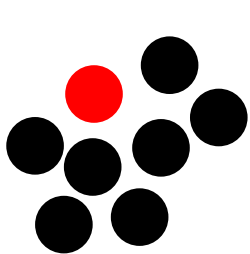
개선된 초기화 방법 #2

34

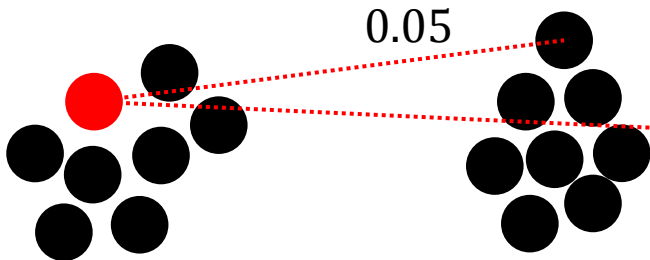
$D(x)$: 데이터 x 에 대해 가장 가까운 중심까지의 거리

$$p(x) = \frac{D(x)^2}{\sum_{x'} D(x')^2} \rightarrow D(x)^2 \text{에 비례}$$

$k=2$



첫 번째 중심은 랜덤으로

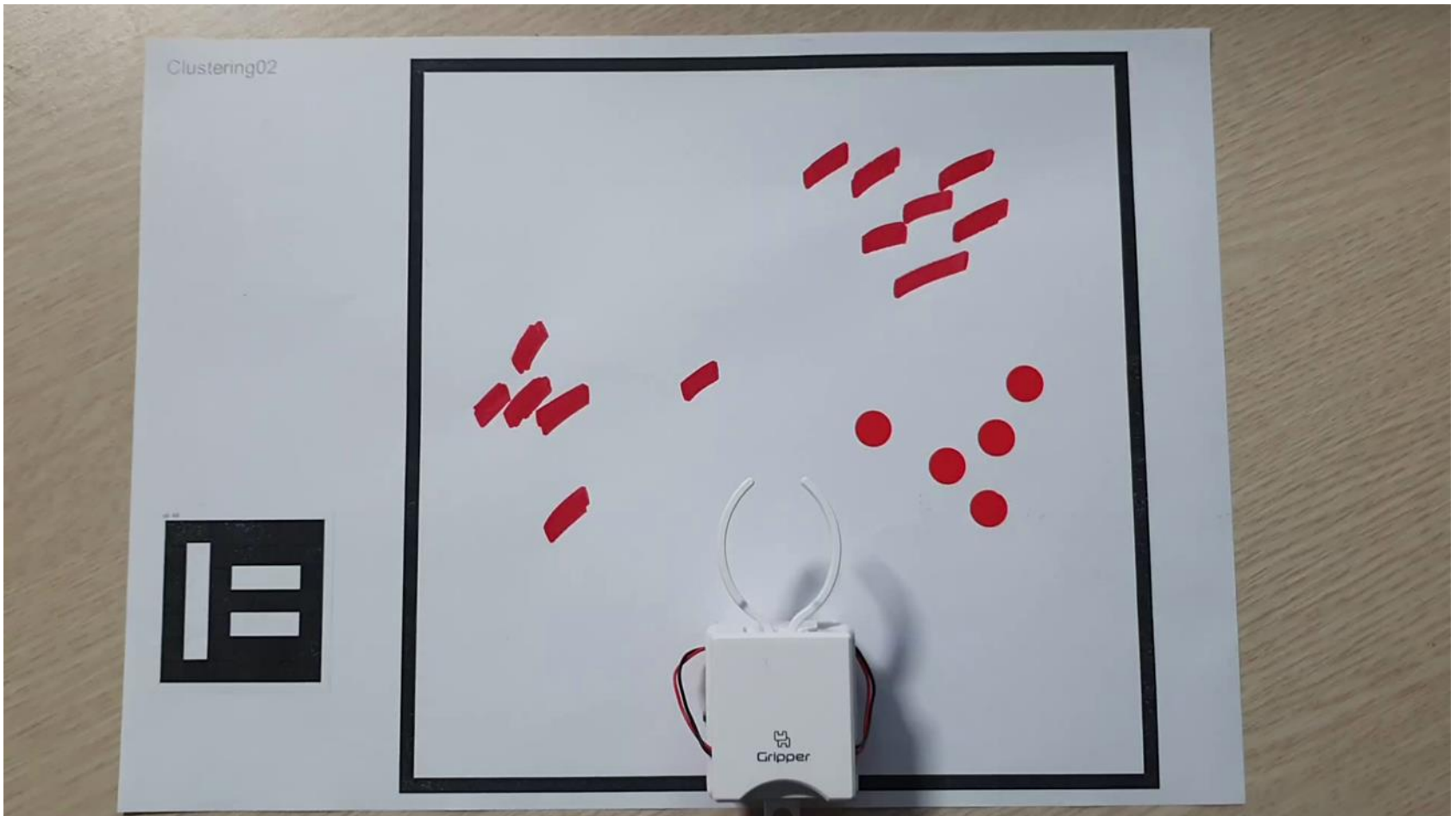


두 번째 중심은 $p(x)$ 확률로 랜덤으로 선택

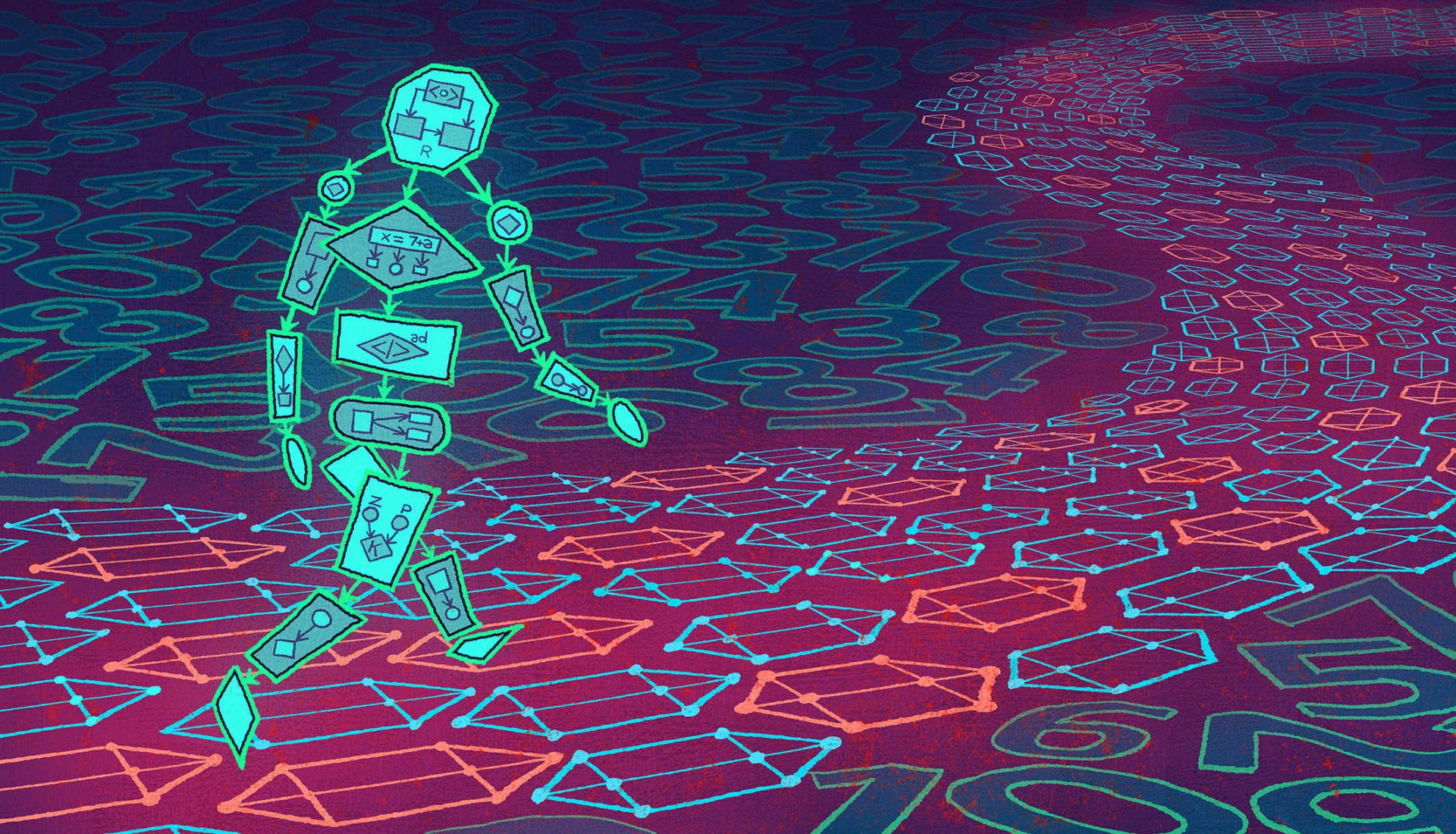
0.1



https://youtu.be/QyqctNf3q_o



길 찾기



규칙을 직접 코딩에서 기계 학습으로

37



시작하기 버튼을 클릭했을 때

2 번 반복하기

말판 앞으로 한 칸 이동하기

말판 왼쪽 ▶ 으로 한 번 돌기

말판 앞으로 한 칸 이동하기

2 번 반복하기

말판 오른쪽 ▶ 으로 한 번 돌기

말판 앞으로 한 칸 이동하기

말판 앞으로 한 칸 이동하기

5 번 반복하기

말판 앞으로 한 칸 이동하기

규칙을 직접 코딩에서 기계 학습으로

38



시작하기 버튼을 클릭했을 때

2

번 반복하기



말판 앞으로 한 칸 이동하기



말판 왼쪽 ▾ 으로 한 번 돌기



말판 앞으로 한 칸 이동하기



2

번 반복하기



말판 오른쪽 ▾ 으로 한 번 돌기



말판 앞으로 한 칸 이동하기



말판 앞으로 한 칸 이동하기



5

번 반복하기



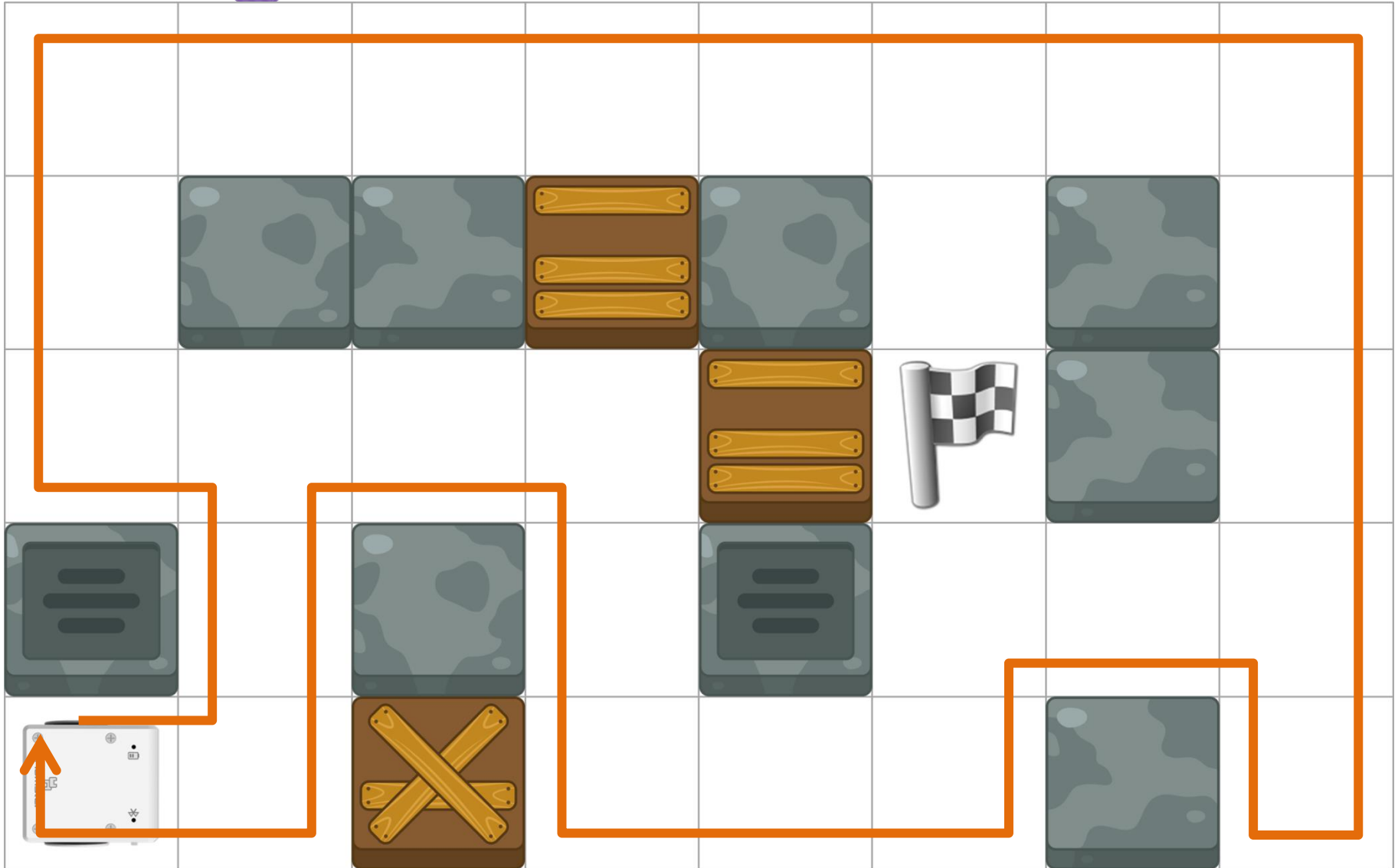
말판 앞으로 한 칸 이동하기



좌수법

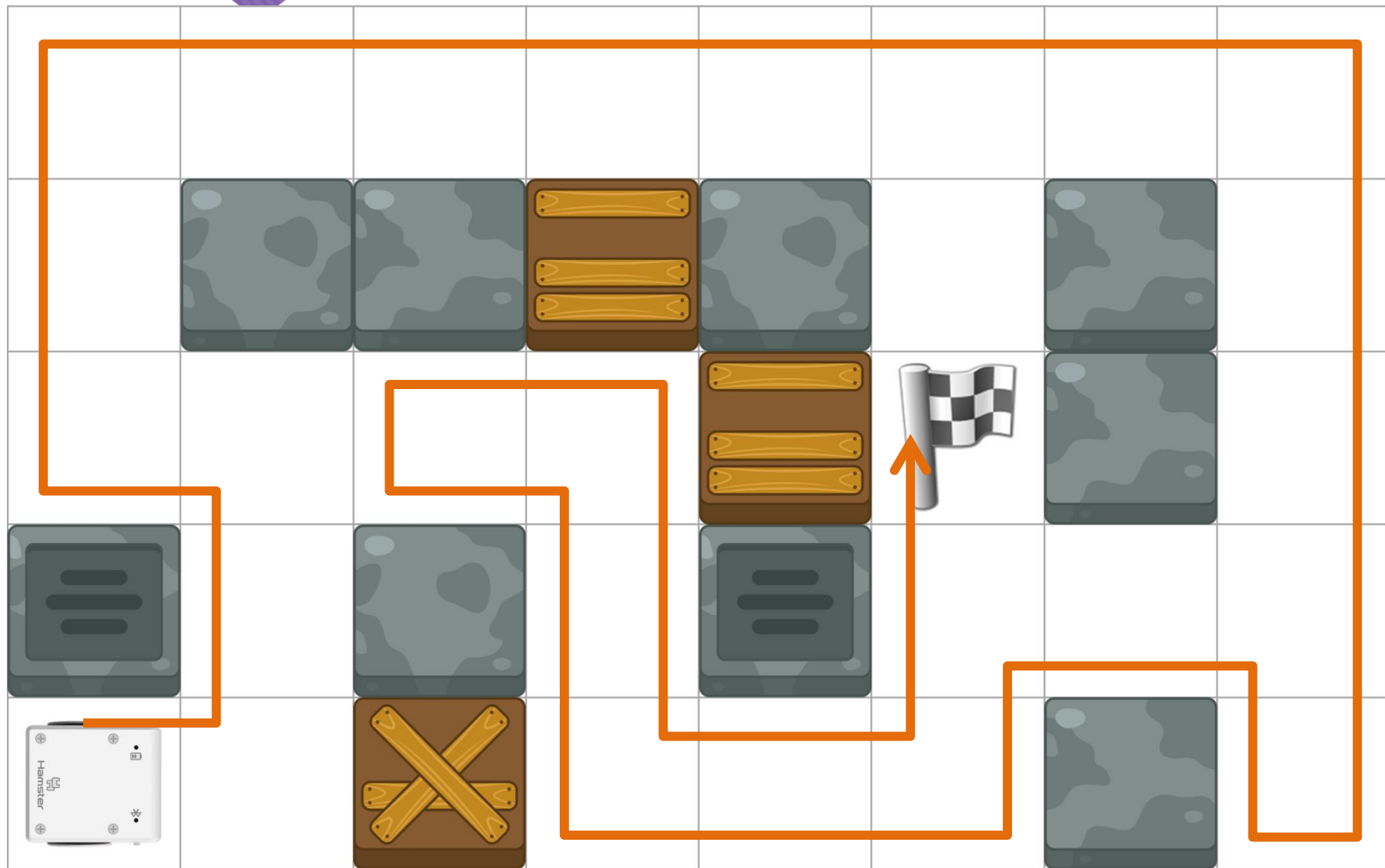
40





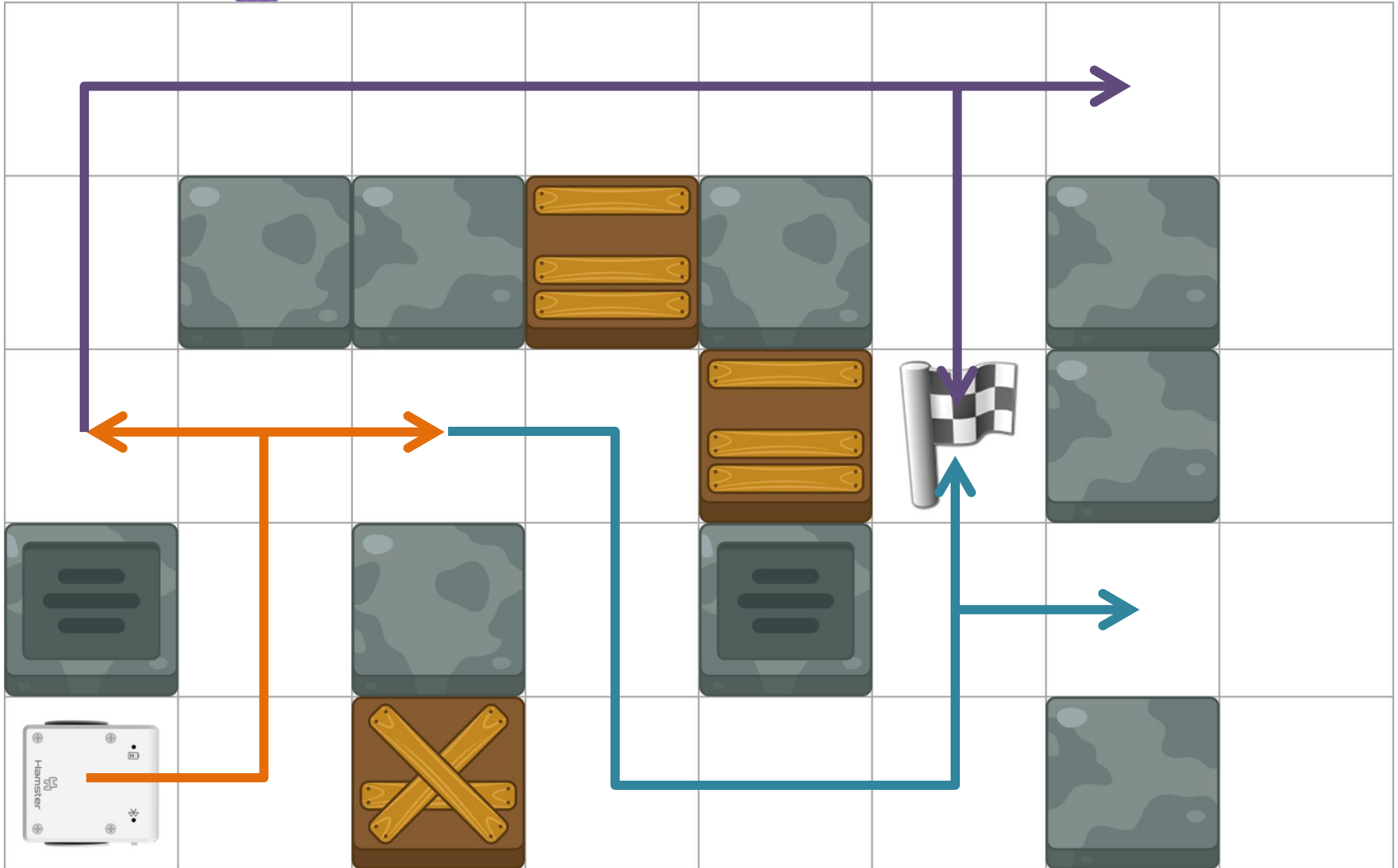
깊이 우선 탐색

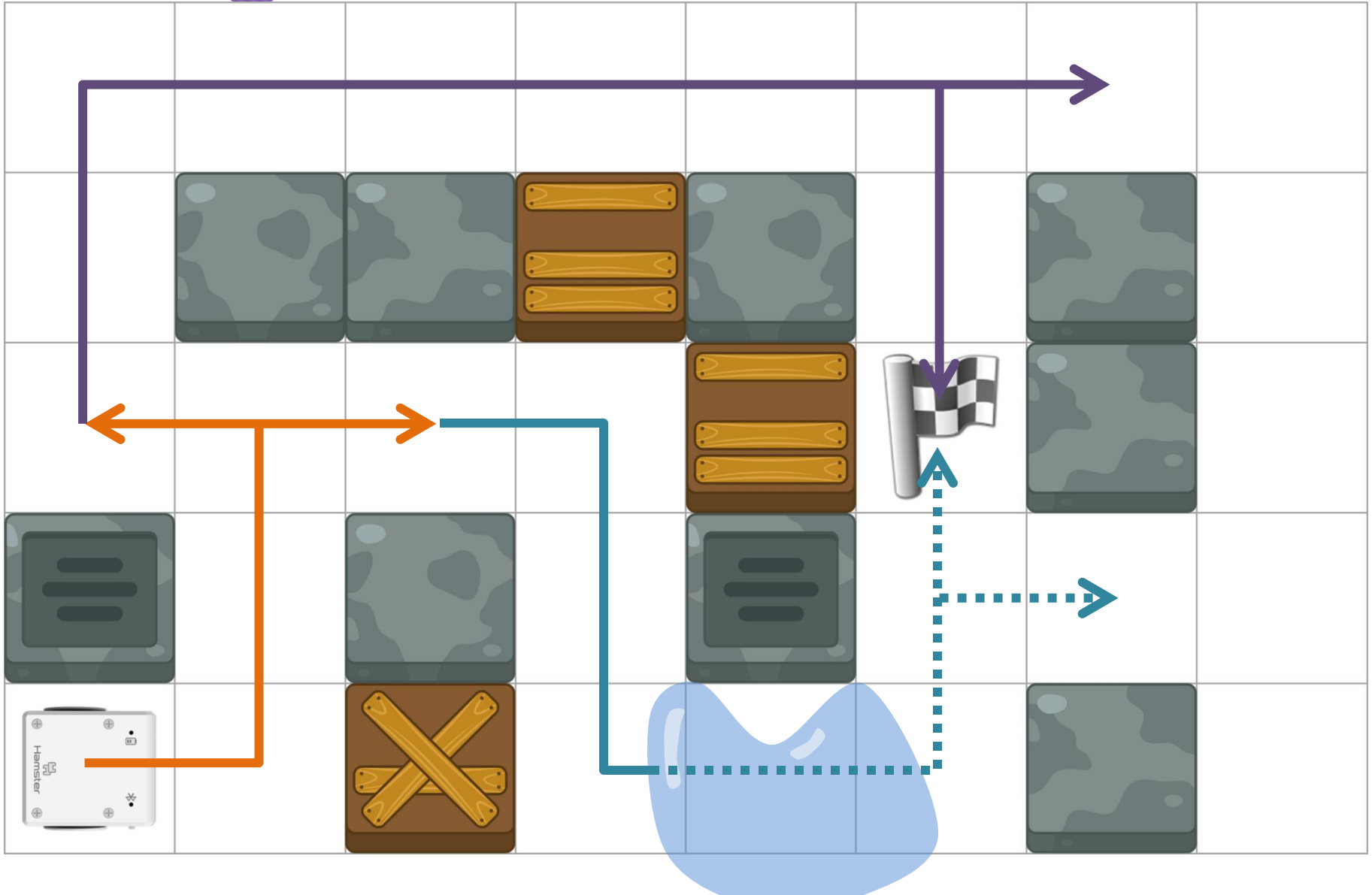
42

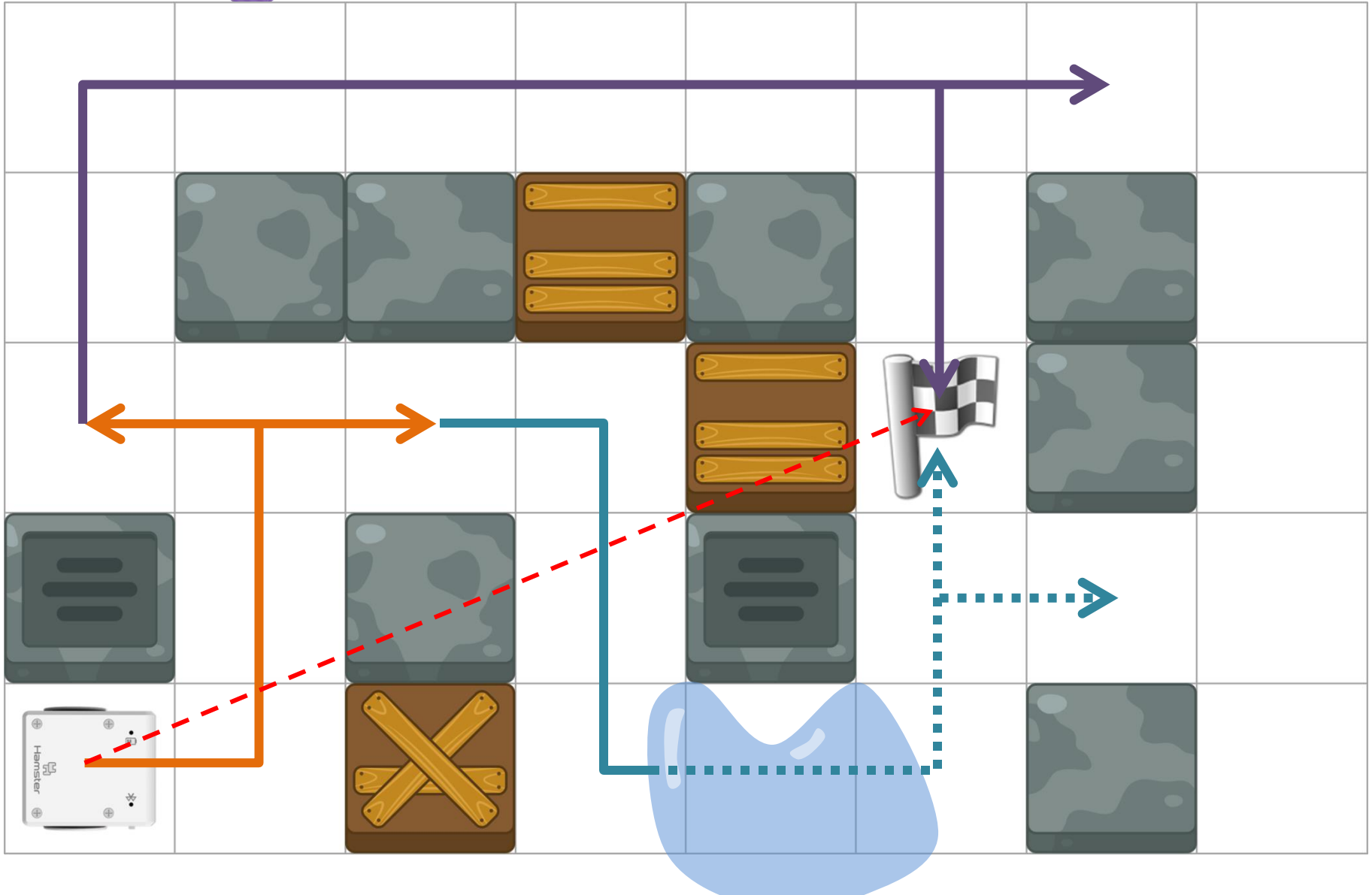


너비 우선 탐색

43

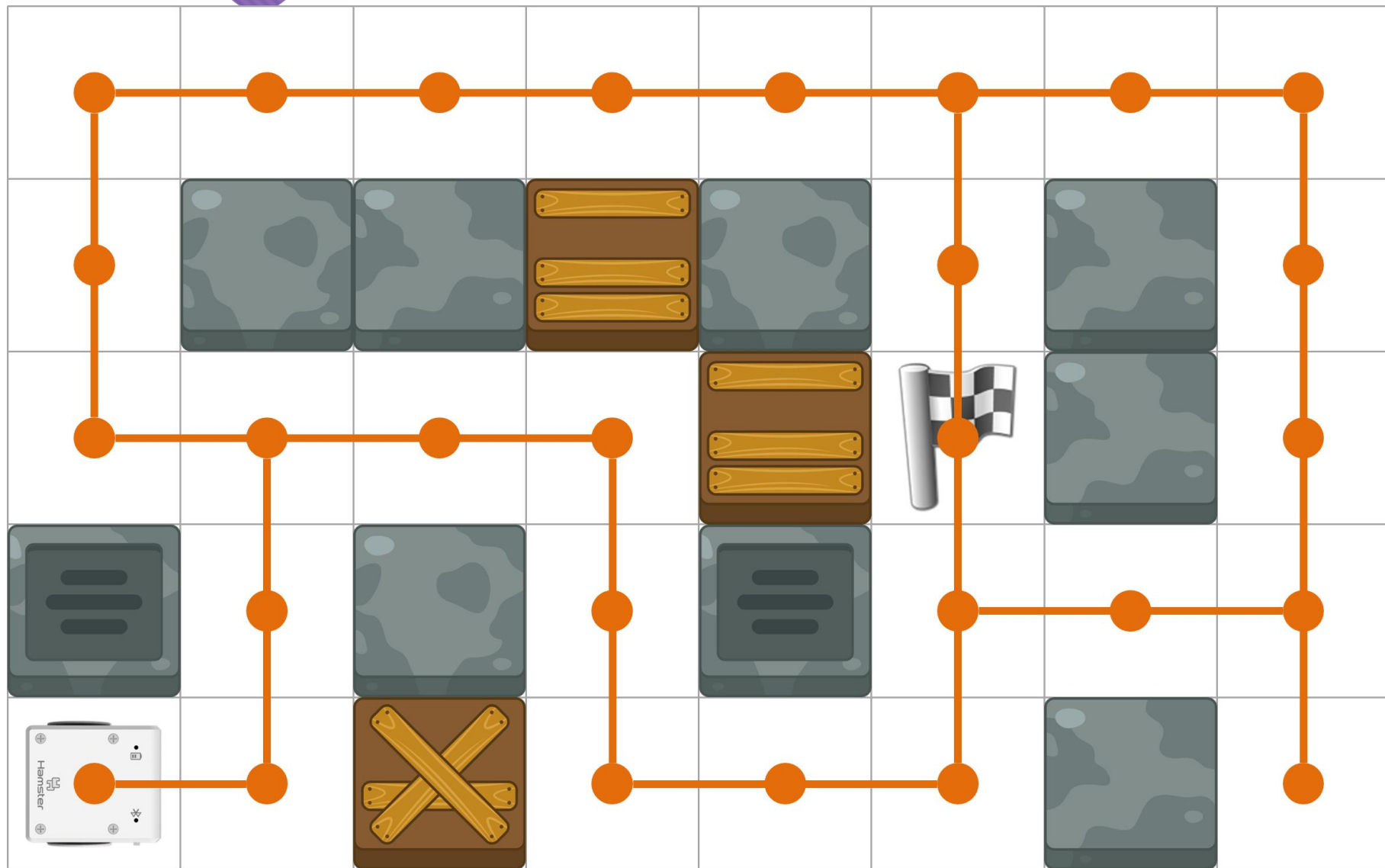






규칙을 직접 코딩에서 기계 학습으로

46



규칙을 직접 코딩에서 기계 학습으로

47



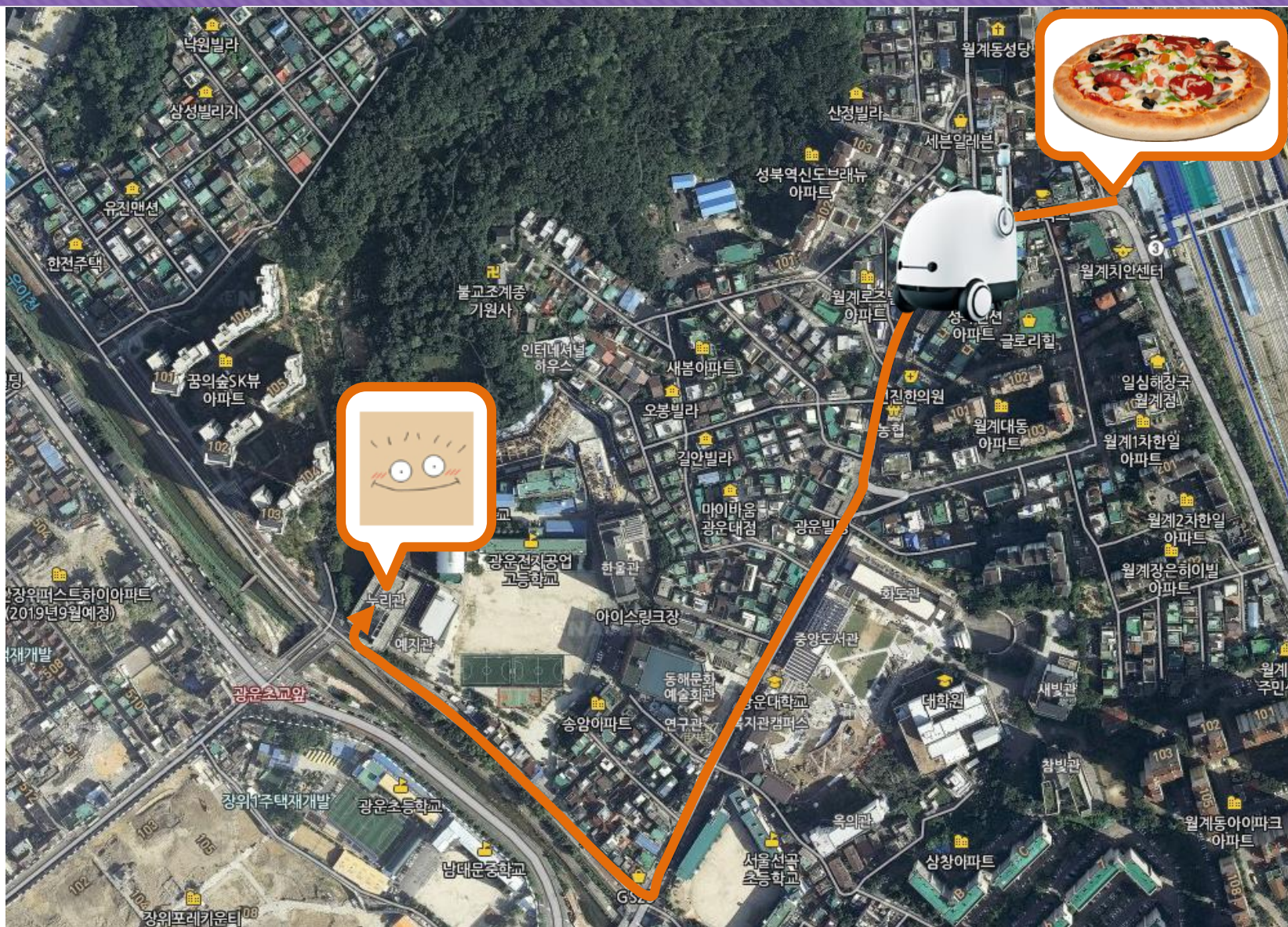
48



49

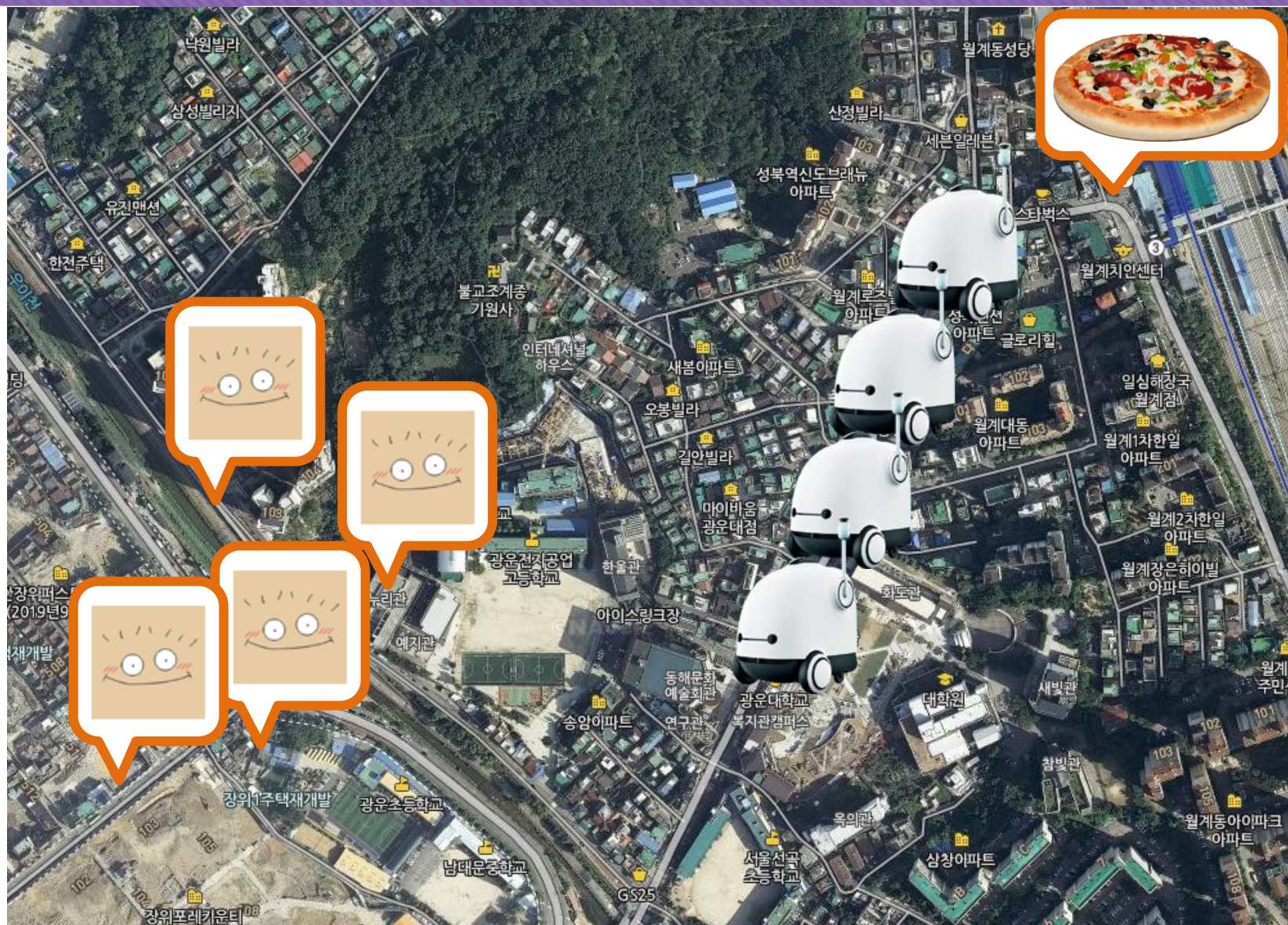


50



규칙을 직접 코딩에서 기계 학습으로

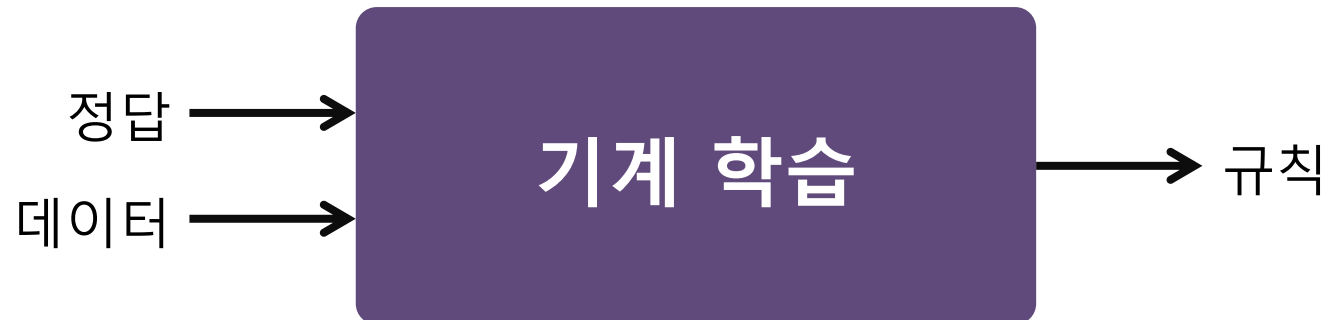
51



규칙을 직접 코딩에서 기계 학습으로

52

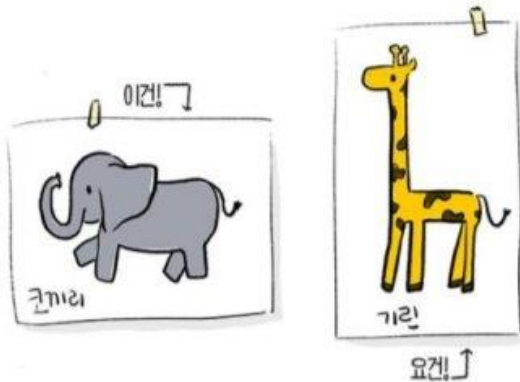
문제가 복잡해 질수록 규칙을 넣어 주기가 어려움



머신 러닝

지도 학습 (Supervised Learning)

문제와 정답을 모두 알려주고
공부시키는 방법



예측, 분류

비지도 학습 (Unsupervised Learning)

답을 가르쳐주지 않고
공부시키는 방법

비지도학습은 답을 가르쳐주지 않고 공부를 시키는거야.



연관 규칙, 군집

강화 학습 (Reinforcement Learning)

보상을 통해
상은 최대화, 벌은 최소화하는
방향으로 행위를 강화하는 학습

강화학습은 일종의 게임 같이 보상해주는거야



보상

초등학교 체험 수준

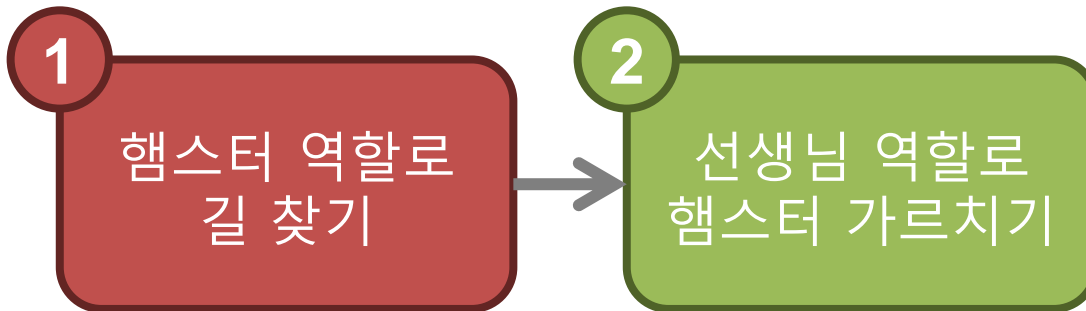


세미
언플러그드

활동 12

강화 학습 역할 놀이

- 로봇이 시행착오를 통해 목표 지점으로 이동하는 학습 과정을 이해할 수 있다.
- 학습을 수행하는 입장(햄스터)과 가르치는 입장의 두 가지 역할을 수행할 수 있다.

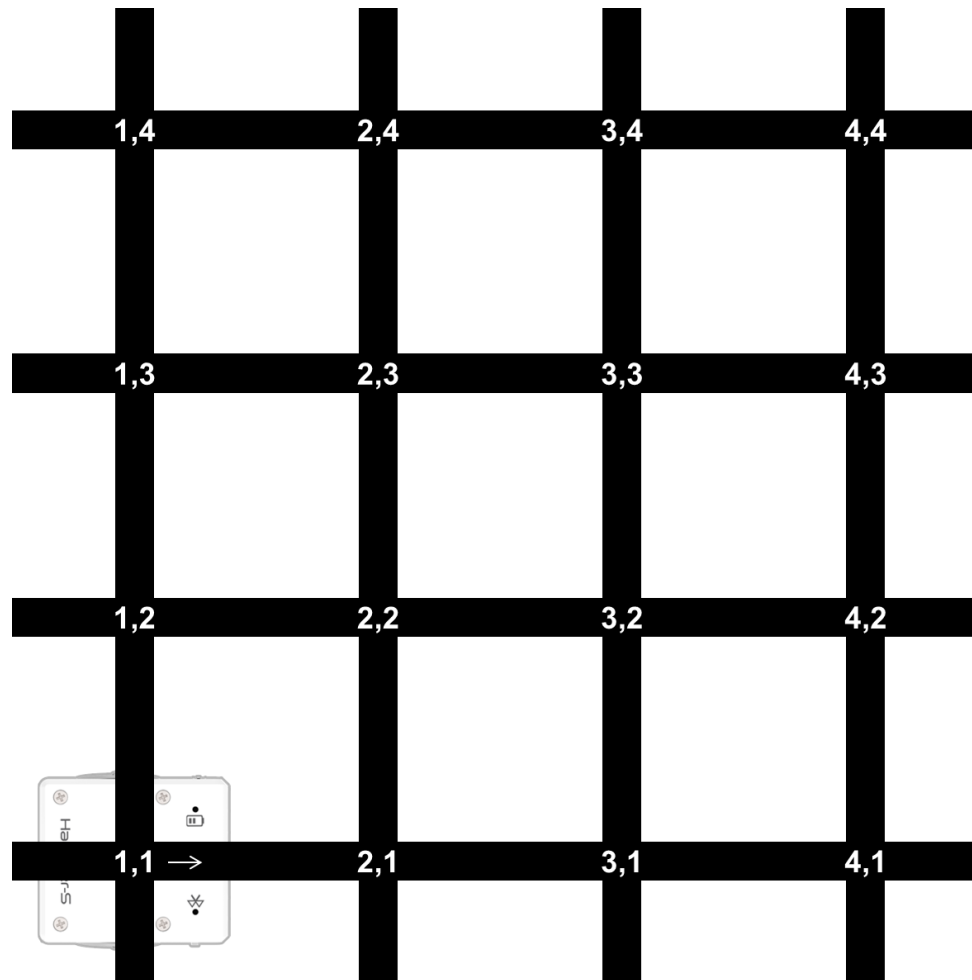


1. 햄스터의 입장

시행 착오를 통해

함정을 피해 목표 지점으로 갑니다.

출발 위치에 로봇의 **방향을 맞추어** 올려 놓습니다.



<http://naver.me/IG Ae3rR9>

The screenshot displays the Playentry web application interface. At the top, a browser address bar shows the URL `playentry.org/ws?lang=ko#!/`. A red arrow points from the URL `http://naver.me/IG Ae3rR9` to the address bar. The application header includes a search bar with the text "200709_작품" and a tab labeled "장면 1". The main workspace features a character animation stage with a blue robot character. The stage coordinates are `X: 234.5, Y: 127.7`. On the left, a sidebar lists various tools: 시각 (Visual), 효율 (Efficiency), 움직임 (Movement), 생김새 (Appearance), 붓 (Brush), 소리 (Sound), 판단 (Judgment), 계산 (Calculation), 자료 (Data), 함수 (Function), 데이터분석 (Data Analysis), 인공지능 (Artificial Intelligence), 확장 (Extension), and 하드웨어 (Hardware). The right sidebar shows a list of events and actions, including "시작하기 버튼을 클릭했을 때" (When the start button is clicked), "카를 눌렀을 때" (When Carl is pressed), "마우스를 클릭했을 때" (When the mouse is clicked), "마우스 클릭을 해제했을 때" (When the mouse click is released), "오브젝트를 클릭했을 때" (When the object is clicked), "오브젝트 클릭을 해제했을 때" (When the object click is released), "대상 없음" (No target), "신호를 받았을 때" (When the signal is received), "신호 보내기" (Send signal), "신호 보내고 기다리기" (Send signal and wait), "장면이 시작되었을 때" (When the scene starts), "장면 1 시작하기" (Start scene 1), and "다음 장면 시작하기" (Start next scene). The bottom left panel shows the character's properties, including X, Y, 크기 (Size), 방향 (Direction), and 이동 방향 (Move direction).

- 함정이 몇 개인지, 함정이 어디에 있는지, 목표 위치가 어디인지 알 수 없습니다.
- 방향키를 눌러 로봇을 조종합니다.
 - 왼쪽 방향키 : 왼쪽으로 90도 회전
 - 오른쪽 방향키: 오른쪽으로 90도 회전
 - 위쪽 방향키: 앞으로 한 칸 이동
- 함정에 빠지면 "삐삐" → 제자리로 돌아갑니다.
- 목표 위치에 도착하면 "도미술" → 제자리로 돌아갑니다.
- 중간에 로봇의 위치가 이상하면 로봇을 손으로 들어 출발 위치에 놓고 "r"키를 눌러 위치 정보를 리셋합니다.
- 함정에 빠지지 않고 목표 위치에 도착하는 것을 연속 3번하면 됩니다.

로봇X

1

로봇Y

1

로봇 방향

오른쪽

문제를 선택하세요.



1

2

3

4

5

2. 선생님의 입장

**채찍과 당근으로
햄스터를 훈련시킵니다.**

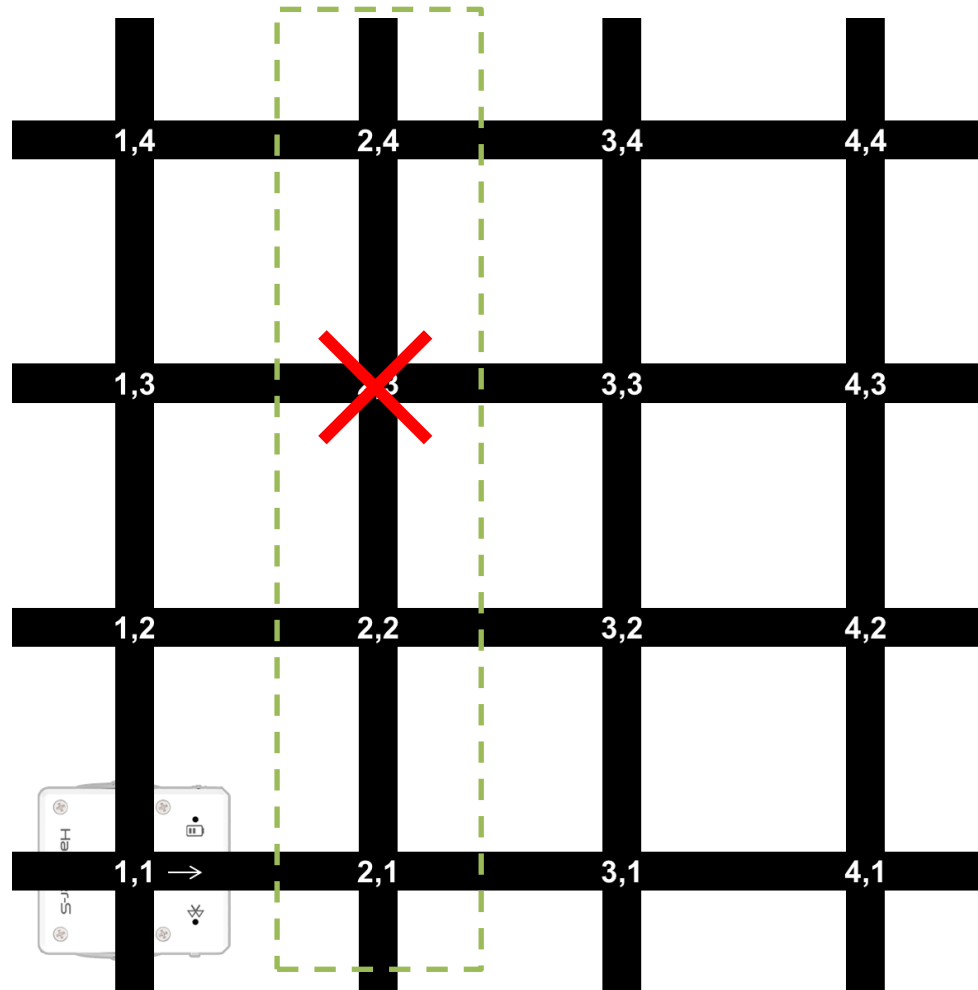
주소 입력

62

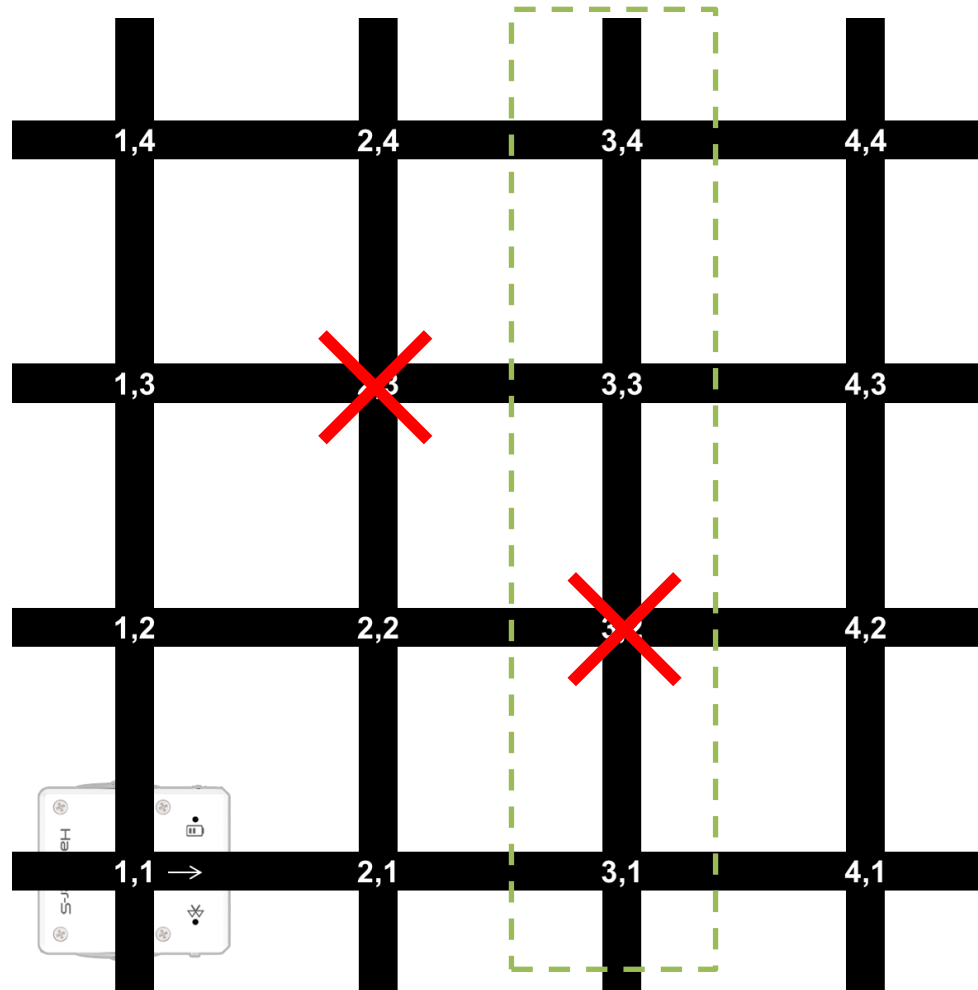
<http://naver.me/5EfiMCTz>

The screenshot displays the Playentry web application interface. At the top, a browser address bar shows the URL `playentry.org/ws?lang=ko#!/`. A red arrow points from the URL `http://naver.me/5EfiMCTz` to the address bar. The main workspace features a character named '엔트리봇' (EntriBot) on a grid. The character's position is indicated as `X: 234.5, Y: 127.7`. The interface includes a left sidebar with various tool categories: 시각 (Visual), 효율 (Efficiency), 움직임 (Movement), 생김새 (Appearance), 붓 (Brush), 소리 (Sound), 판단 (Judgment), 계산 (Calculation), 자료 (Data), 함수 (Function), 데이터분석 (Data Analysis), 인공지능 (Artificial Intelligence), 확장 (Extension), and 하드웨어 (Hardware). The right sidebar shows a list of events and actions, such as '시작하기 버튼을 클릭했을 때' (When the start button is clicked), '카를 눌렀을 때' (When Carl is pressed), '마우스를 클릭했을 때' (When the mouse is clicked), and '시작하기 버튼을 클릭했을 때' (When the start button is clicked). The bottom panel displays the character's properties, including its name '엔트리봇', position `X: 0.0, Y: 0.0`, size `크기: 100.0`, direction `방향(*): 0.0`, and movement direction `이동 방향(*): 90.0`. The '회전방식' (Rotation Method) is set to '회전' (Rotate).

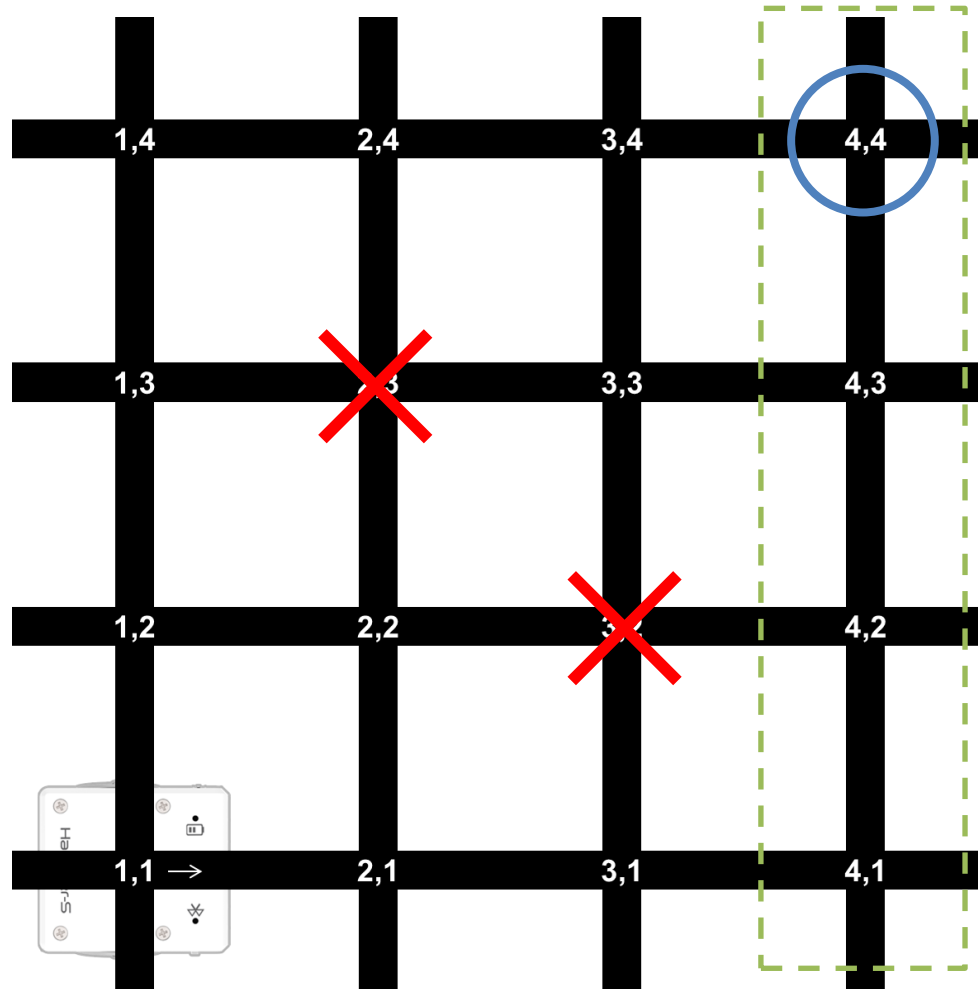
두 번째 열의 4개 칸 중 하나에 함정을 표시합니다.



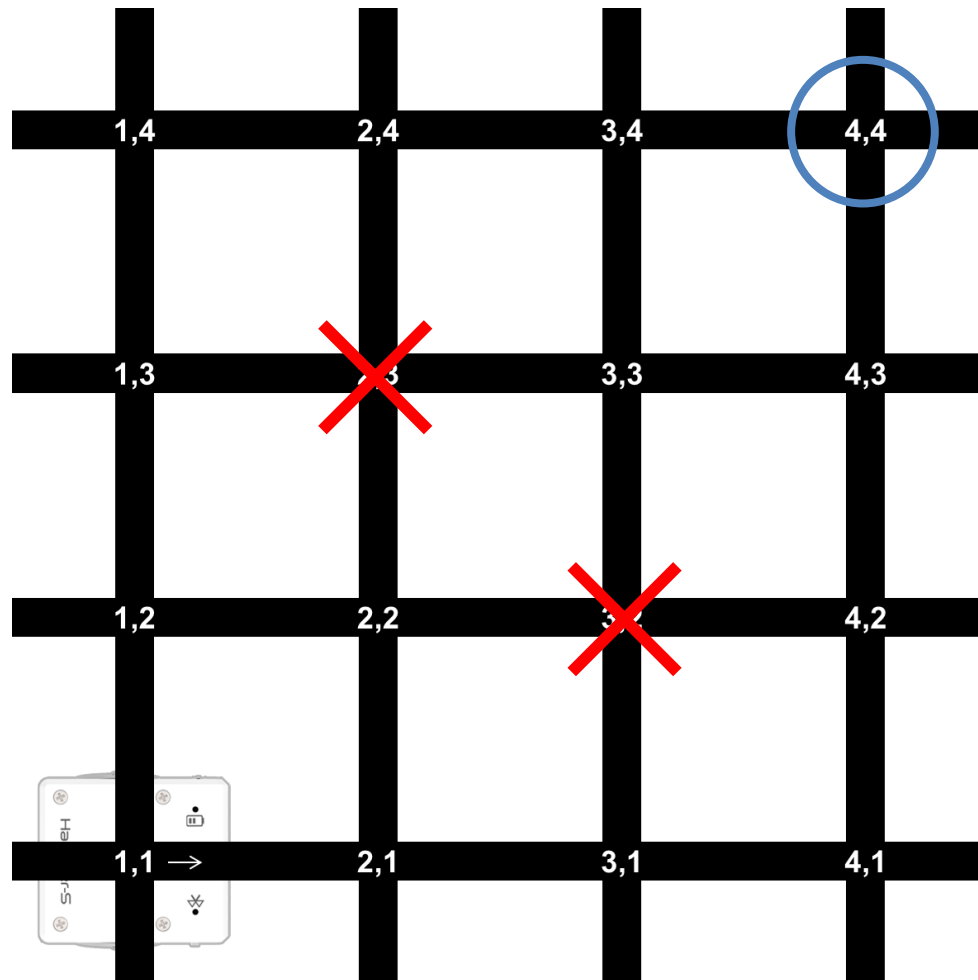
세 번째 열의 4개 칸 중 하나에 함정을 표시합니다.



네 번째 열의 4개 칸 중 하나에 목표 지점을 표시합니다.



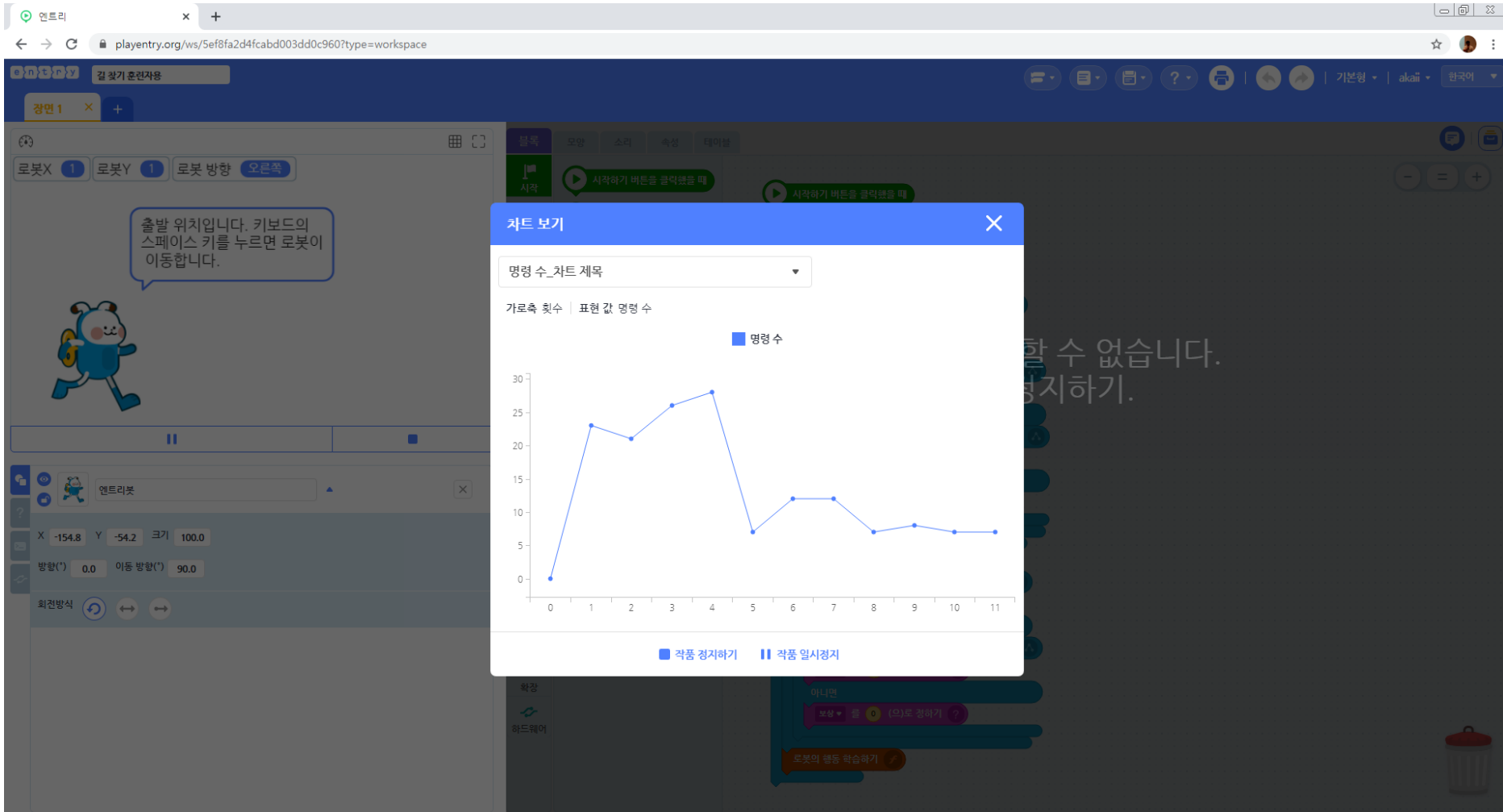
출발 위치에 로봇의 **방향을 맞추어** 올려 놓습니다.



- 스페이스 키를 누를 때마다 로봇이 한 번씩 움직입니다.
- 함정에 빠지면 키보드의 x 키를 누릅니다.
- 목표 위치에 도착하면 키보드의 o 키를 누릅니다.
- 로봇이 출발 위치로 이동하면 다시 반복합니다.
- 로봇이 목표 지점으로 더 잘 주행함을 차트를 통해 관찰합니다.
차트에는 목표 위치에 도착할 때까지 내린 명령 횟수가 표시됩니다.
명령 횟수가 점점 줄어듦을 관찰합니다.
- 중간에 로봇의 위치가 이상하면 로봇을 손으로 들어 출발 위치에 놓고
"r"키를 눌러 위치 정보를 리셋합니다.

성공할 때까지의 누적 명령 수 변화

68

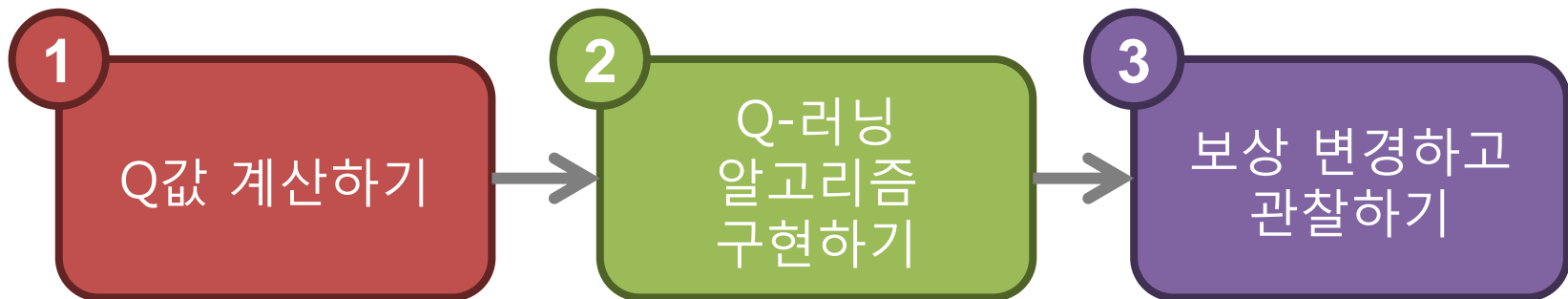


초등학교 동아리 ~ 중학교

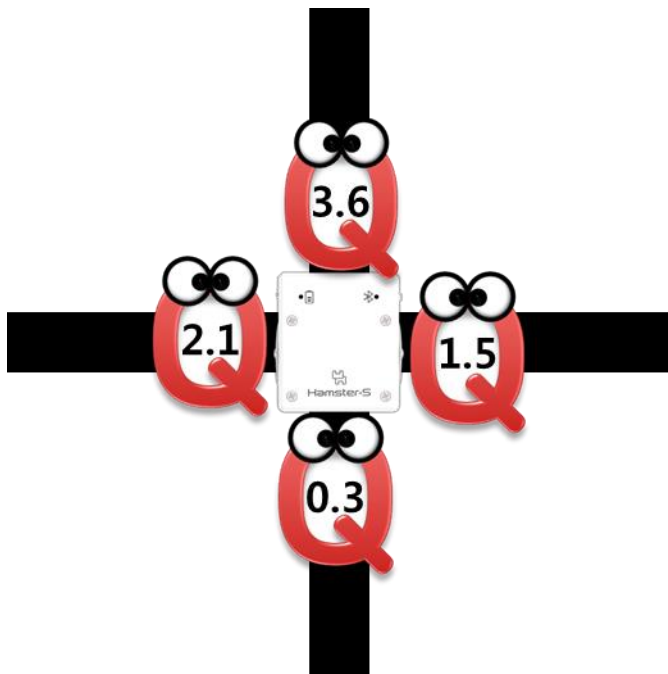
활동 13

강화 학습을 활용한
자율 주행 자동차

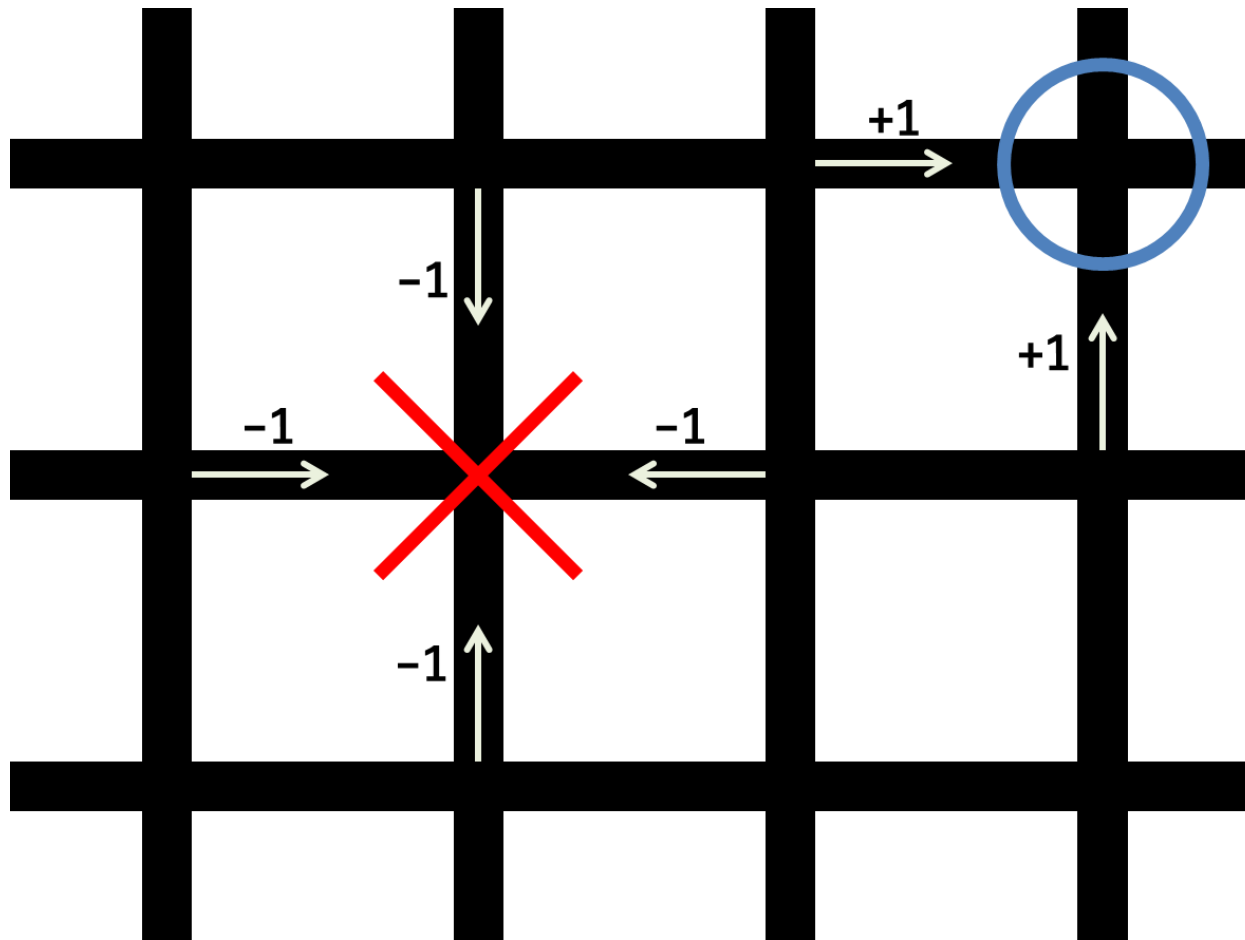
- 경험과 보상을 통해 로봇이 목표 지점으로 이동하는 학습 과정을 이해할 수 있다.
- 강화 학습 기법 중 하나인 Q-러닝 알고리즘을 코드로 구현할 수 있다.



어느 방향으로 가면 좋을지를 네 명의 Q친구들에게 물어 봅니다.
가장 큰 숫자를 얘기하는 방향으로 로봇이 이동하면 됩니다.



Q 친구를 지나갈 때 각 이동(행동)에 대한 보상(점수)



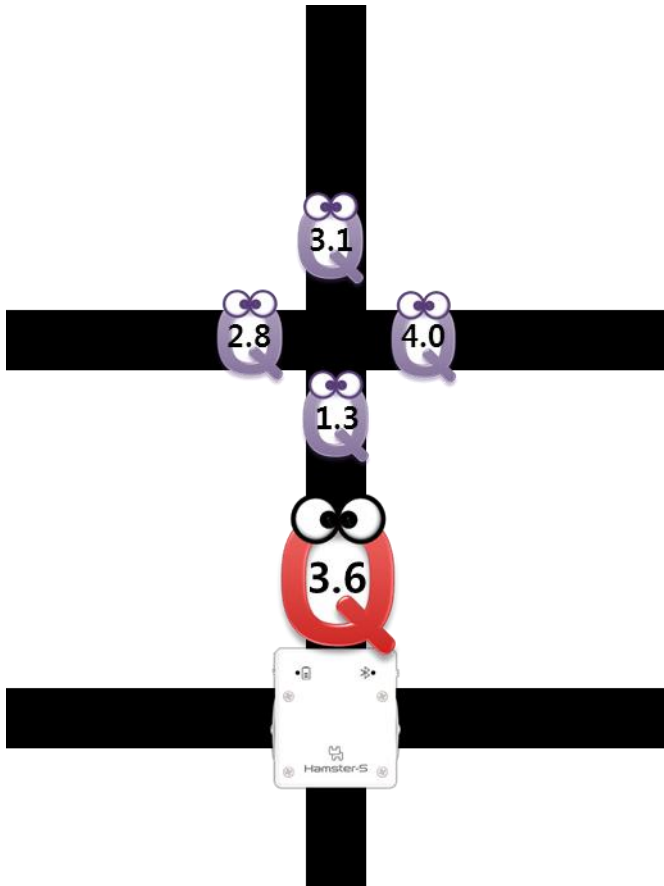
Q 친구들이 말하는 숫자

73

다음 교차로의 Q친구들에게 물어 보고 가장 큰 숫자

$\times 0.9$

+ 자기 점수

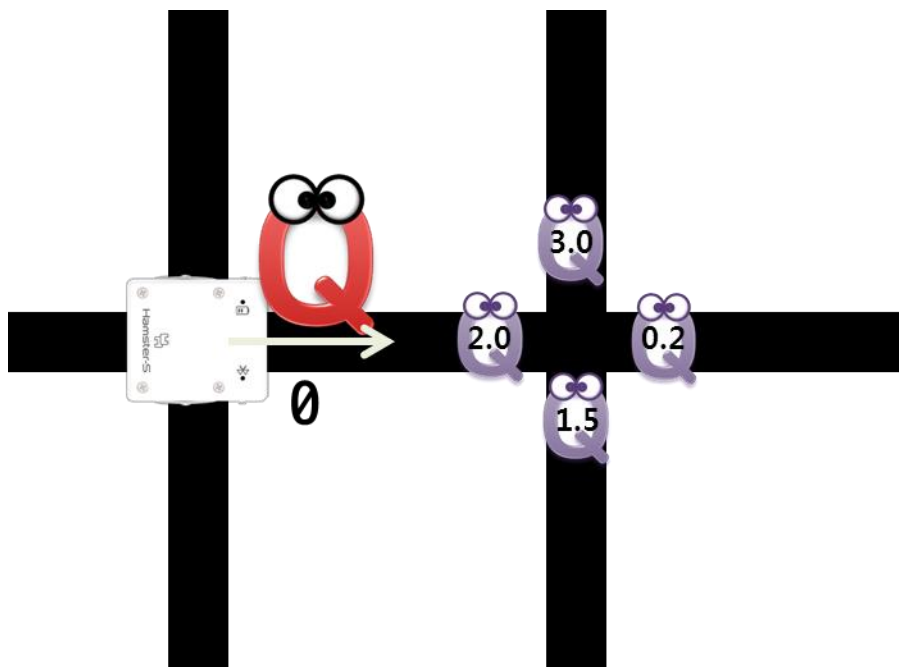


Q값 계산해 보기

74

[현재 교차로에서 Qa 값] \leftarrow [점수 r] + 0.9 x [다음 교차로에서 Qa 값들 중 최댓값]

점수 r	다음 교차로의 보라색 Q값					현재 교차로의 오른쪽 방향 빨간색 Q값
0	왼쪽	오른쪽	위쪽	아래쪽	최댓값	

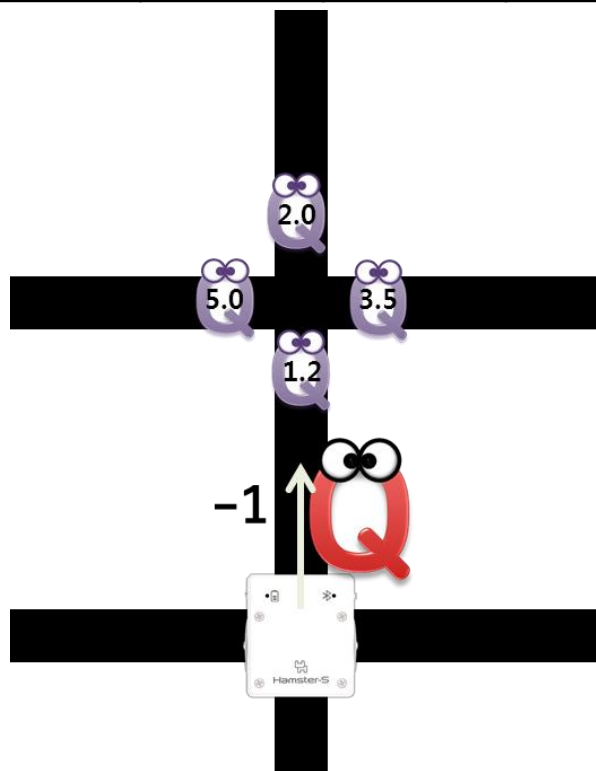


Q값 계산해 보기

75

[현재 교차로에서 Qa 값] \leftarrow [점수 r] + 0.9 x [다음 교차로에서 Qa 값들 중 최댓값]

점수 r	다음 교차로의 보라색 Q값					현재 교차로의 오른쪽 방향 빨간색 Q값
-1	왼쪽	오른쪽	위쪽	아래쪽	최댓값	

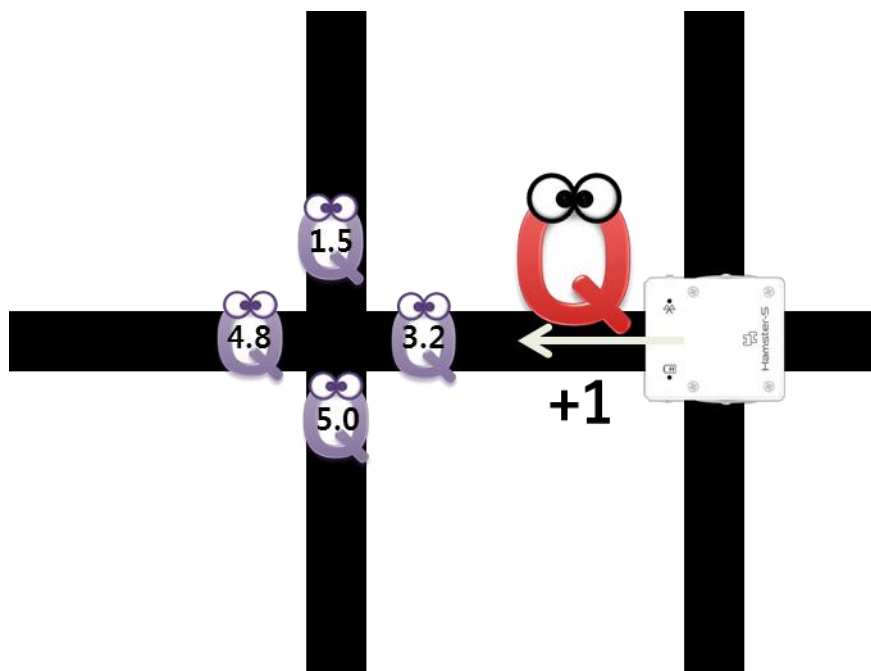


Q값 계산해 보기

76

[현재 교차로에서 Qa 값] \leftarrow [점수 r] + 0.9 x [다음 교차로에서 Qa 값들 중 최댓값]

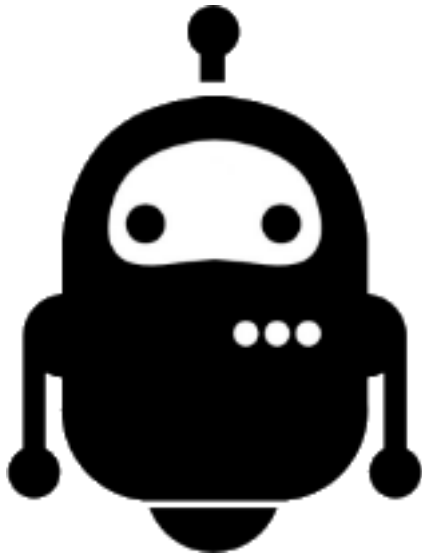
점수 r	다음 교차로의 보라색 Q값					현재 교차로의 오른쪽 방향 빨간색 Q값
1	왼쪽	오른쪽	위쪽	아래쪽	최댓값	



선생님을 위한 보충 설명



에이전트 (agent)



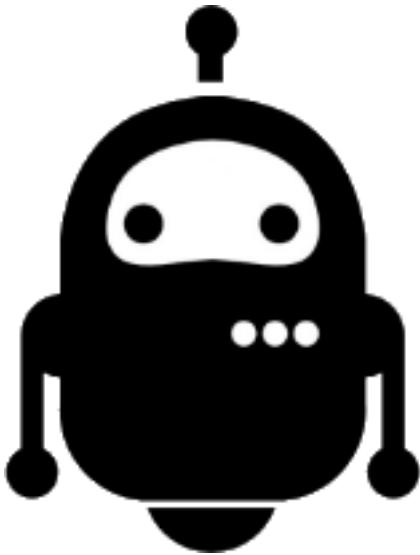
환경 (environment)



환경을 인지하고 그에 대응하는 행동을 하는
컴퓨터 프로그램

- 인터넷 봇 (웹 크롤러)
- 게임 봇 (인공지능 플레이어)
- 챗봇

에이전트 (agent)



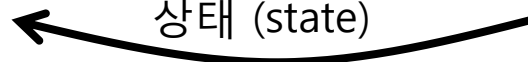
환경 (environment)



행동 (action)



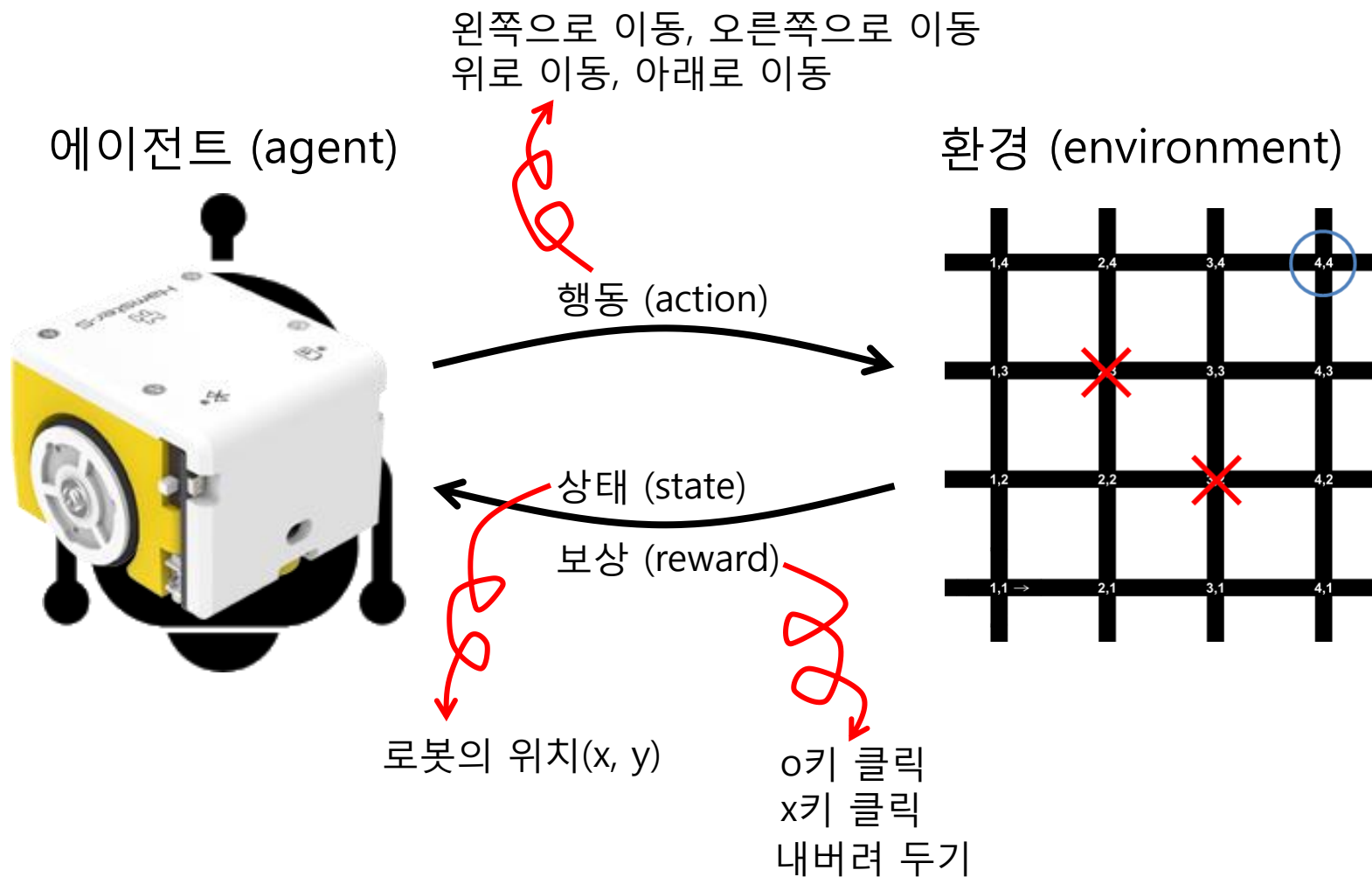
상태 (state)



보상 (reward)

환경을 인지하고 그에 대응하는 행동을 하는
컴퓨터 프로그램

- 인터넷 봇 (웹 크롤러)
- 게임 봇 (인공지능 플레이어)
- 챗봇



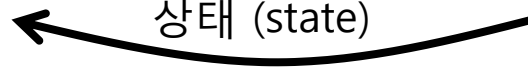
에이전트 (agent)



행동 (action)

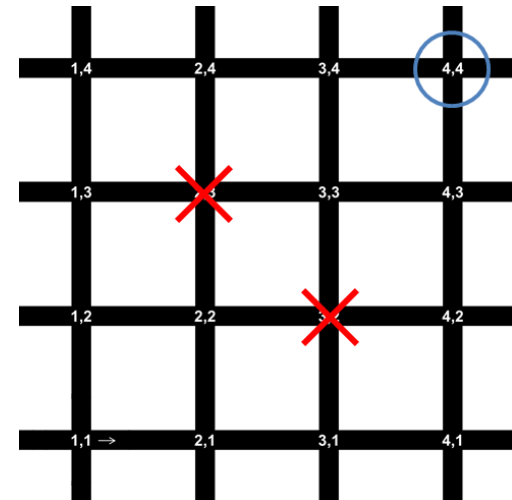


상태 (state)



보상 (reward)

환경 (environment)

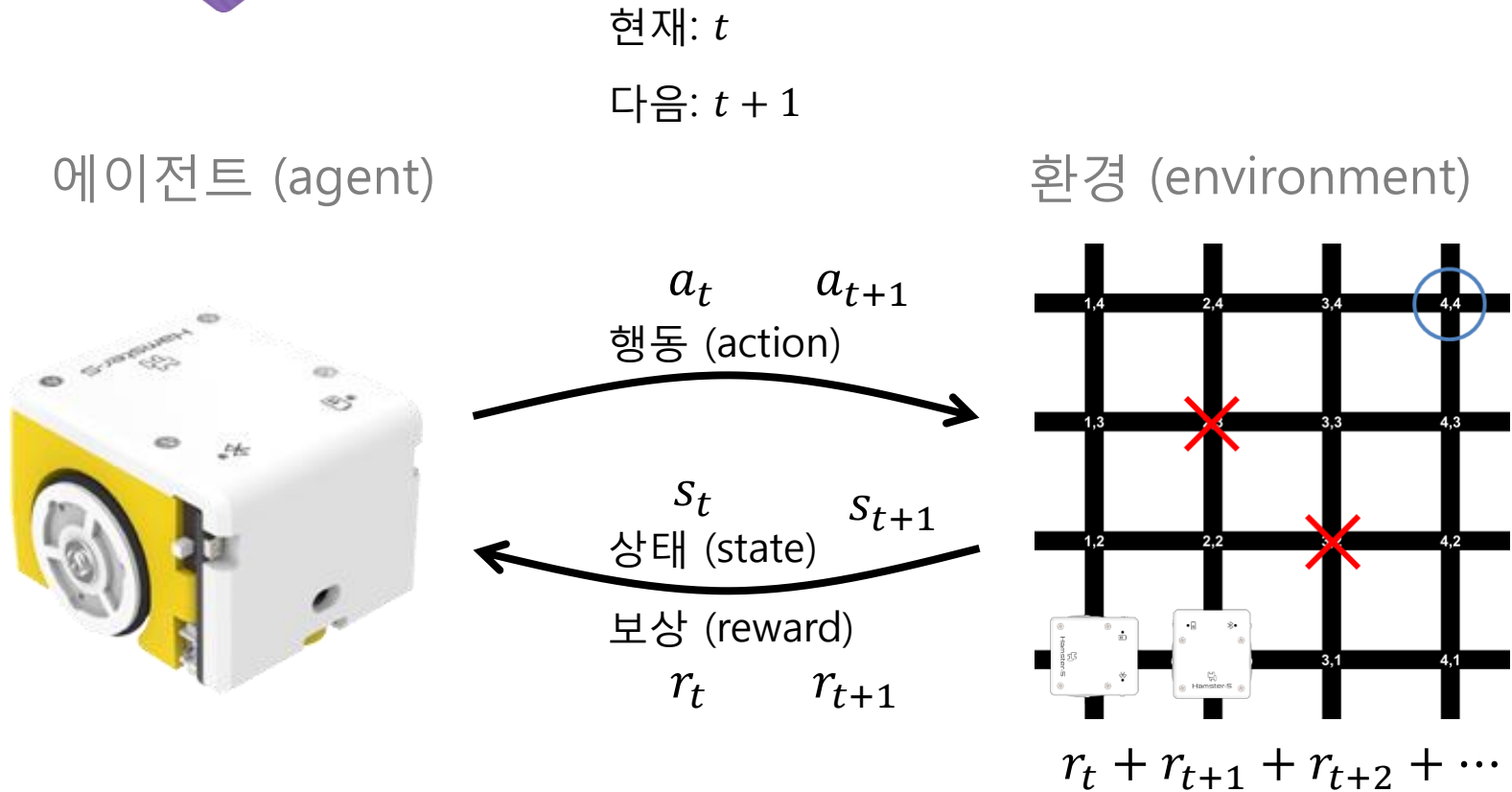


목적 : 현재부터 미래까지의 모든 보상의 합을 최대화

학습의 결과 : 각 상태에서 취해야 할 행동 → 정책 (policy)

로봇의 위치

왼쪽으로, 오른쪽으로
위로, 아래로



목적 : 현재부터 미래까지의 모든 보상의 합을 최대화

학습의 결과 : 각 상태에서 취해야 할 행동 \rightarrow 정책 (policy)

로봇의 위치

왼쪽으로, 오른쪽으로
위로, 아래로

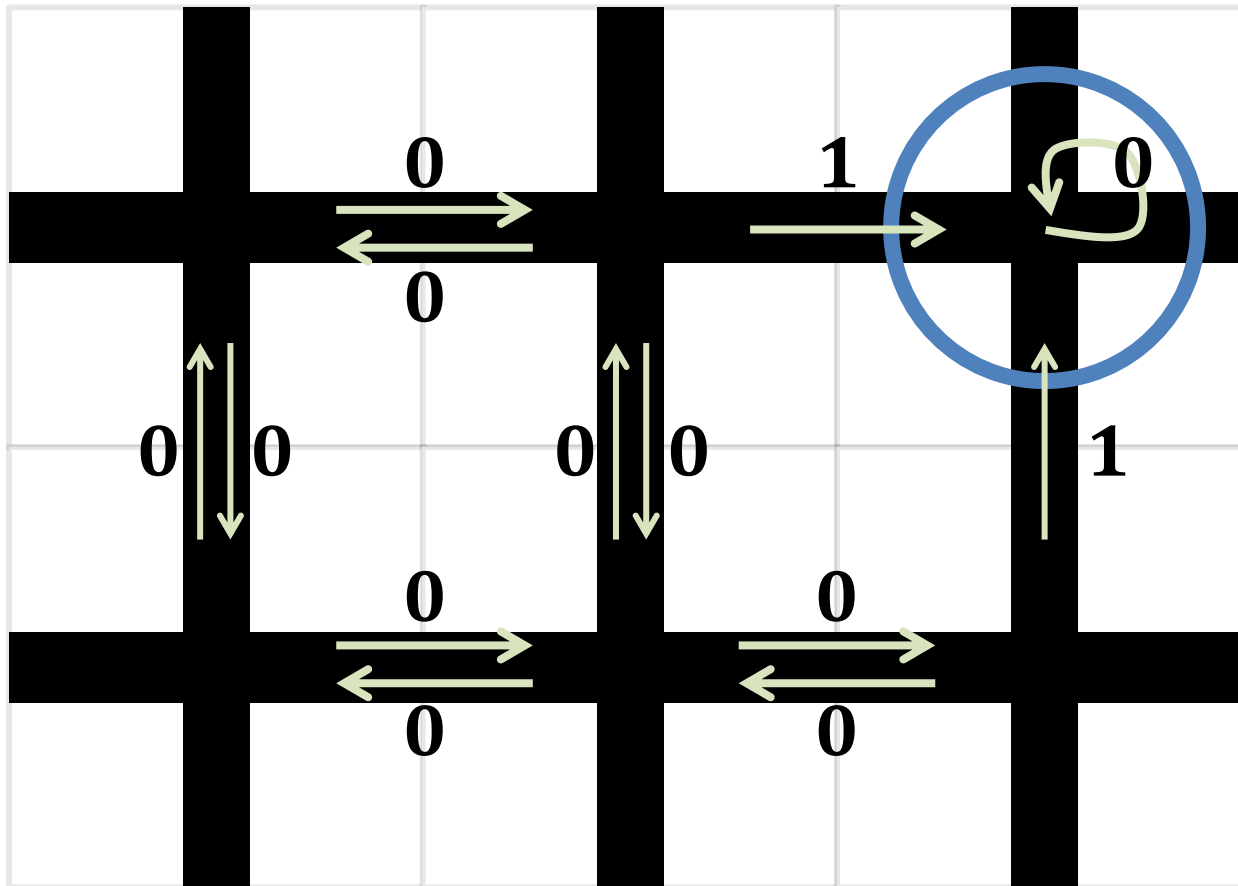
각 상태에서의 행동에 대한 보상

84

상태 (state) : 로봇의 위치 → 회색 테두리의 사각형 칸

행동 (action) : 연두색 화살표

보상 (reward) : 화살표 옆의 숫자 (0 또는 1)

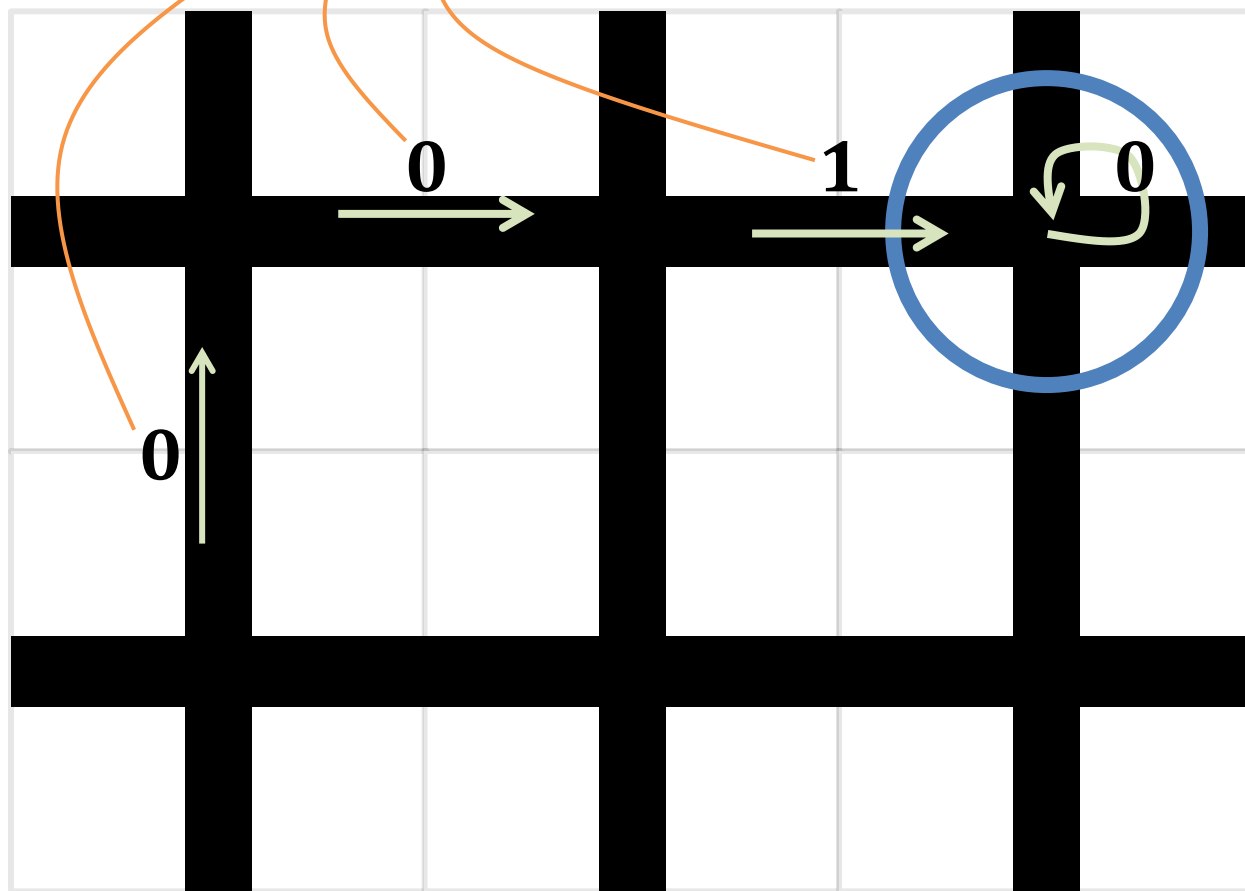


누적 보상

현재부터 미래까지의 모든 보상의 합
 $r_t + r_{t+1} + r_{t+2} + \dots$

85

누적 보상 : $0 + 0 + 1 = 1$

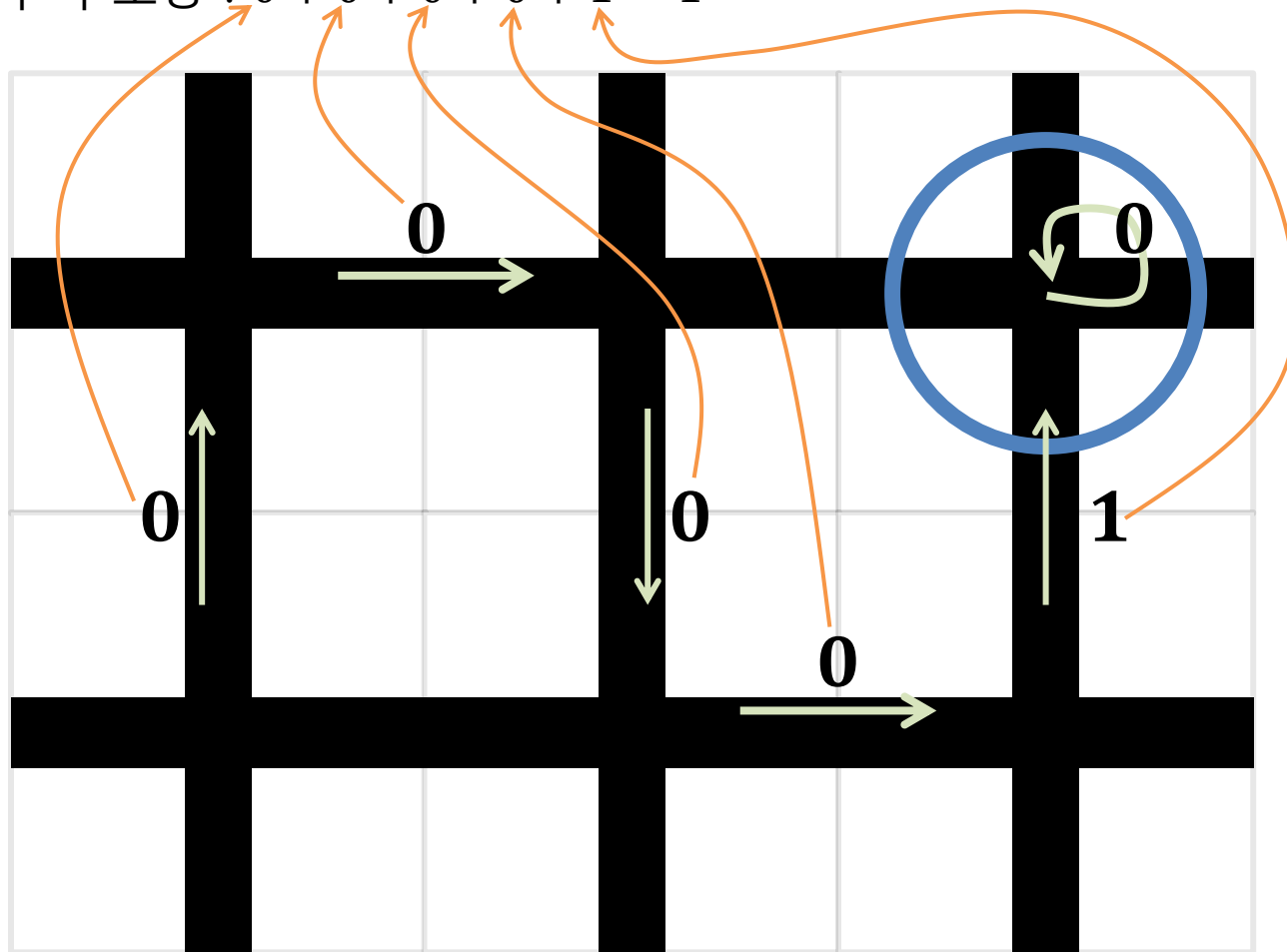


누적 보상

현재부터 미래까지의 모든 보상의 합
 $r_t + r_{t+1} + r_{t+2} + \dots$

86

누적 보상 : $0 + 0 + 0 + 0 + 1 = 1$



$$r_t + r_{t+1} + r_{t+2} + \dots$$



$$r_t + (0.9)r_{t+1} + (0.9)^2 r_{t+2} + (0.9)^3 r_{t+3} + \dots$$

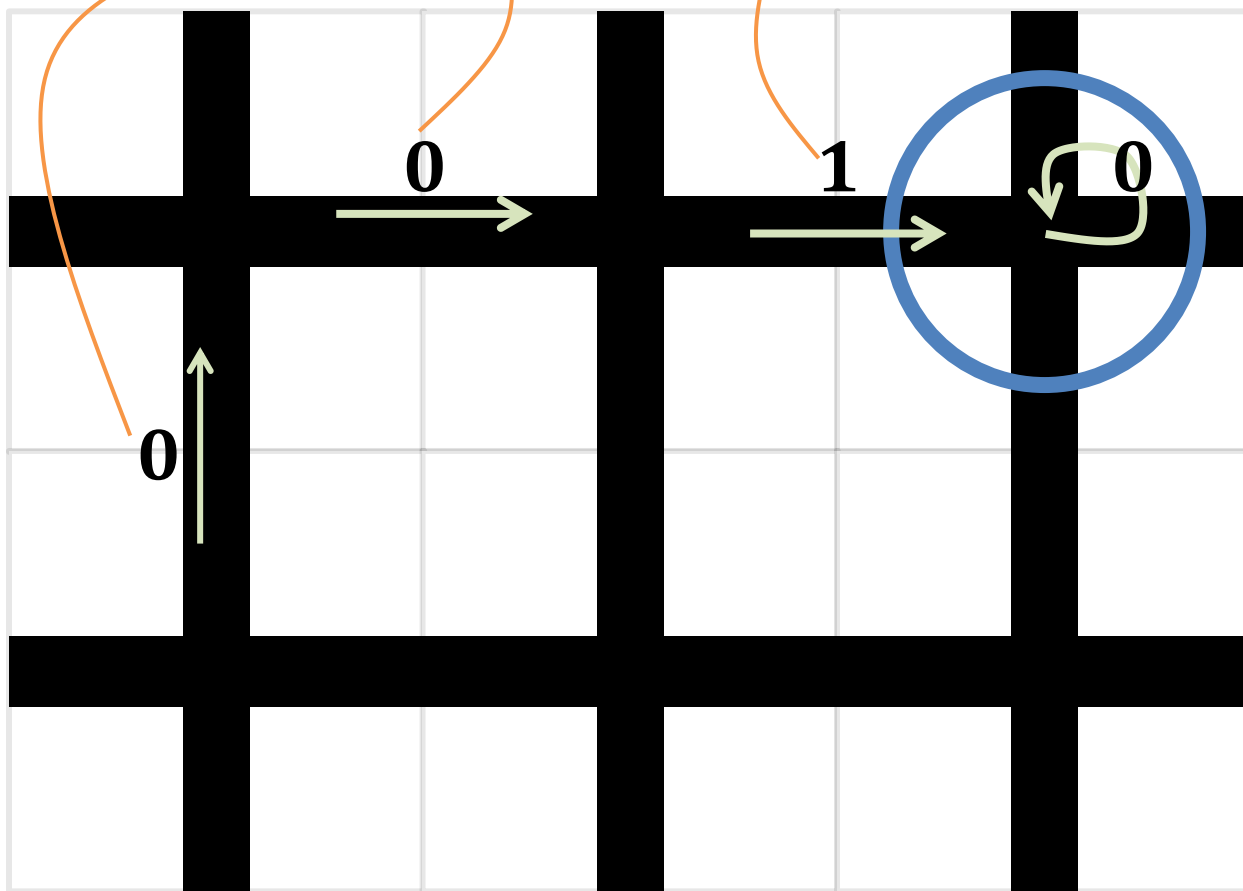
누적 보상

현재부터 미래까지의 모든 보상의 합

$$r_t + (0.9)r_{t+1} + (0.9)^2r_{t+2} + (0.9)^3r_{t+3} + \dots$$

88

누적 보상 : $0 + (0.9) \times 0 + (0.9)^2 \times 1 = 0.81$



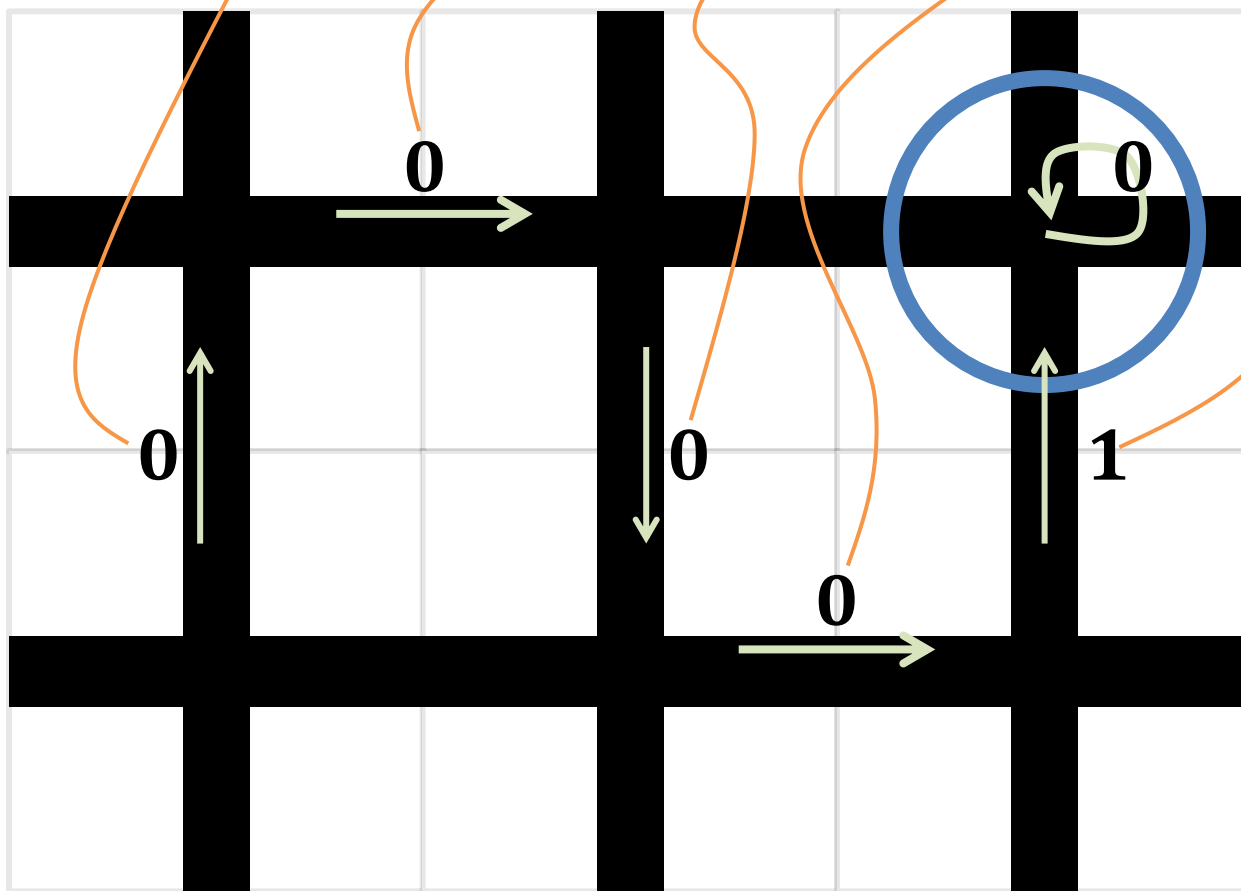
누적 보상

현재부터 미래까지의 모든 보상의 합

$$r_t + (0.9)r_{t+1} + (0.9)^2r_{t+2} + (0.9)^3r_{t+3} + \dots$$

89

누적 보상 : $0 + (0.9) \times 0 + (0.9)^2 \times 0 + (0.9)^3 \times 0 + (0.9)^4 \times 1 = 0.6561$



개미는 정말 짧은 길을 알아낼 수 있을까?

90

<https://youtu.be/5E32W45TuMk>



$$r_t + r_{t+1} + r_{t+2} + \dots$$



$$r_t + (0.9)r_{t+1} + (0.9)^2 r_{t+2} + (0.9)^3 r_{t+3} + \dots$$

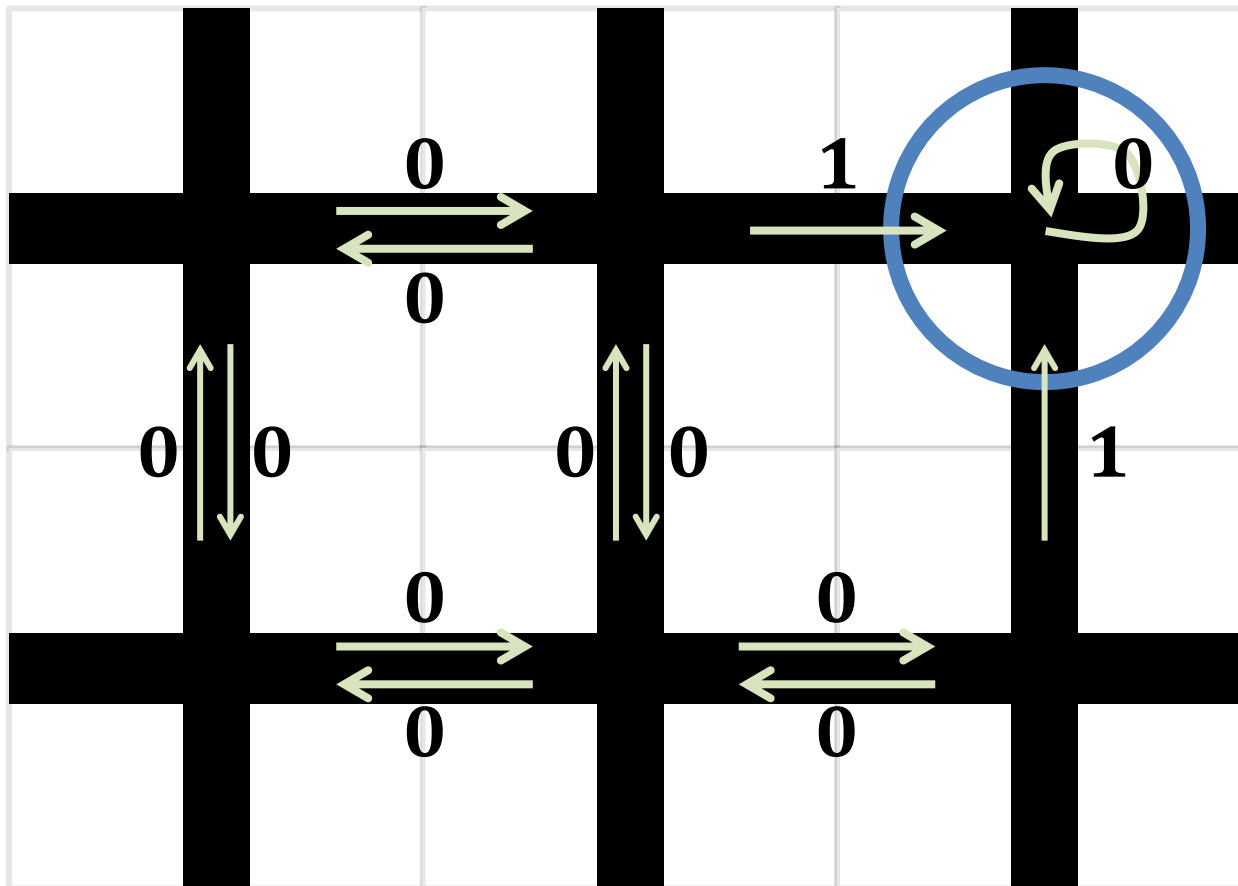


γ : 할인율(discount factor), $0 \leq \gamma < 1$

각 상태에서 취해야 할 행동

누적 보상이 최대가 되도록 하는 정책을 구하라?

→ 미래의 상태 변화, 보상을 모두 알고 있어야 함



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


0.9

모든 상태에서의 모든 행동을 계속 경험하면
 $\hat{Q}(s, a) \rightarrow Q(s, a)$ 로 수렴

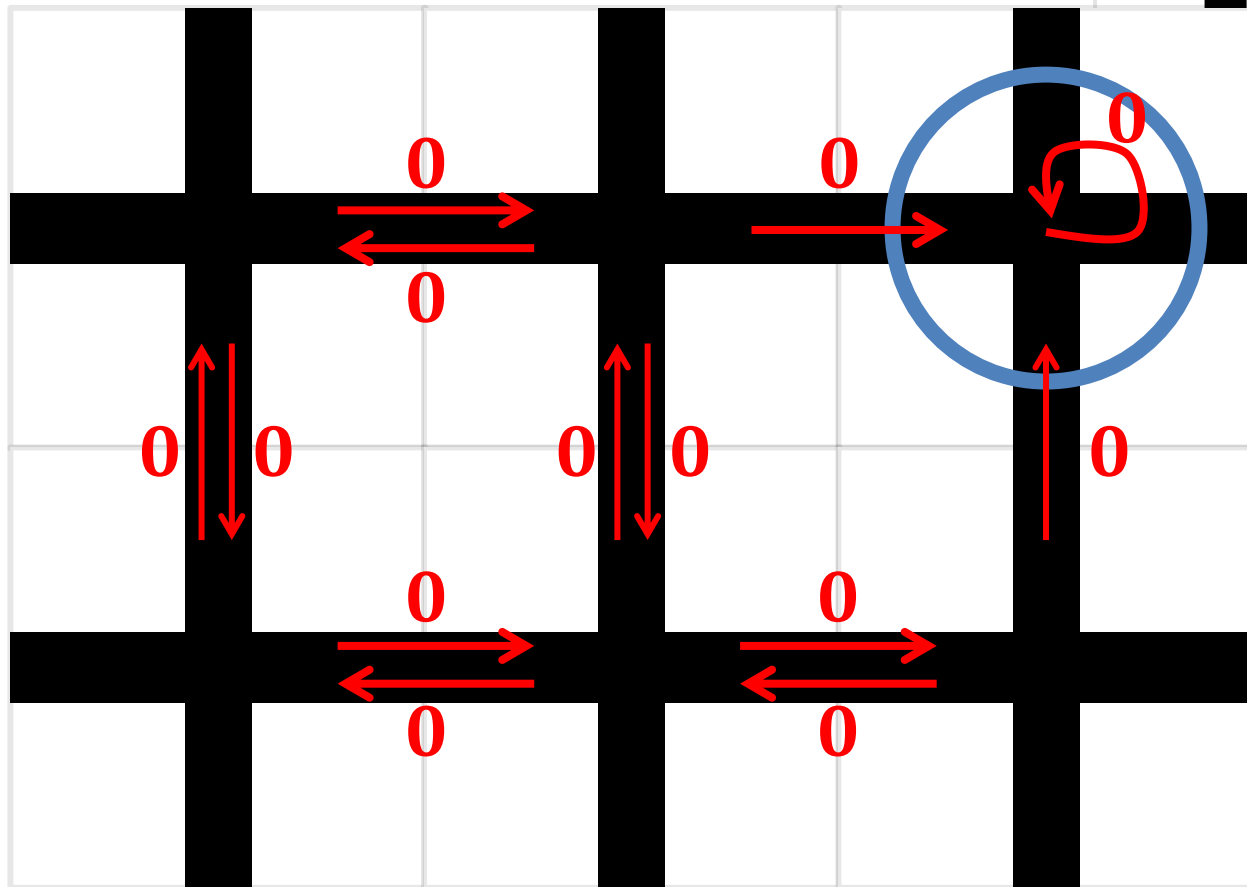
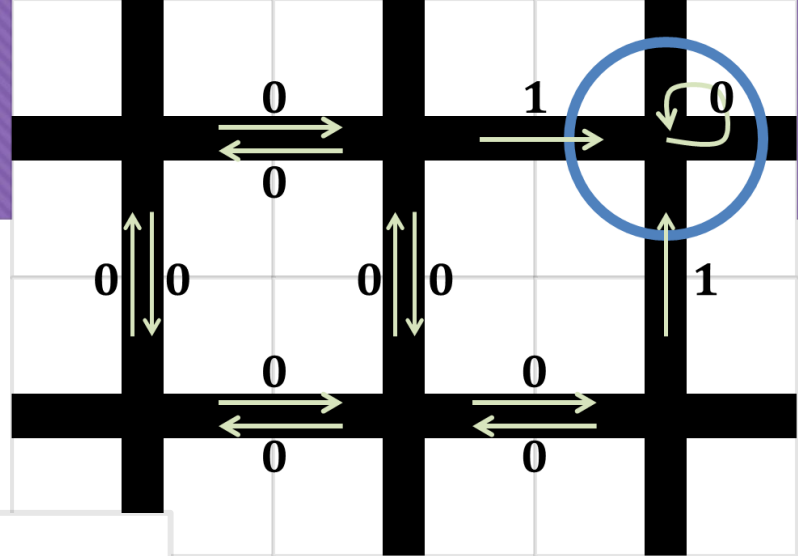
$$\pi^*(s) = \operatorname{argmax}_a Q(s, a)$$

각 상태에서 $Q(s, a)$ 가 최대인 행동(a)을
취하면 됨

Q-러닝

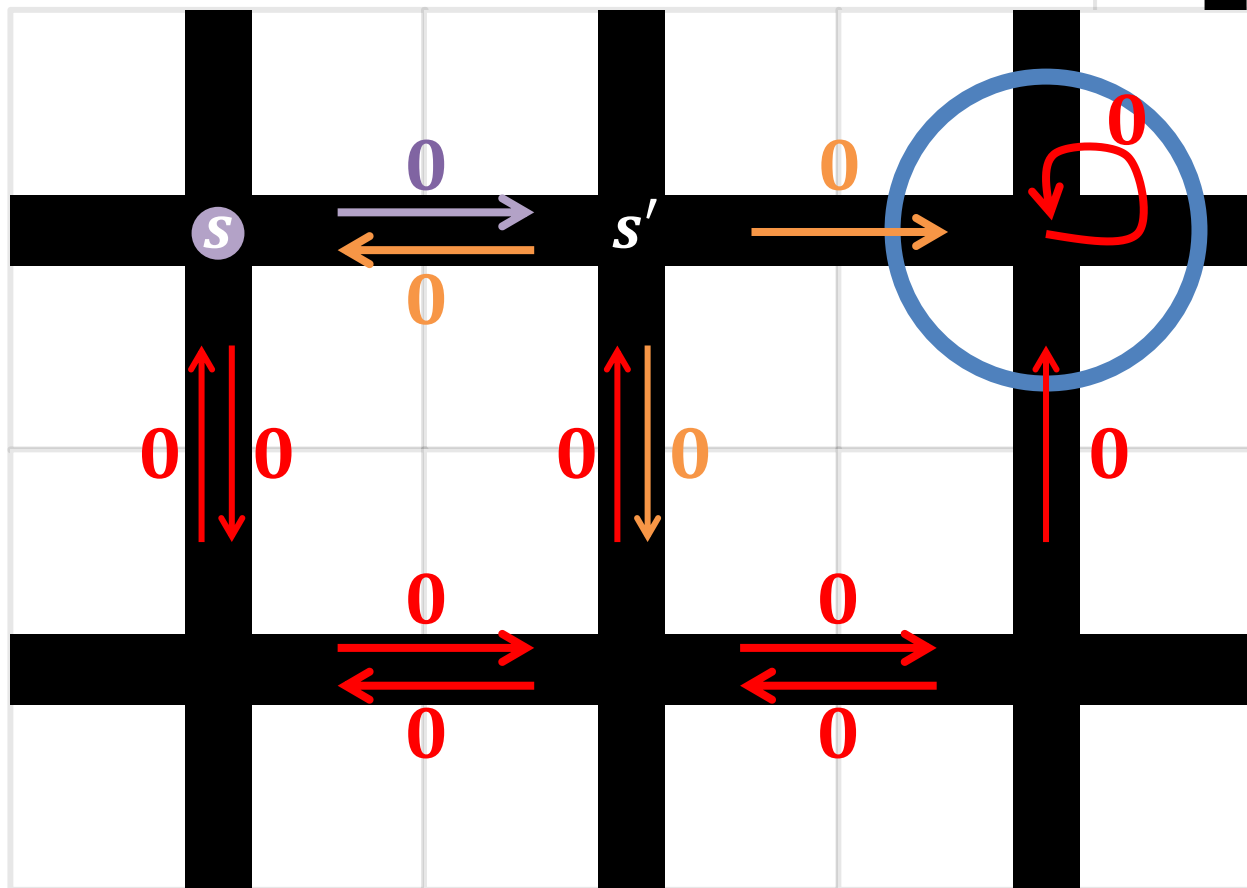
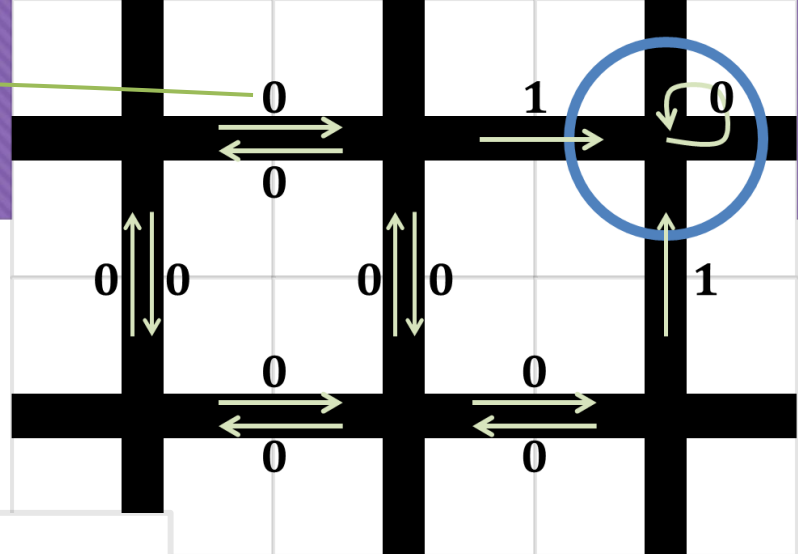
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

\hat{Q} 초기 값 : 모두 0



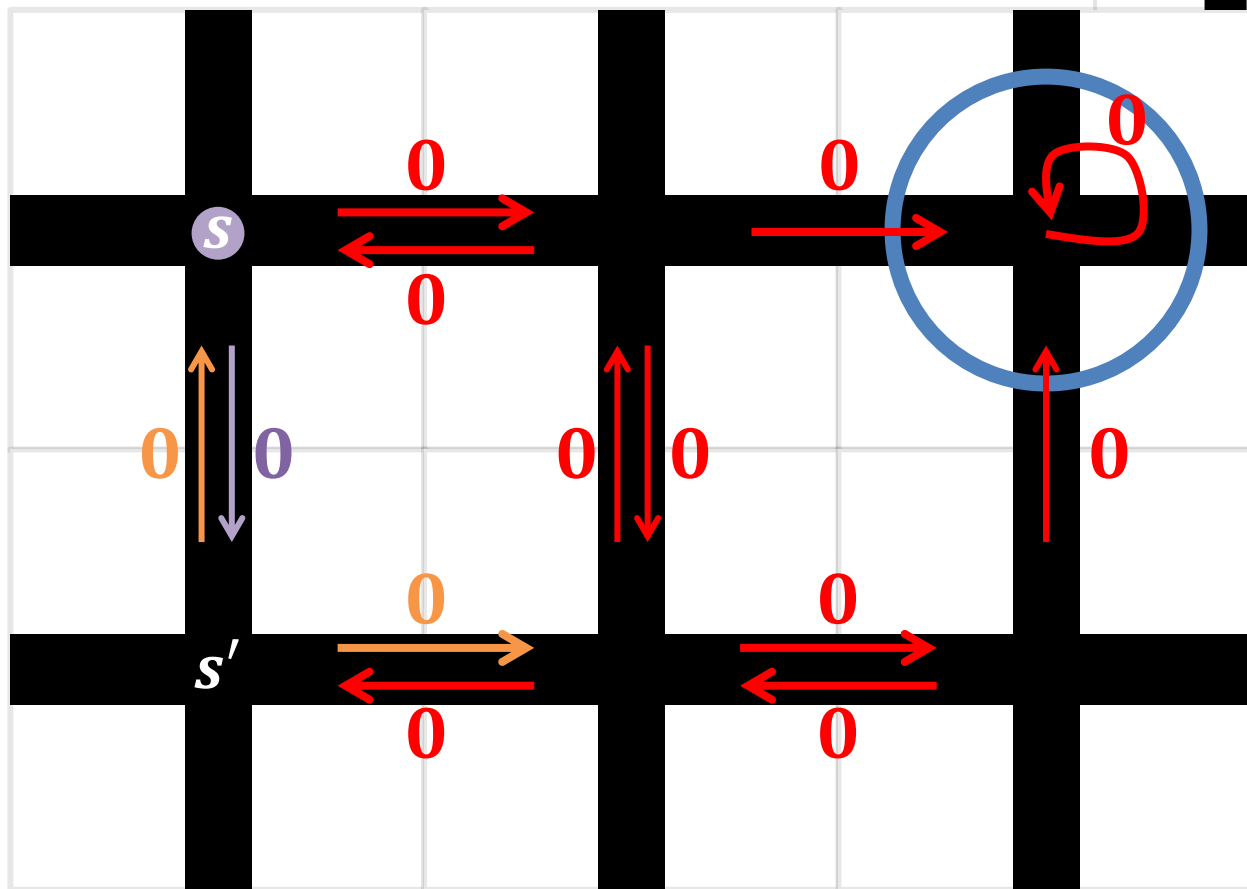
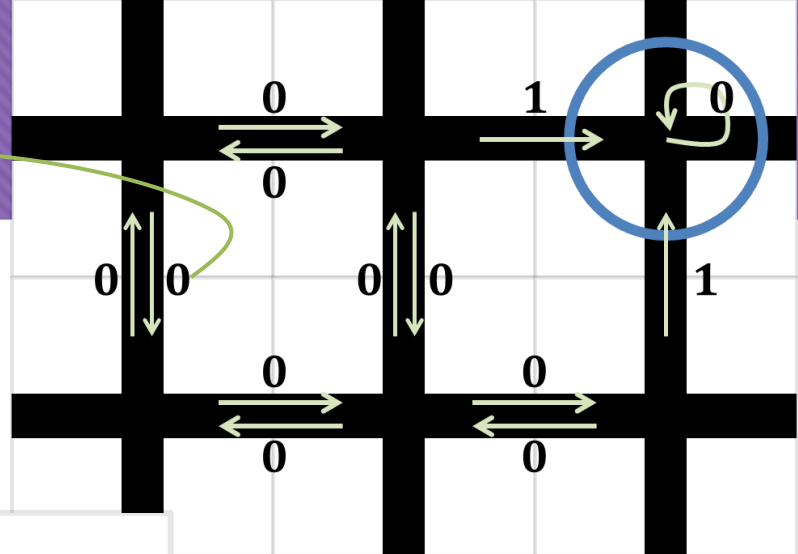
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

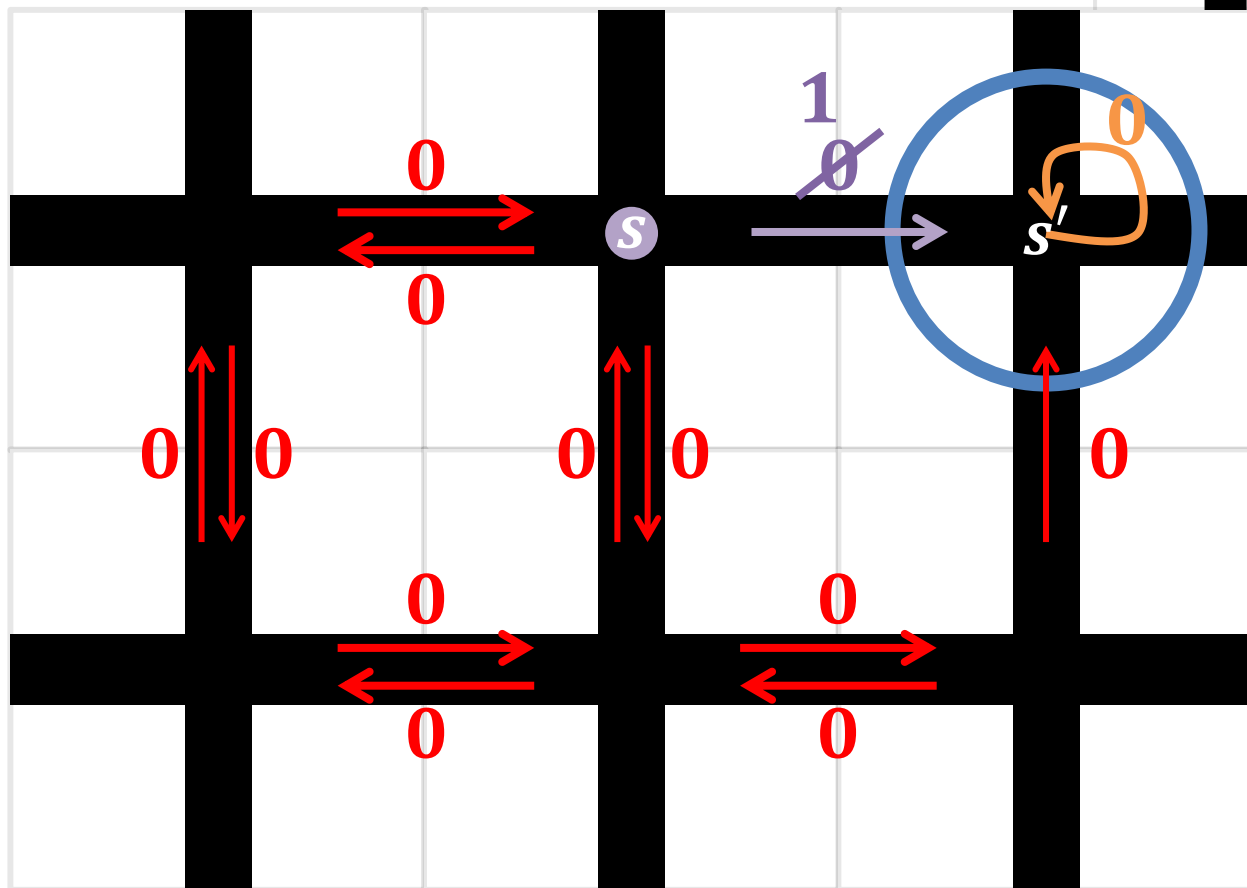
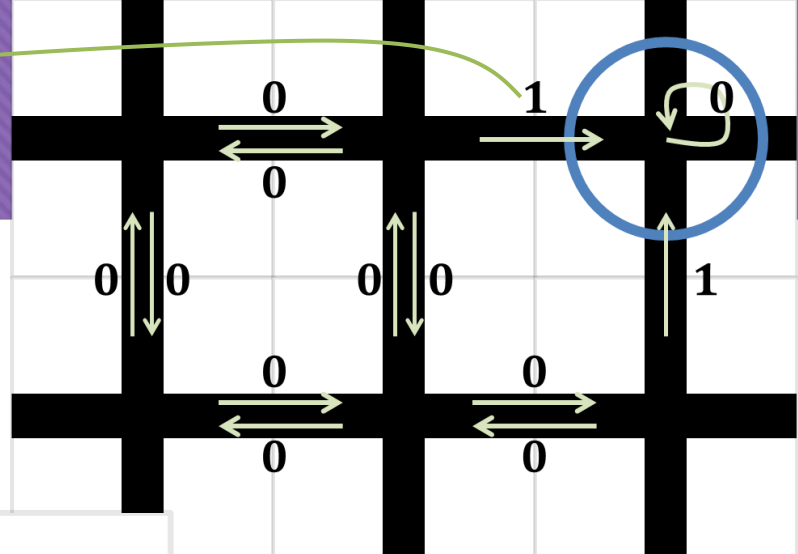
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

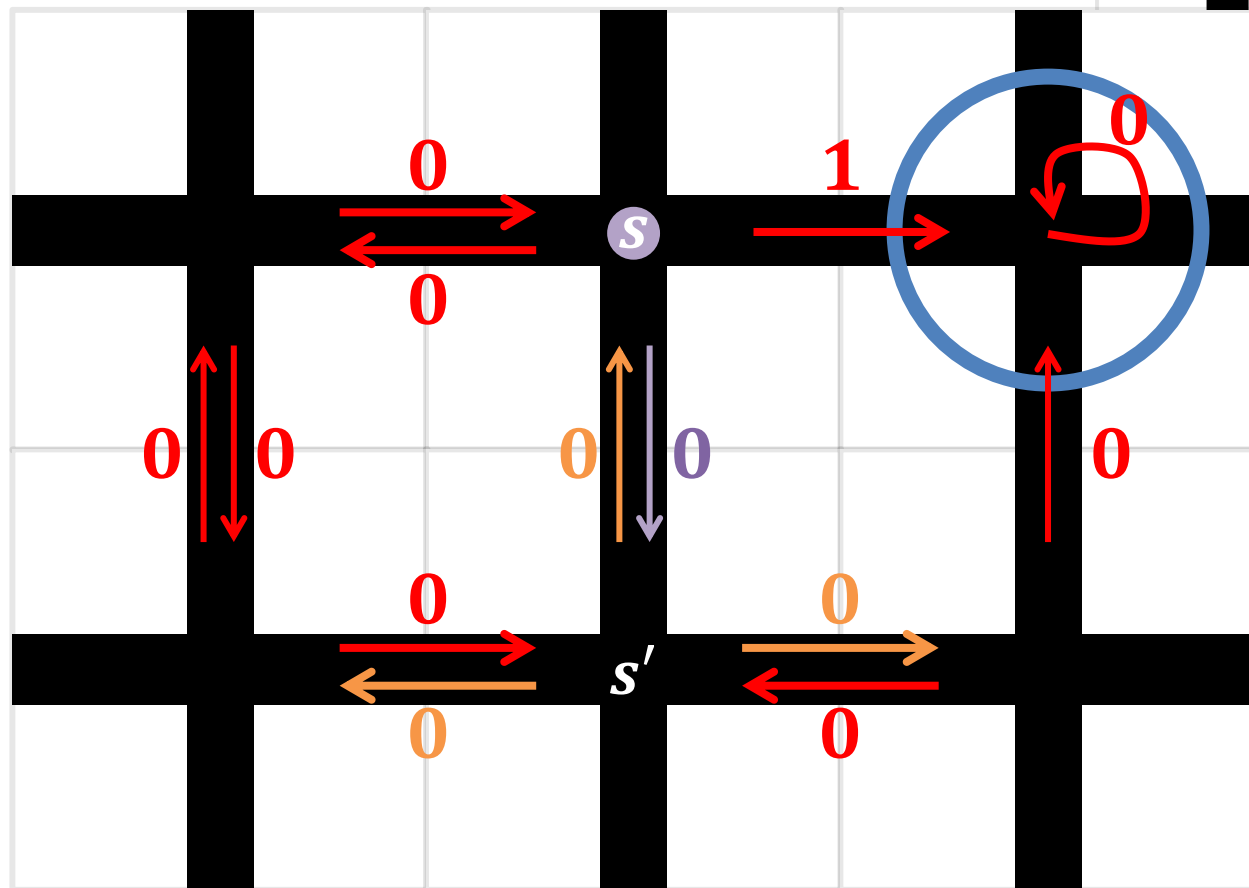
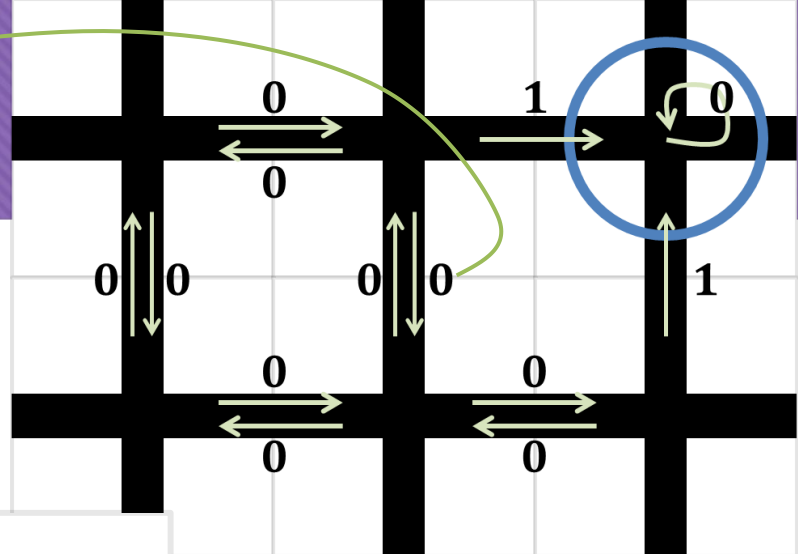
1 0



Q-러닝

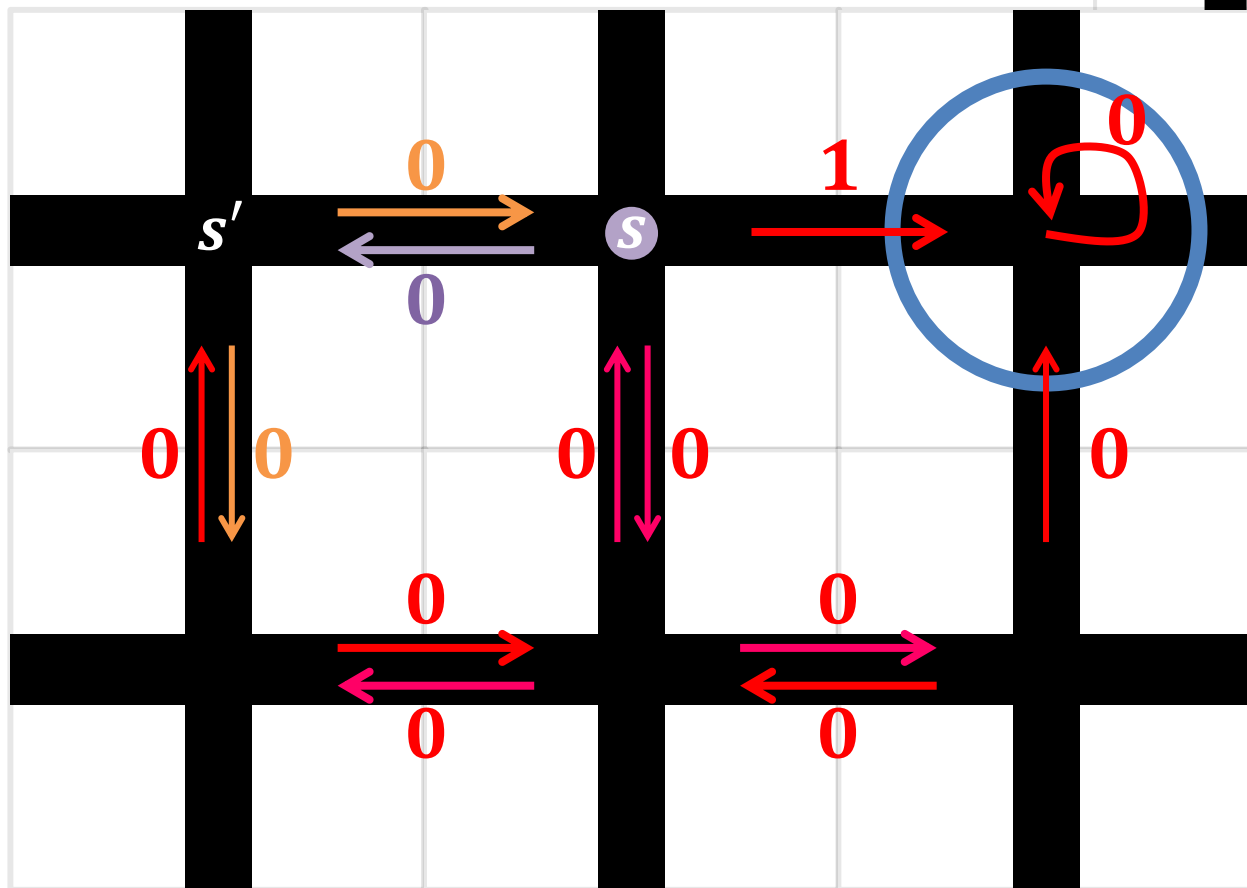
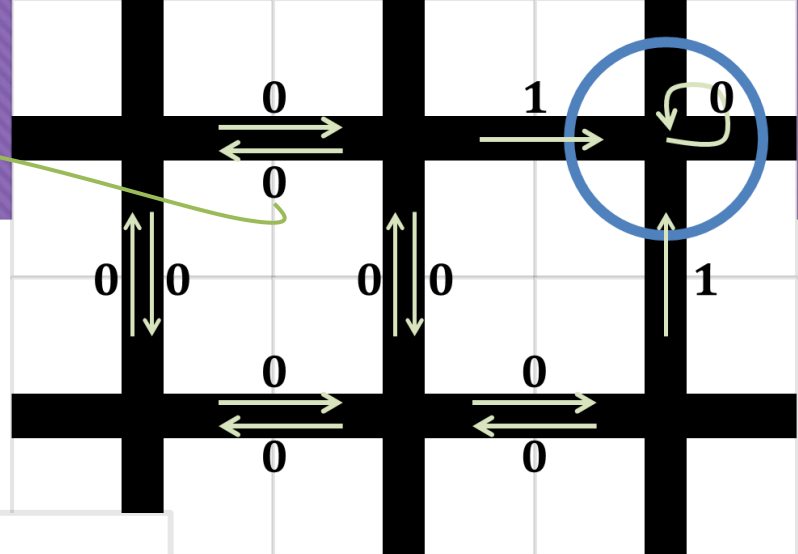
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

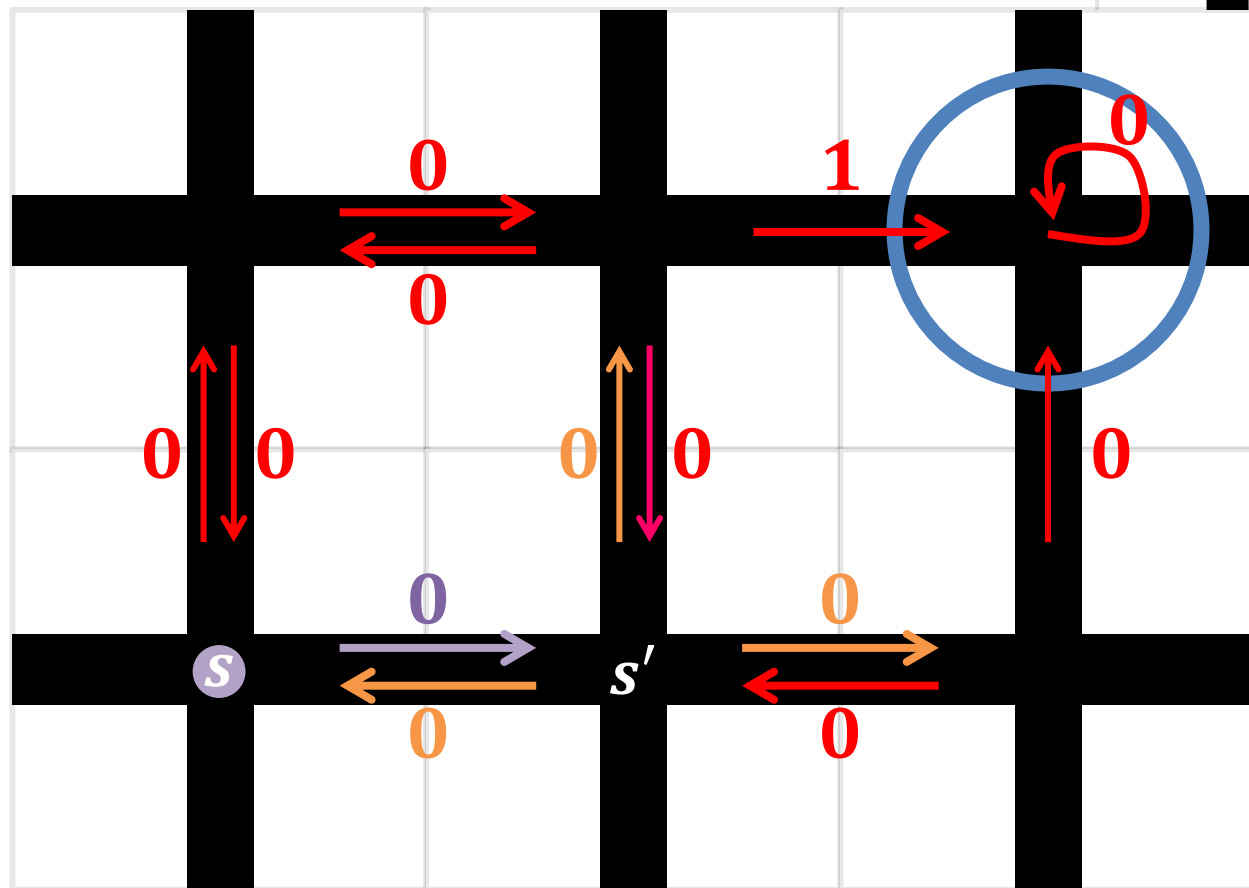
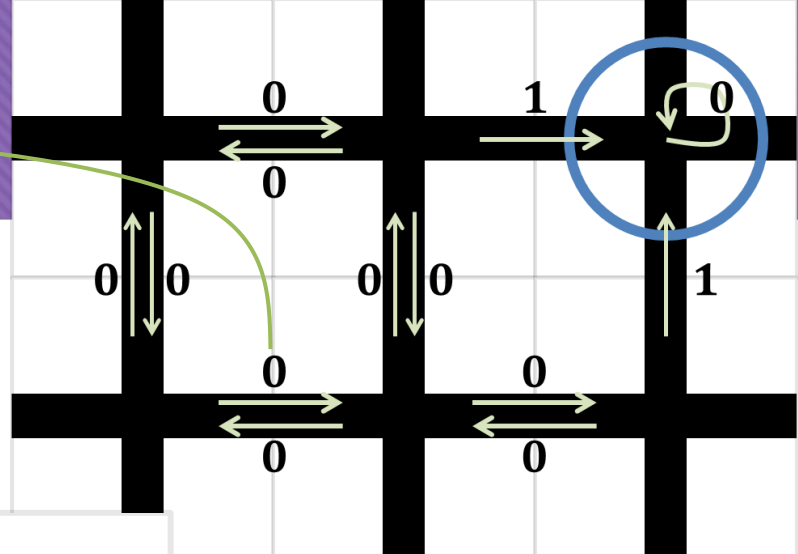
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

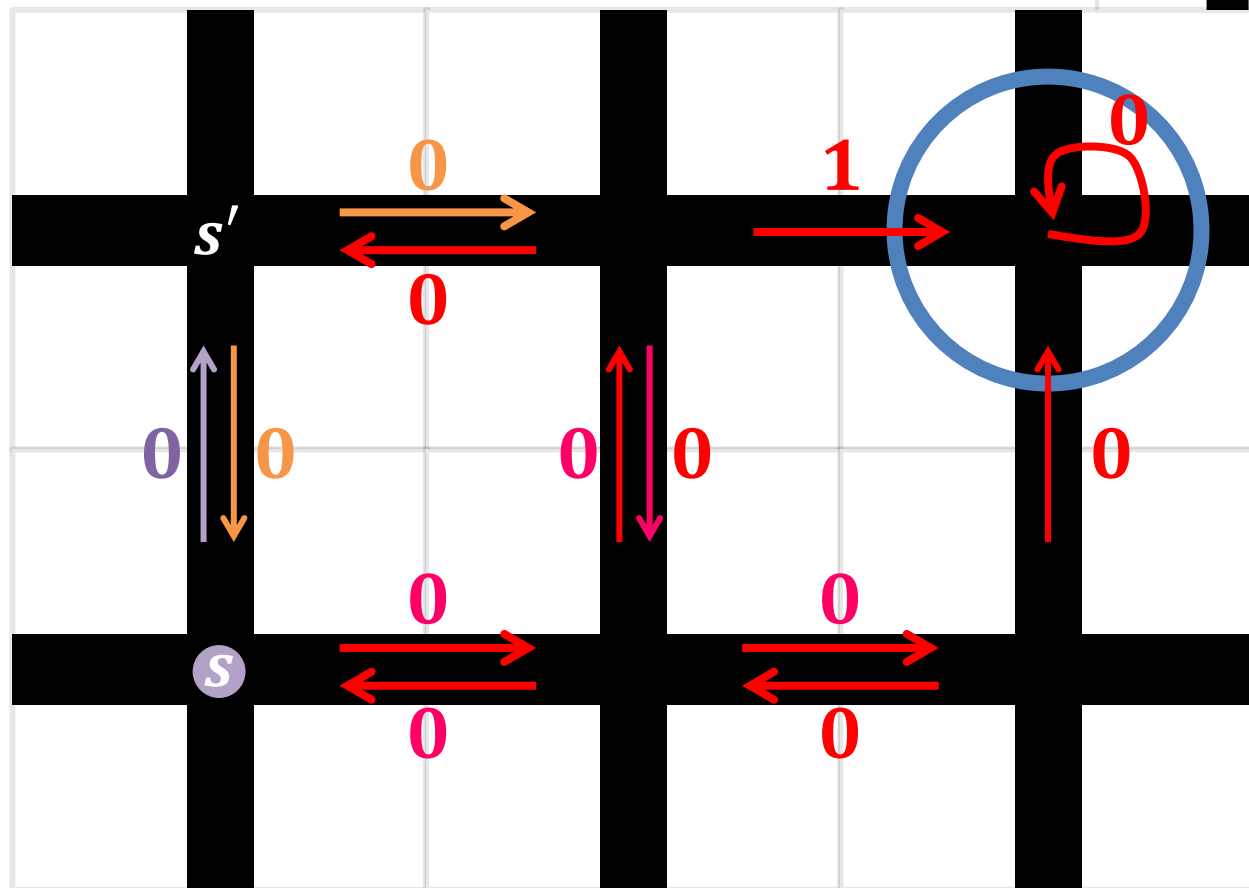
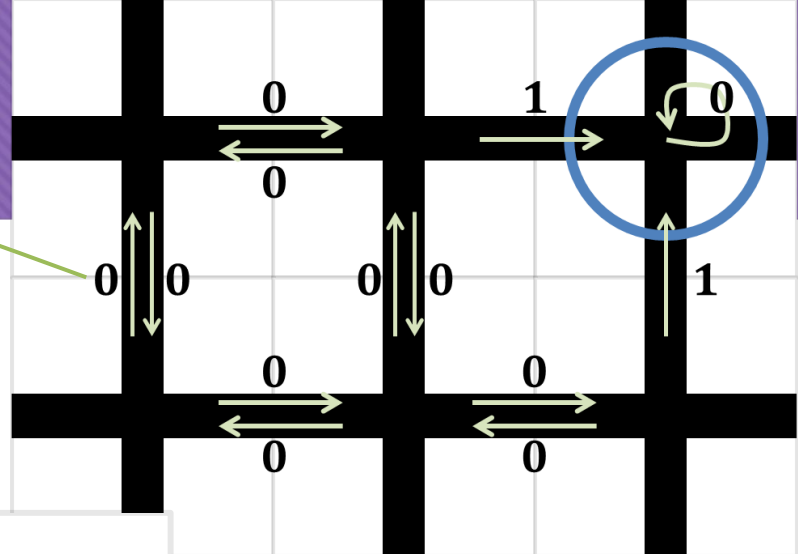
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

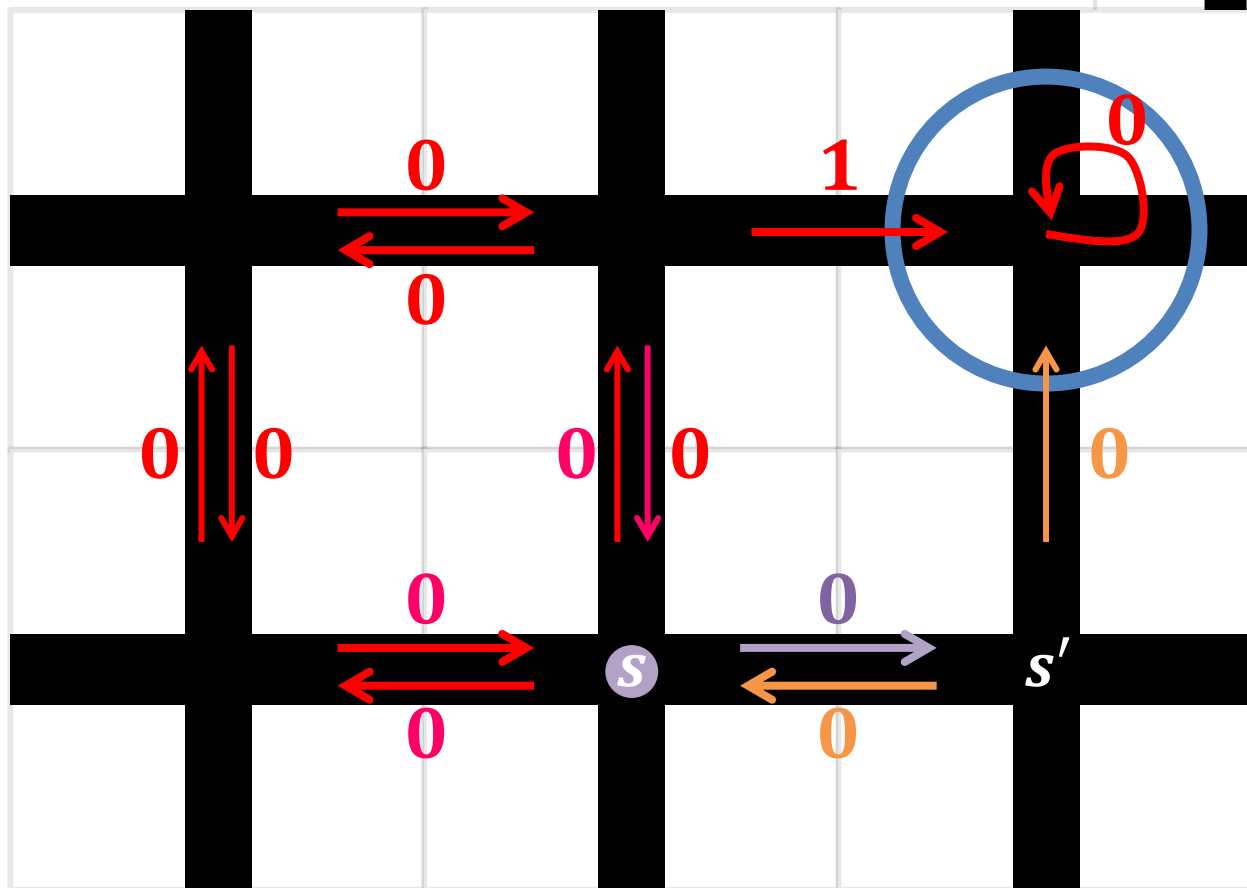
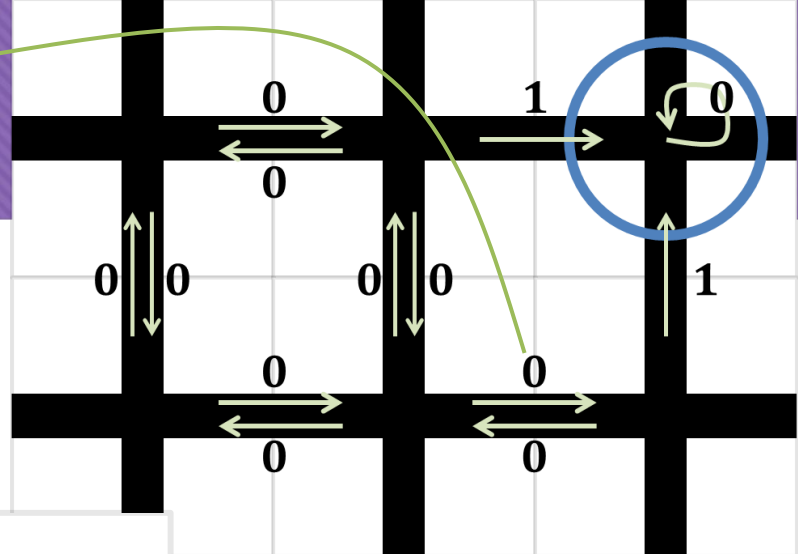
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

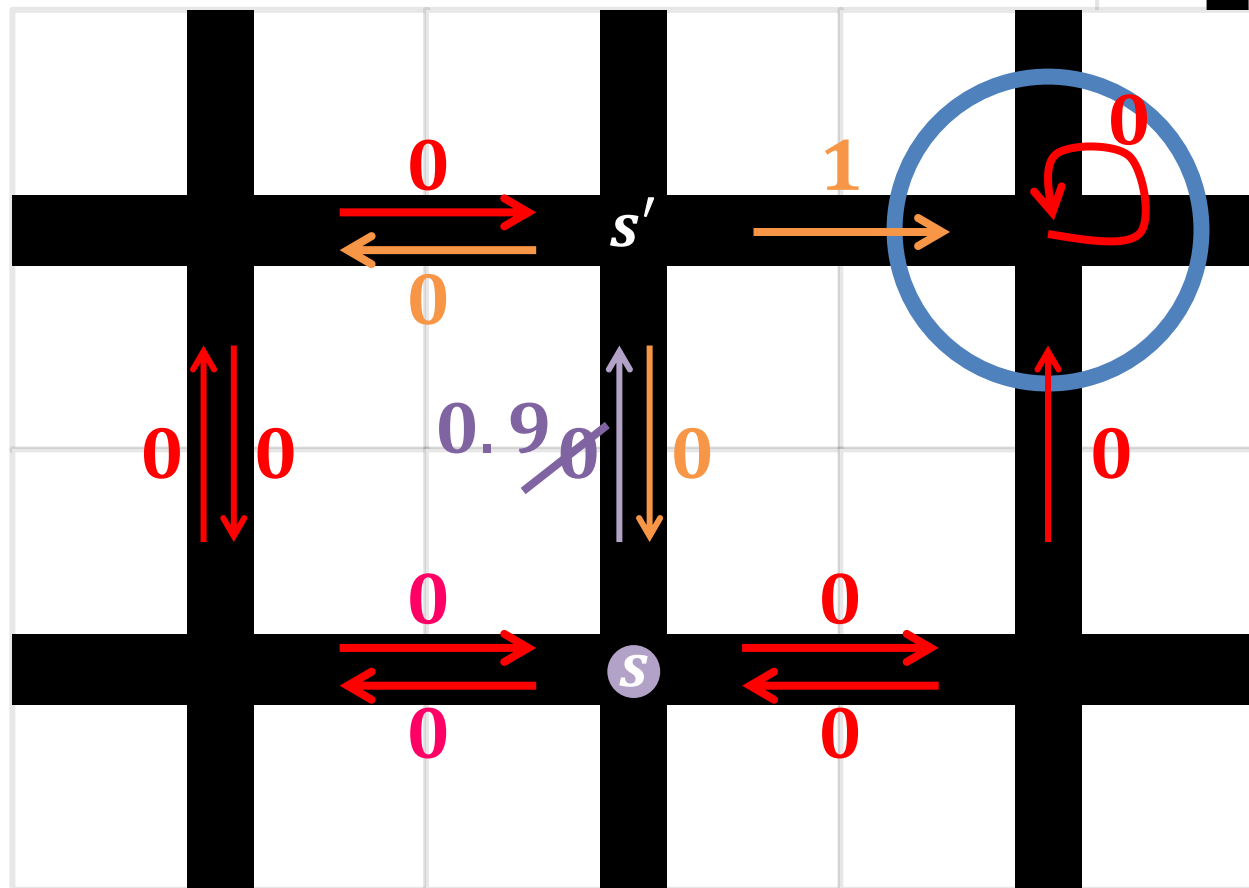
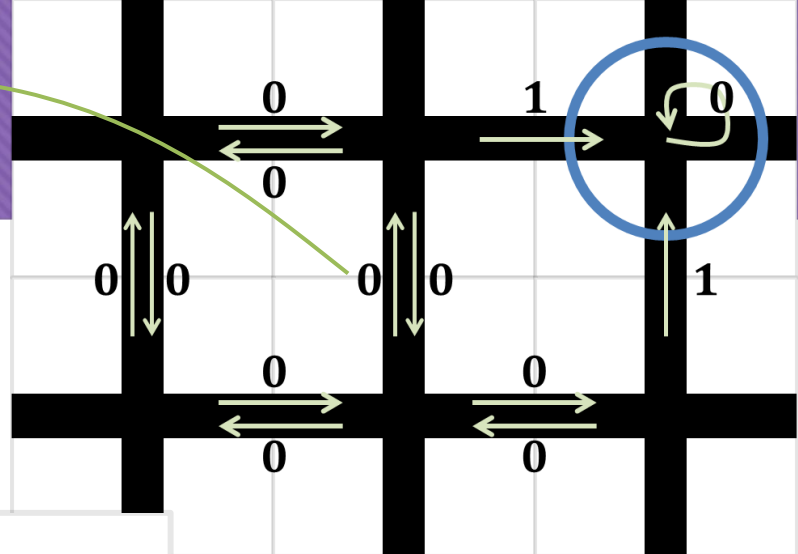
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



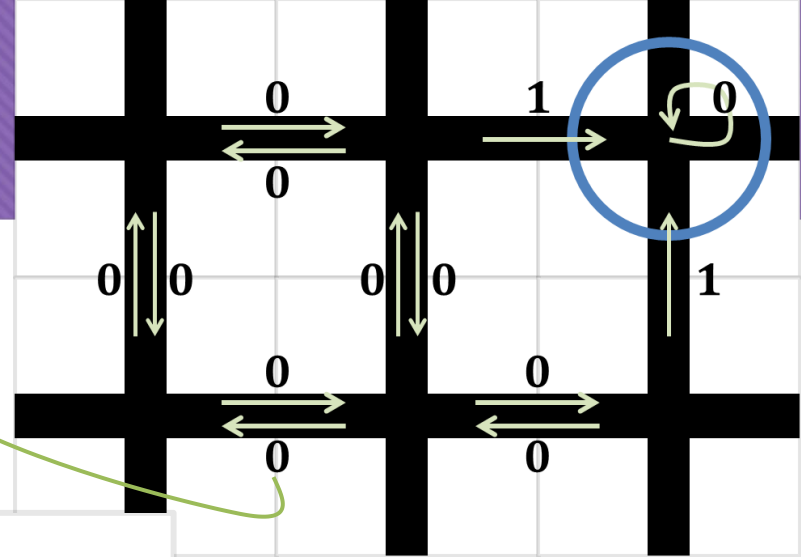
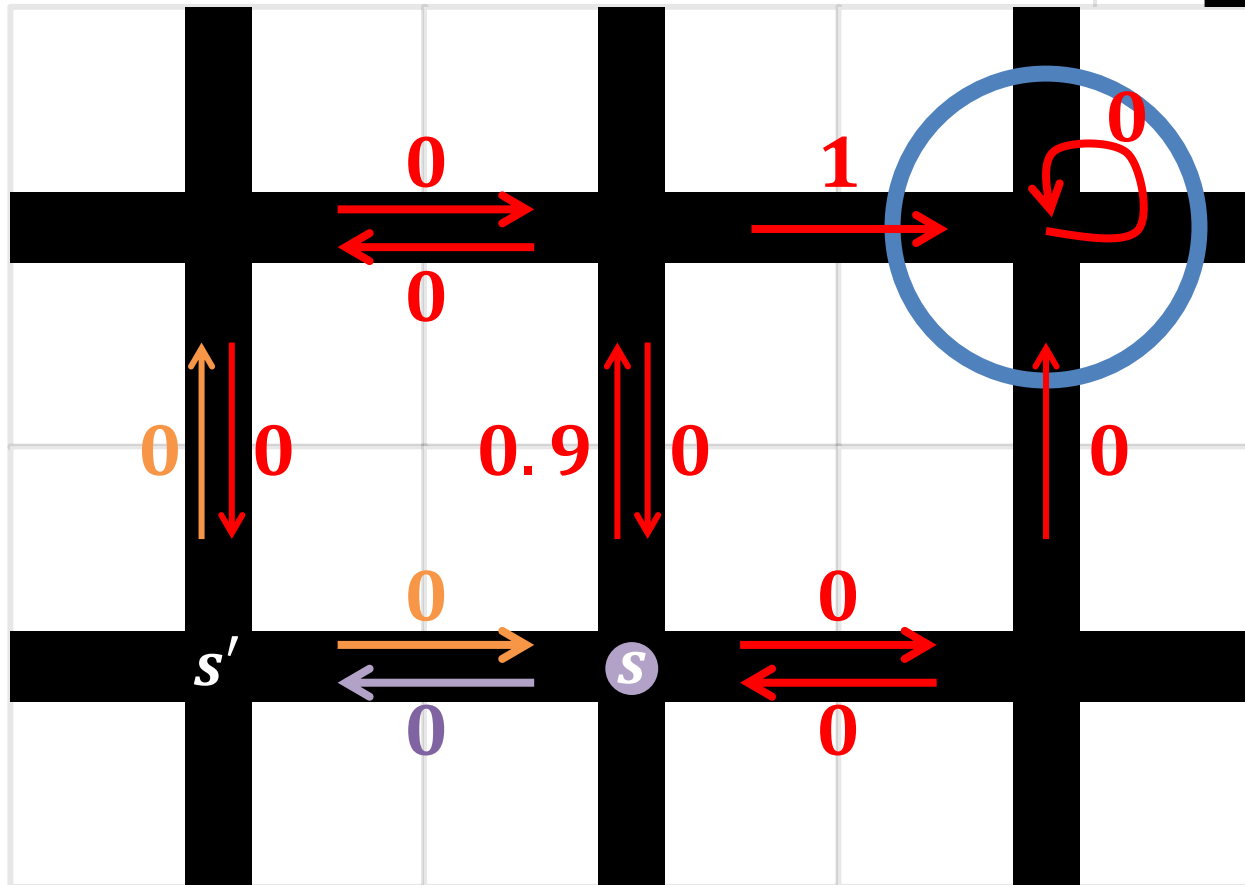
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

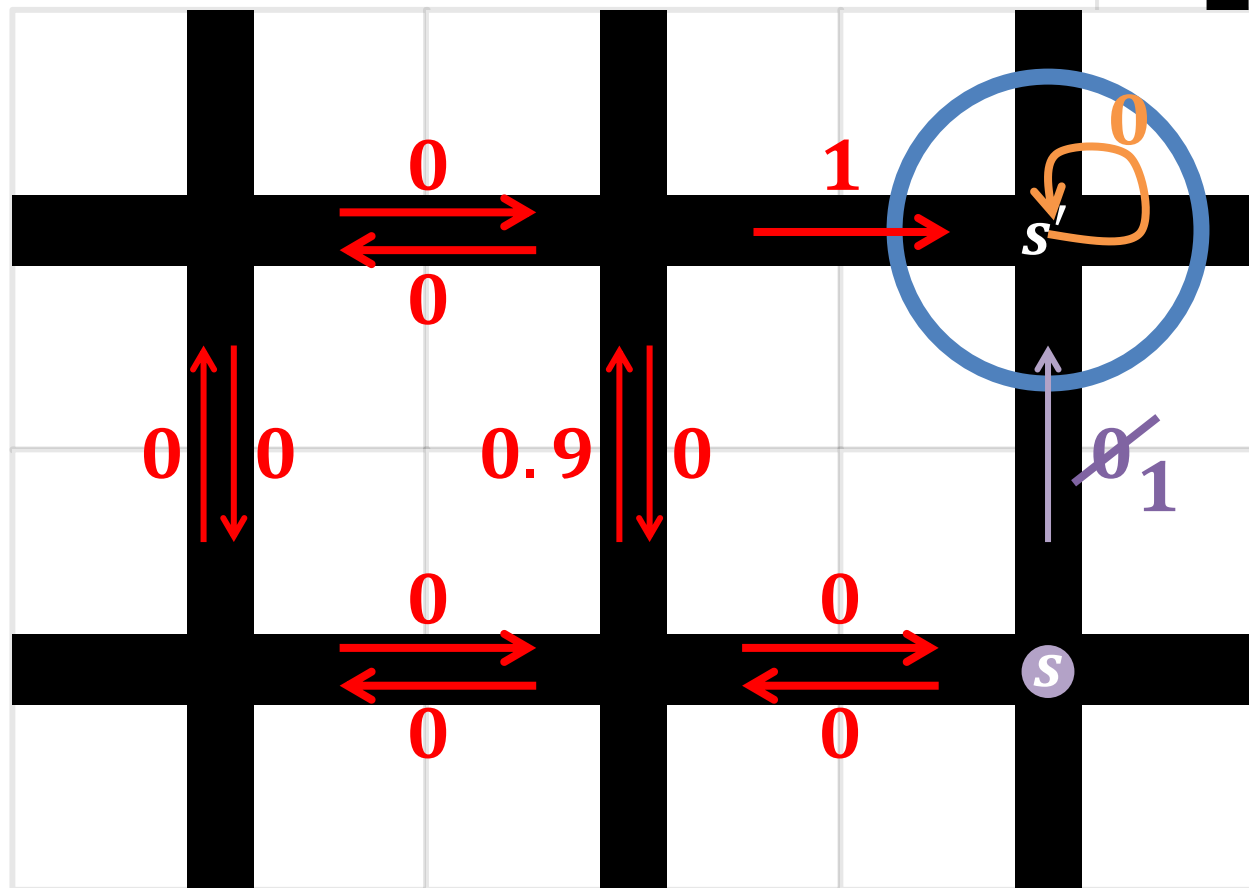
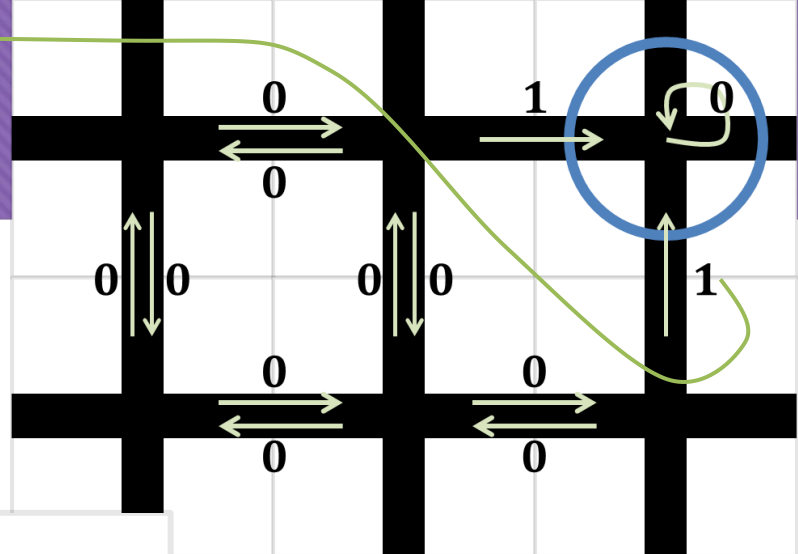
$$\hat{Q}(s, a) \leftarrow \underset{0}{\underset{0}{\mathbf{r}}} + \mathbf{0.9} \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

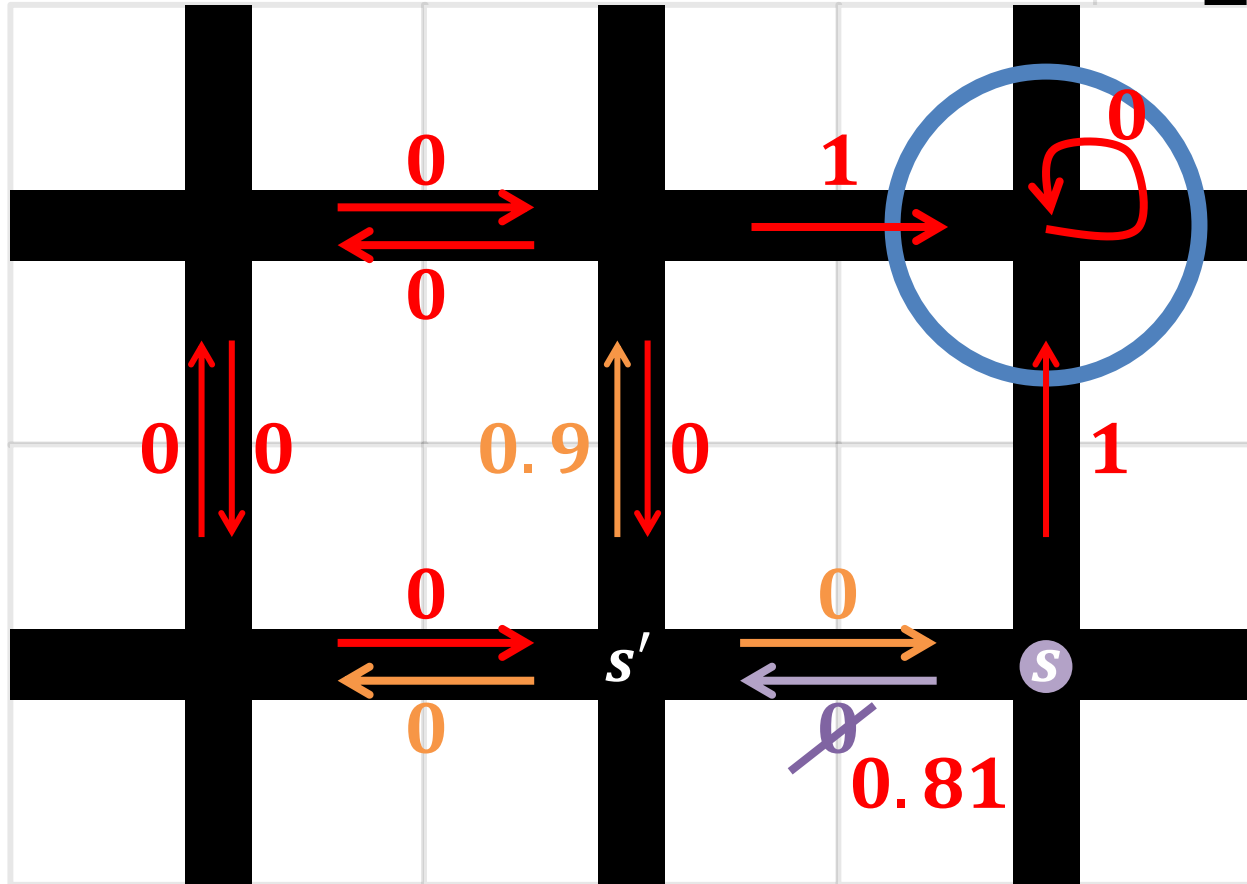
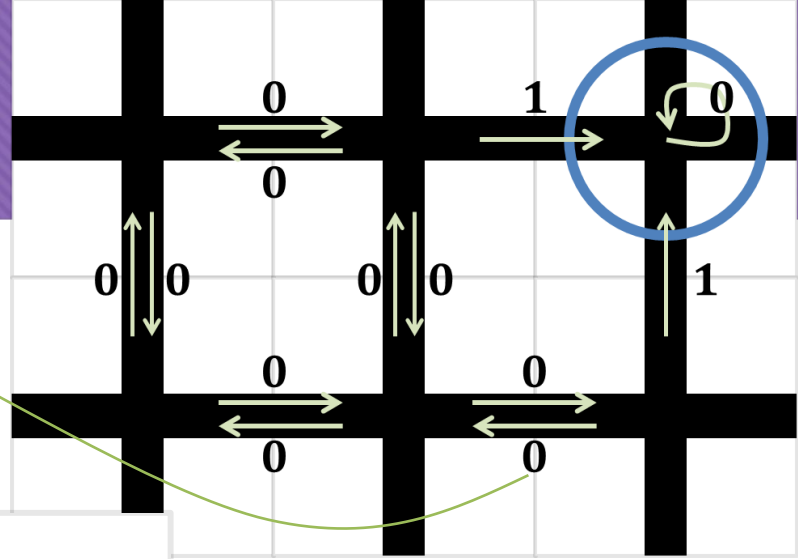
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

1 0



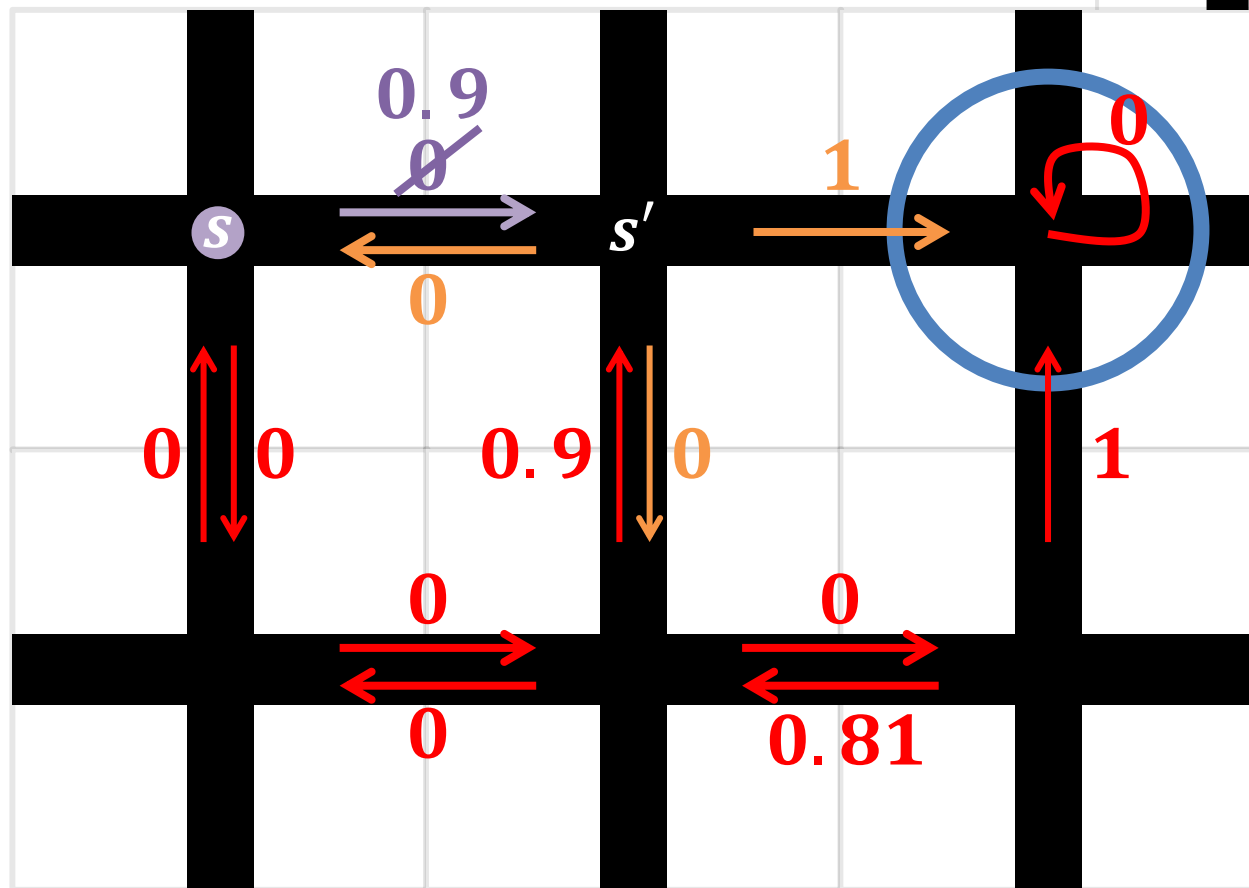
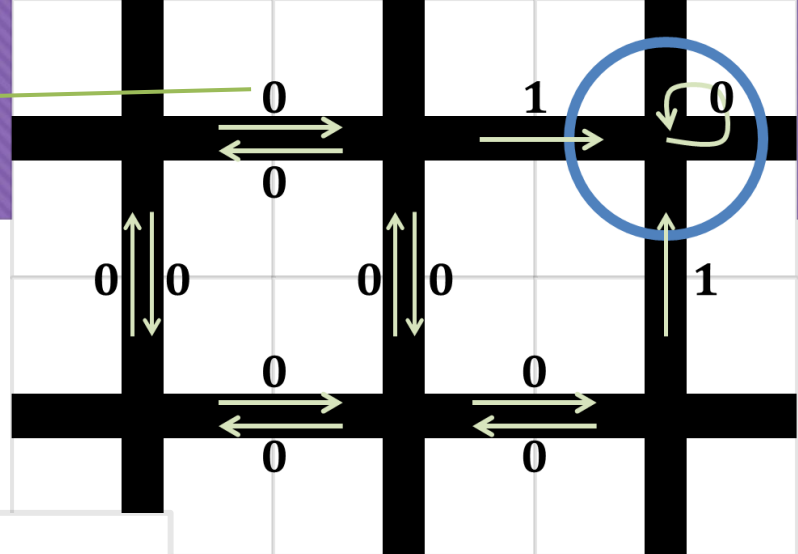
Q-러닝

$$\hat{Q}(s, a) \leftarrow \underset{0}{\mathbf{r}} + \underset{0.9}{\mathbf{0.9}} \times \max_{a'} \hat{Q}(s', a')$$



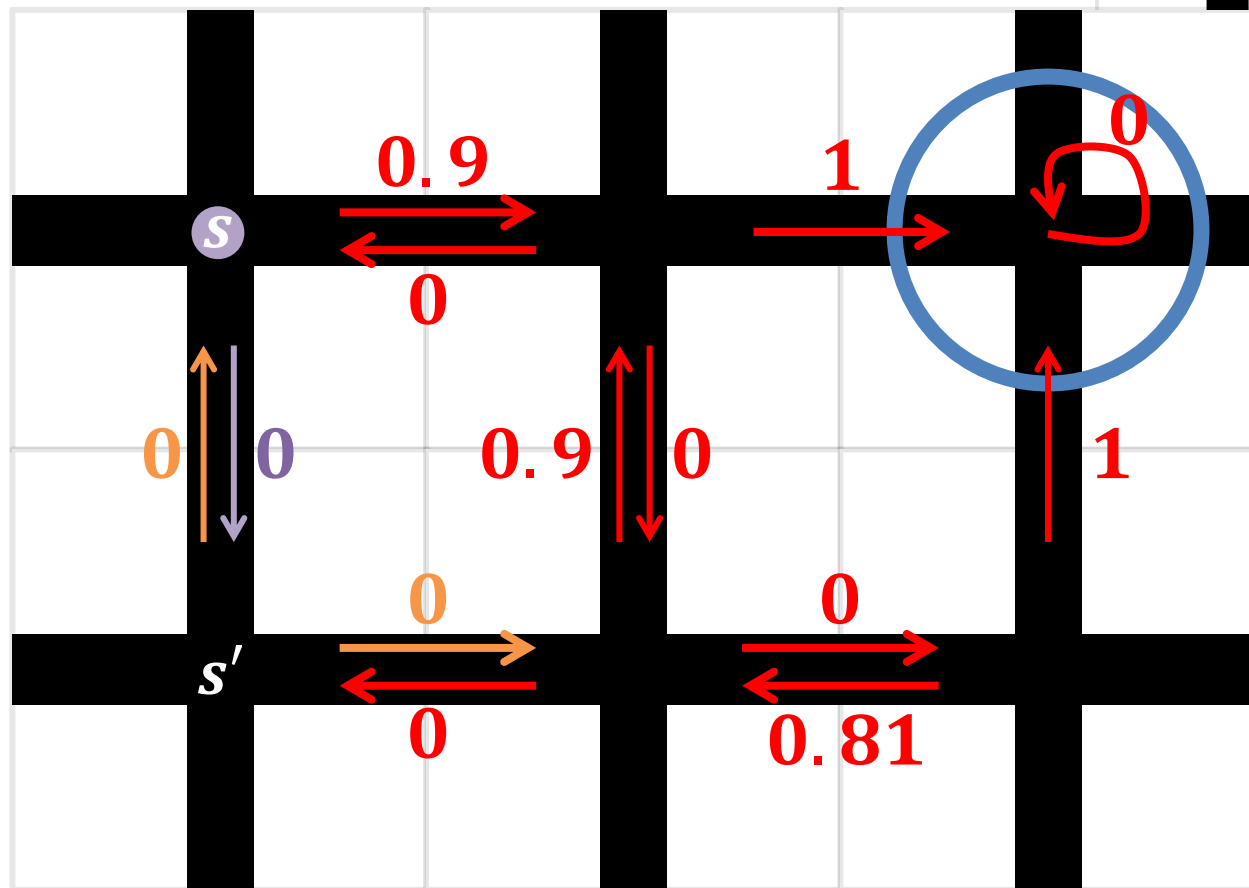
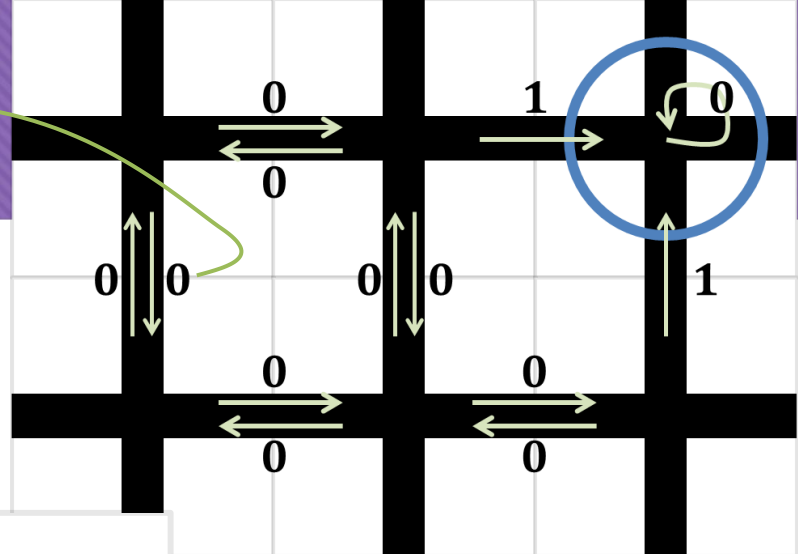
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



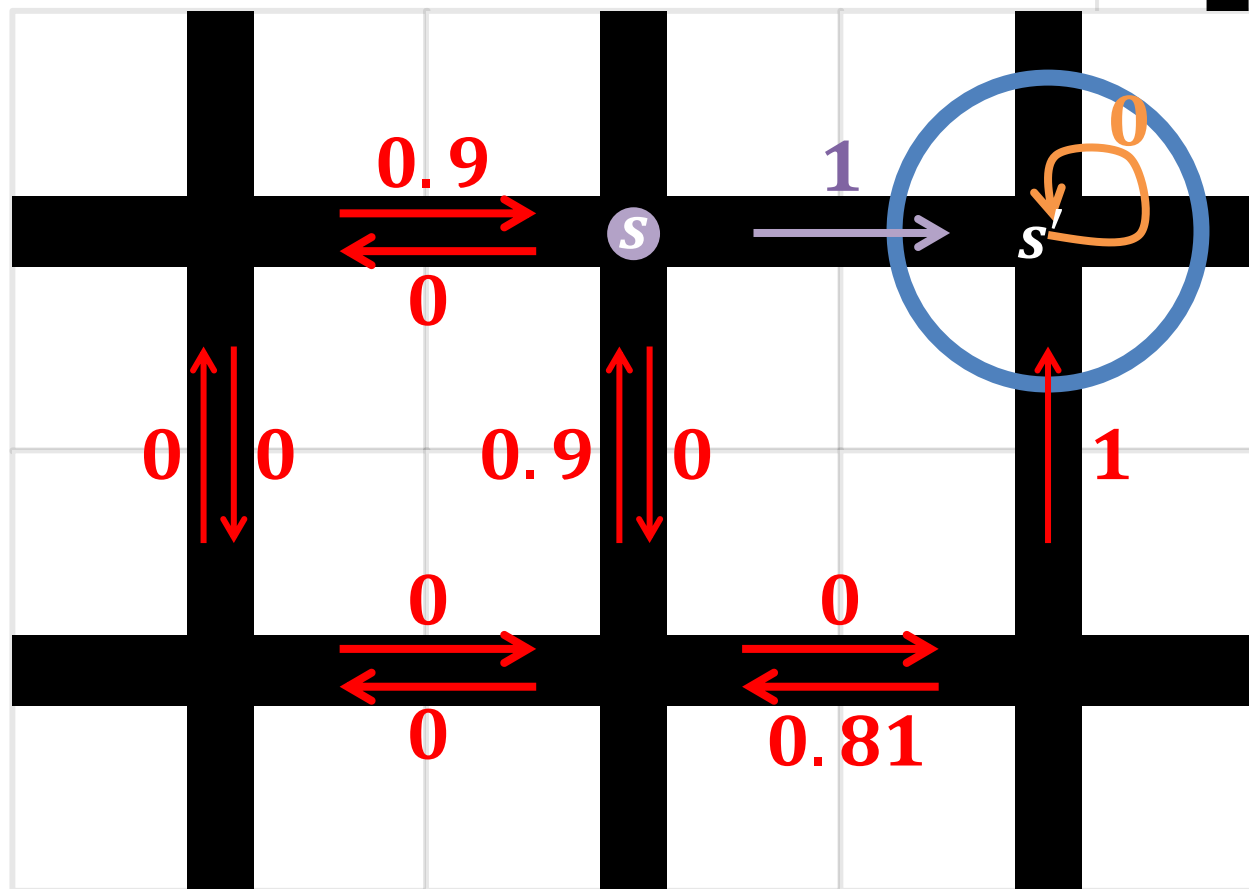
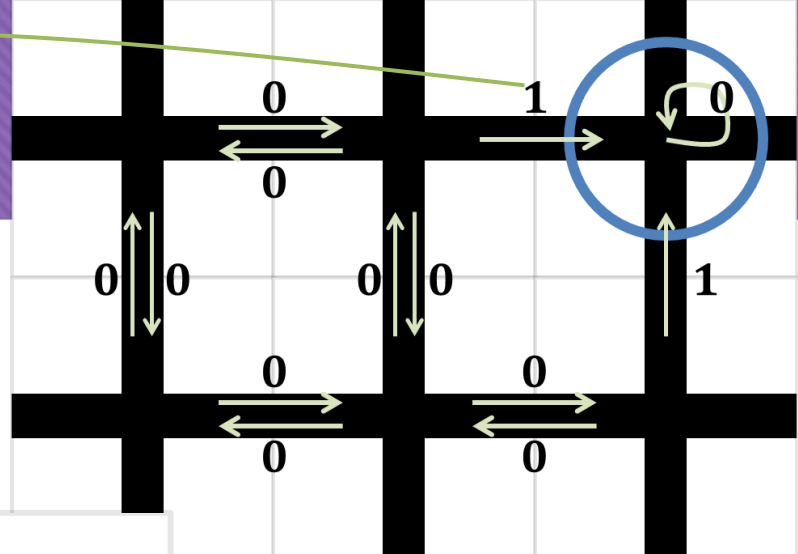
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



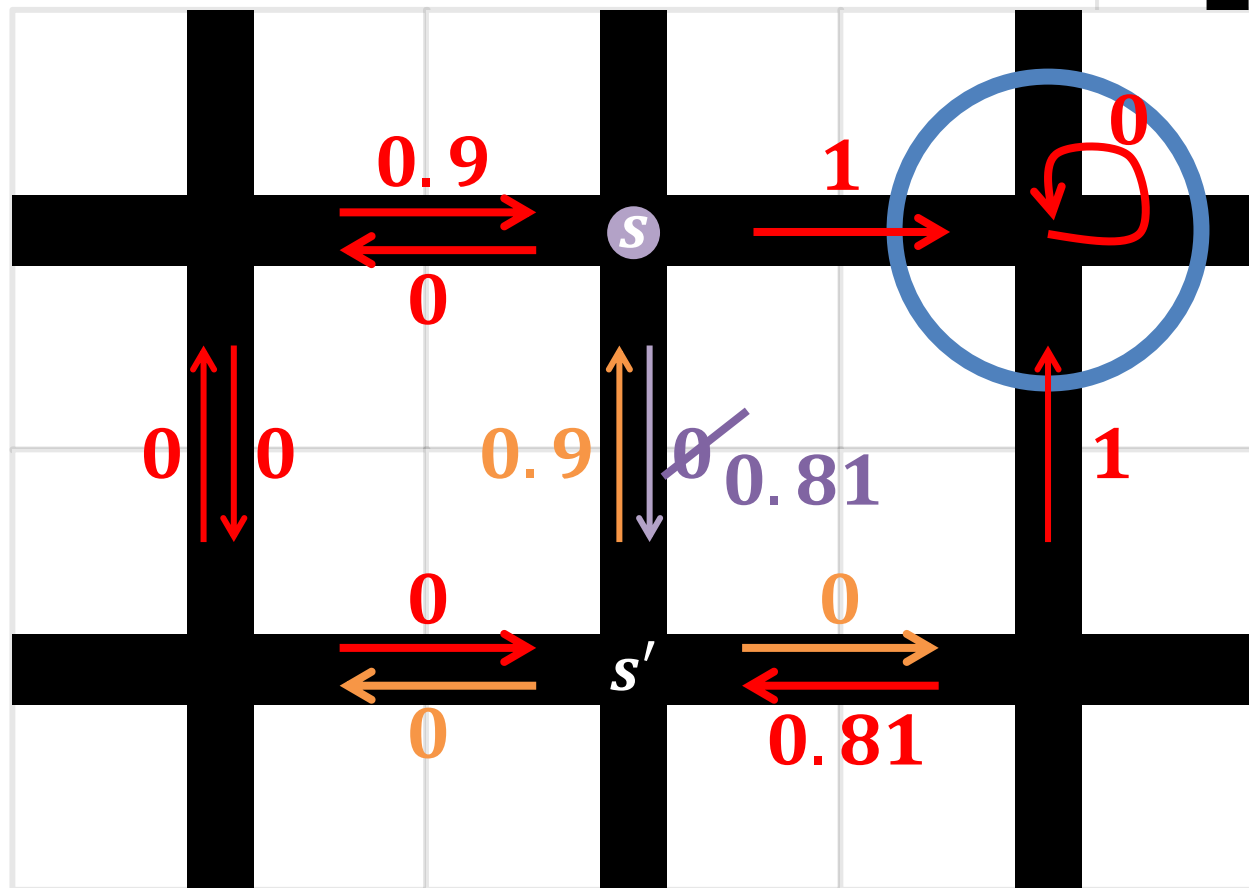
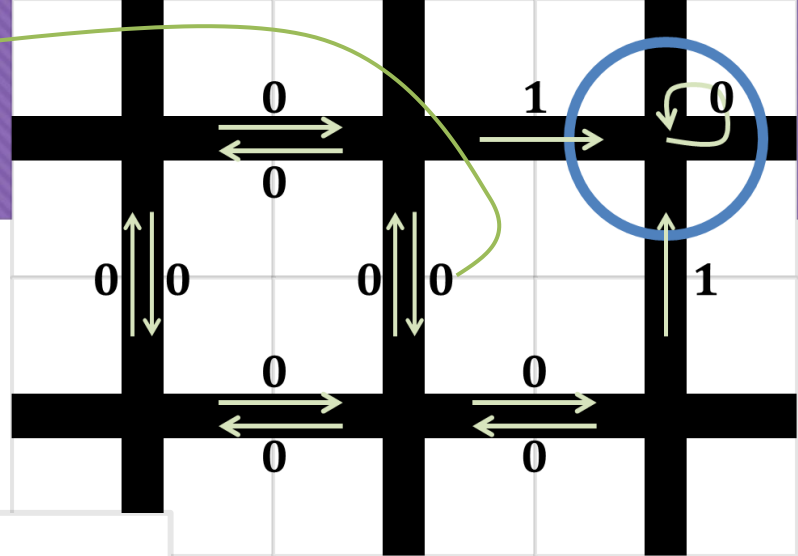
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

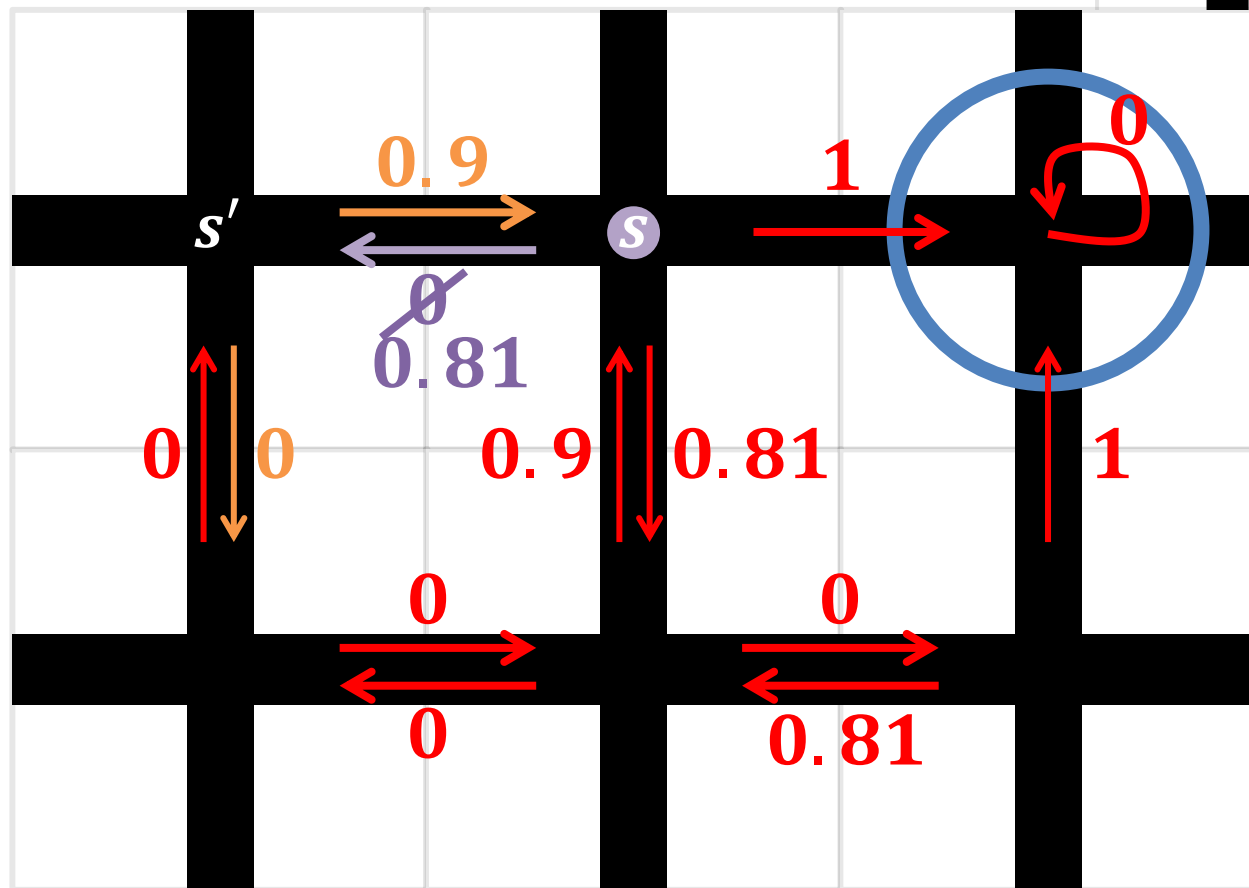
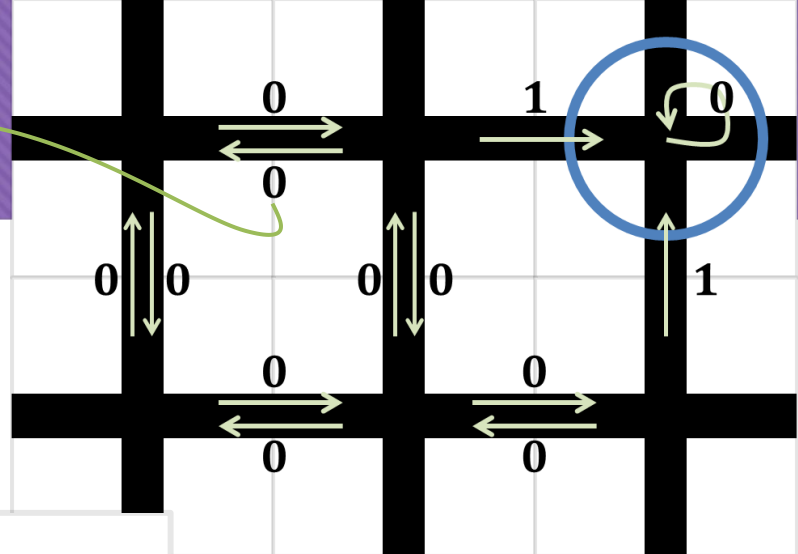
$$\hat{Q}(s, a) \leftarrow \underset{0}{\overset{\text{green}}{r}} + \underset{0.9}{\text{orange}} \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

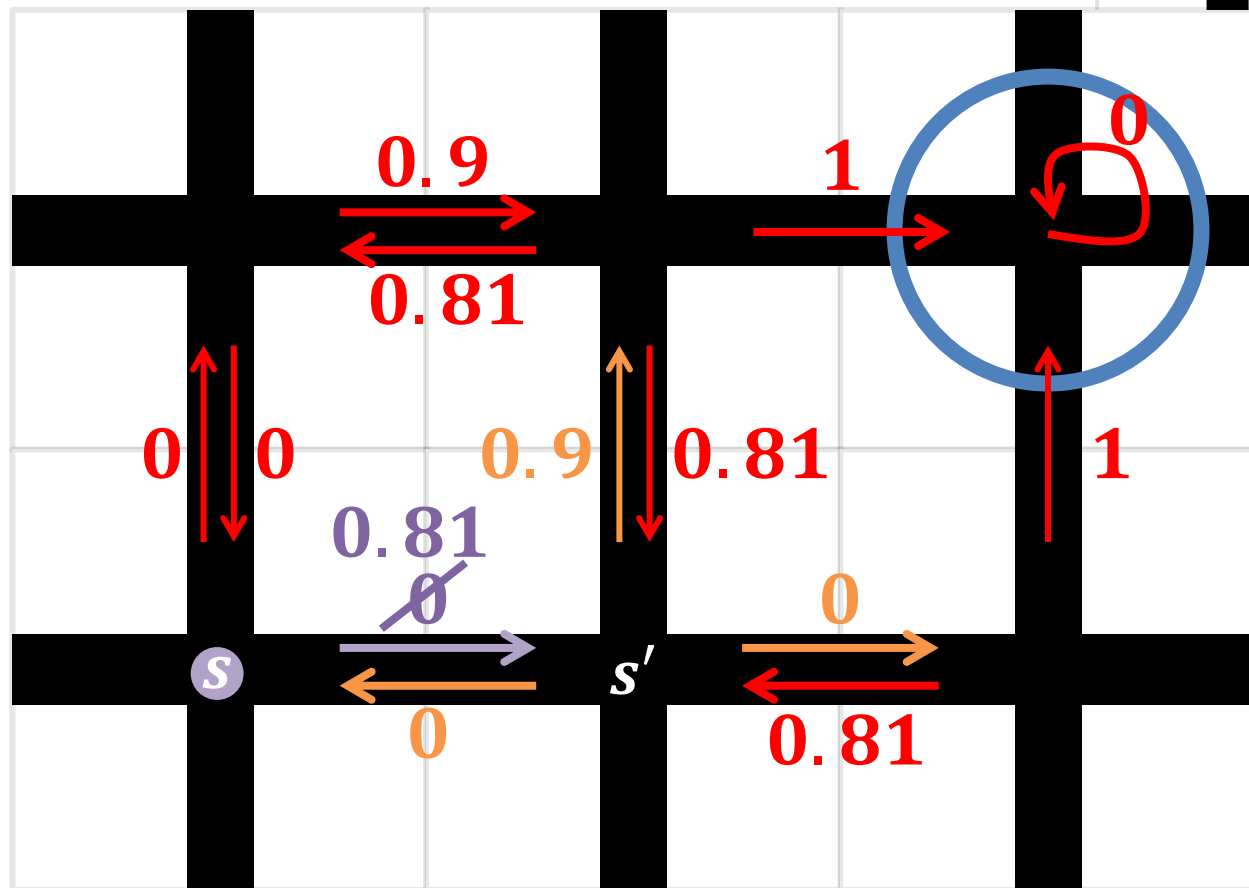
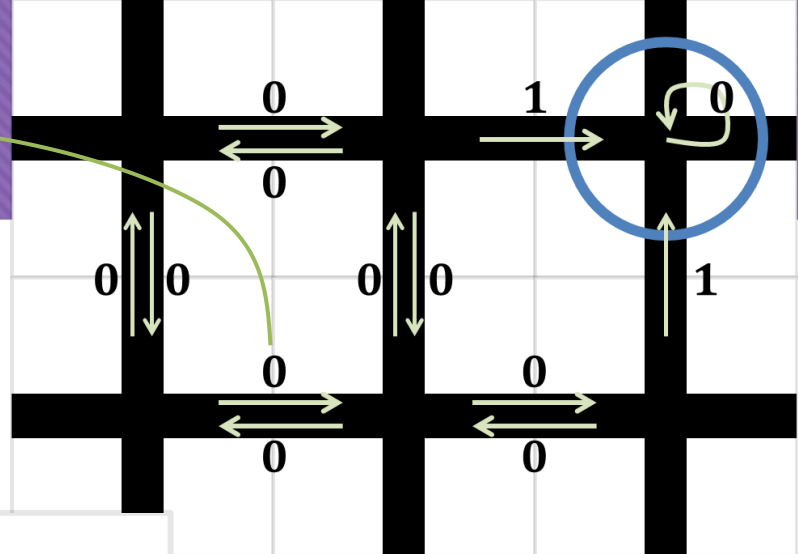
0 0.9



Q-러닝

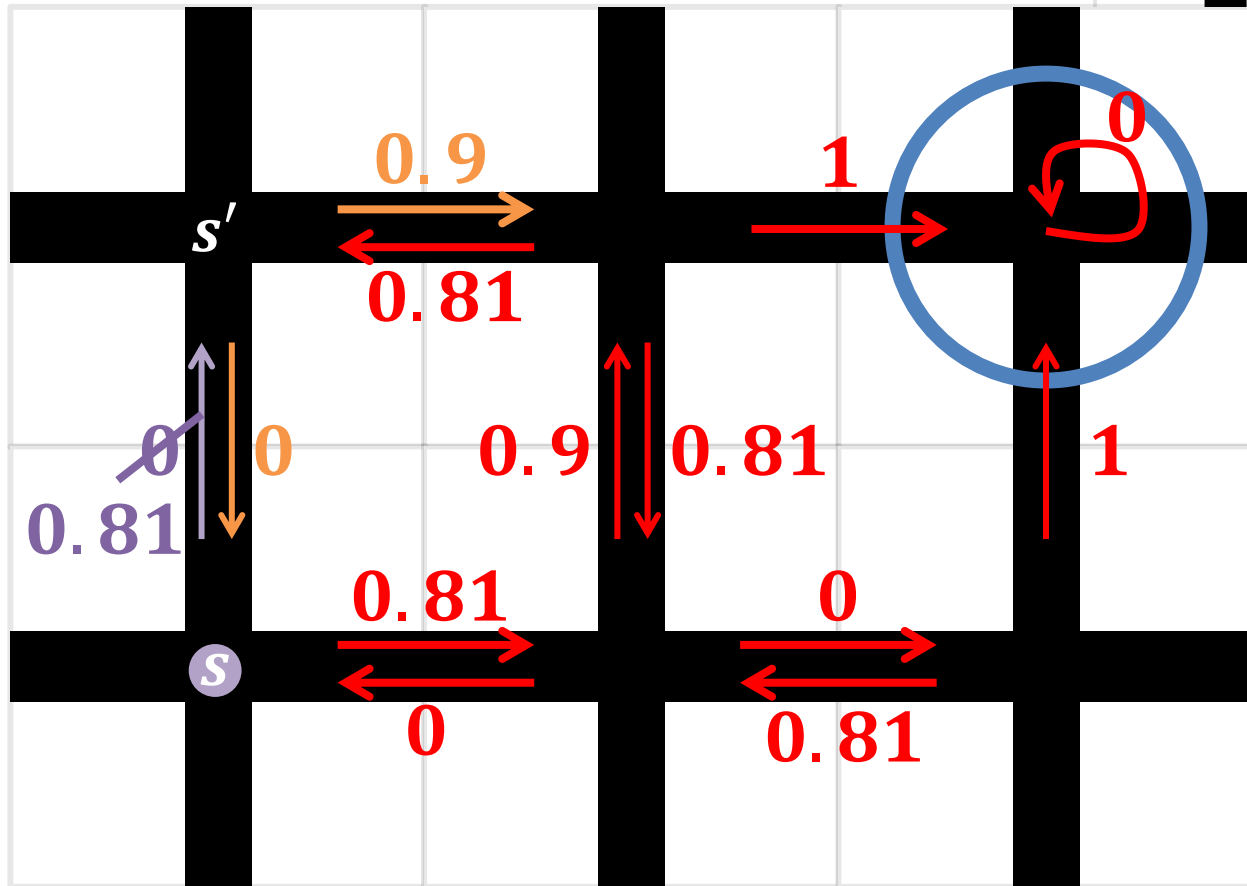
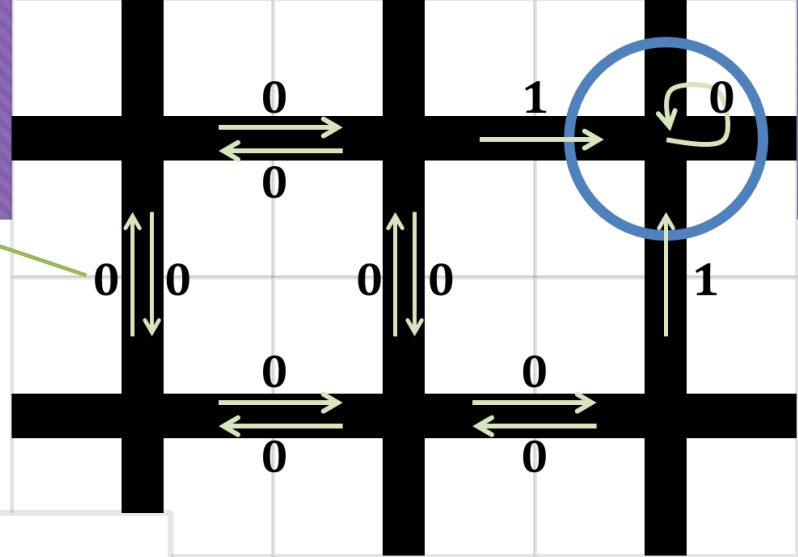
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



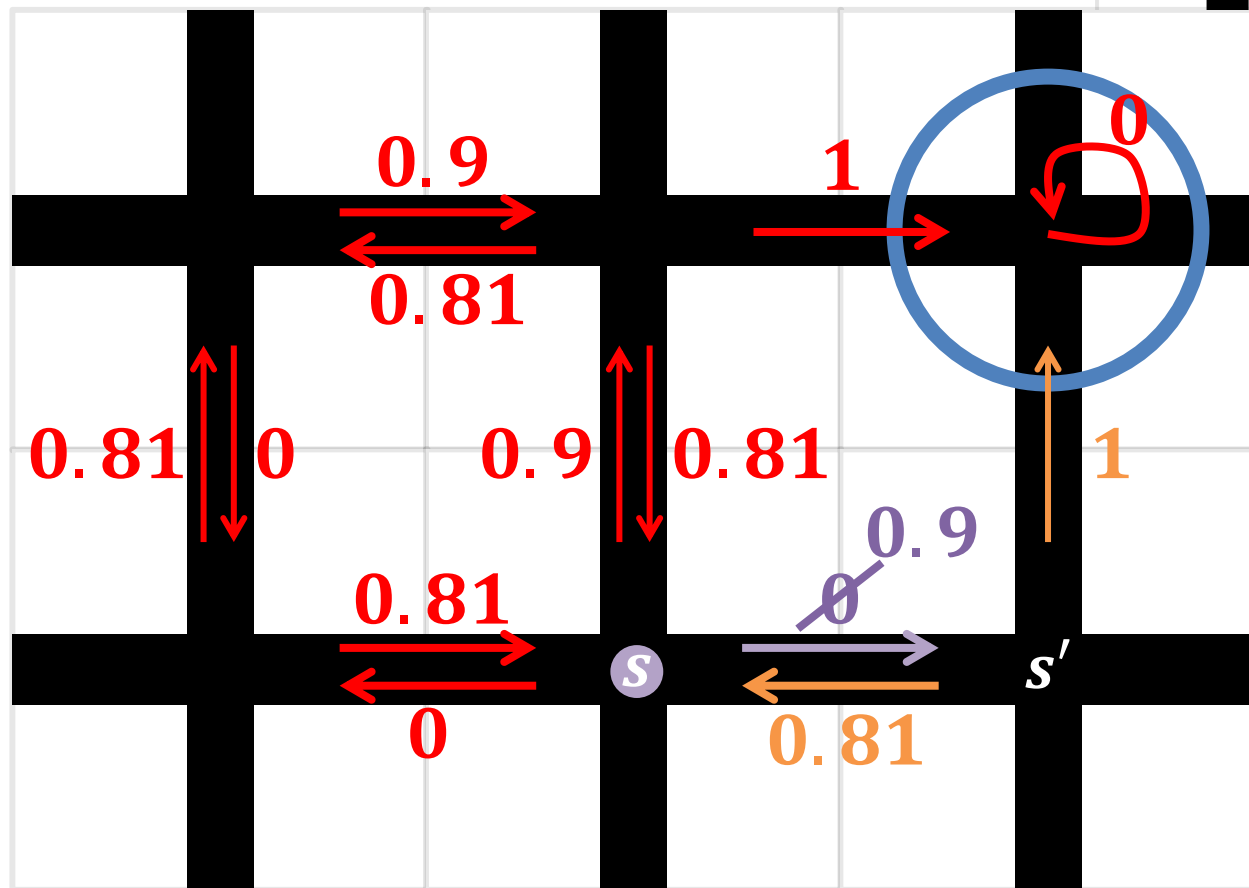
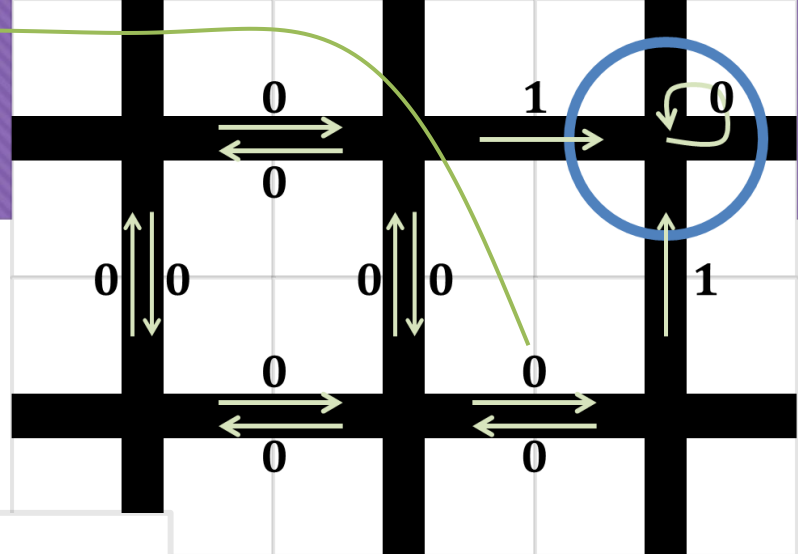
Q-러닝

$$\widehat{Q}(s, a) \leftarrow \underset{0}{\underset{0.9}{\mathbf{r} + \mathbf{0.9} \times \max_{a'} \widehat{Q}(s', a')}} \quad \text{with } \mathbf{0.9} \leftarrow \gamma$$



Q-러닝

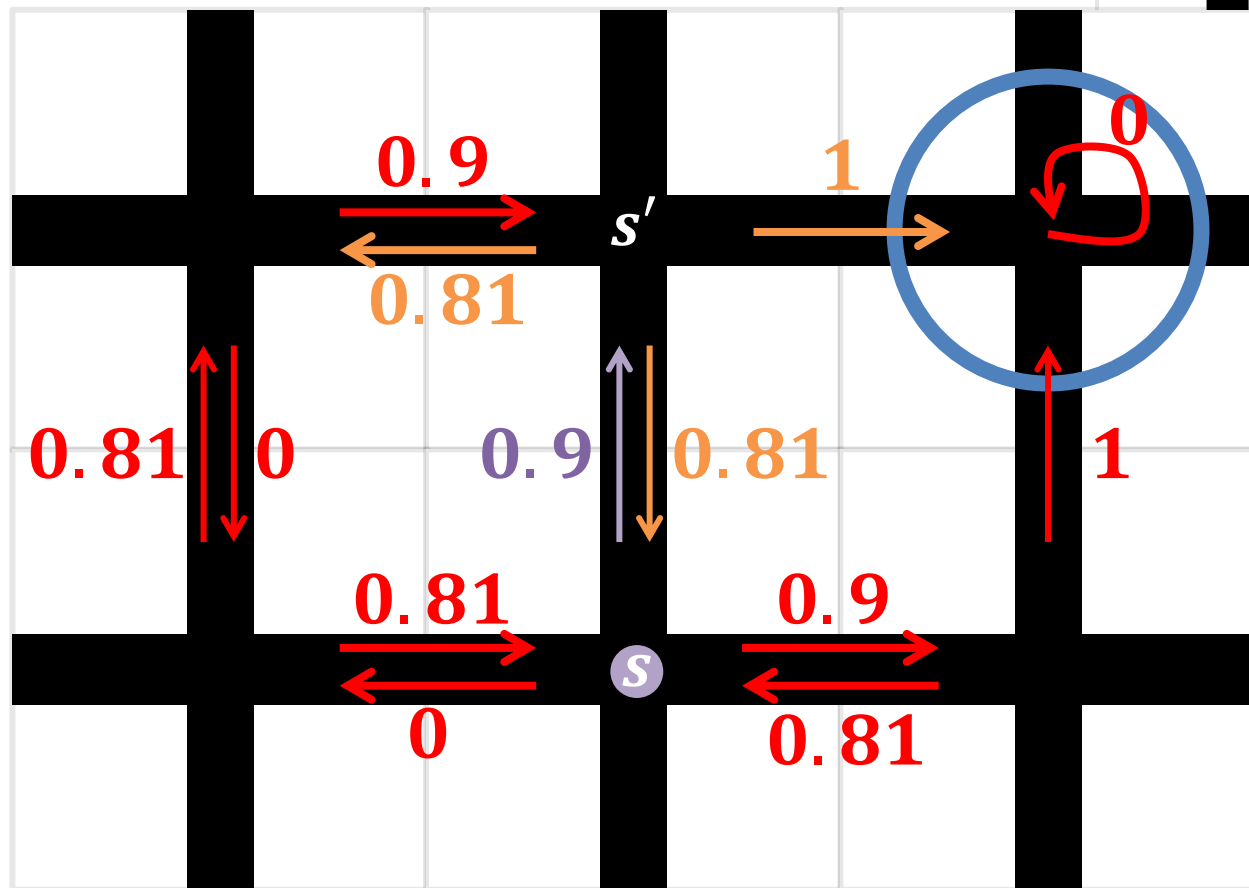
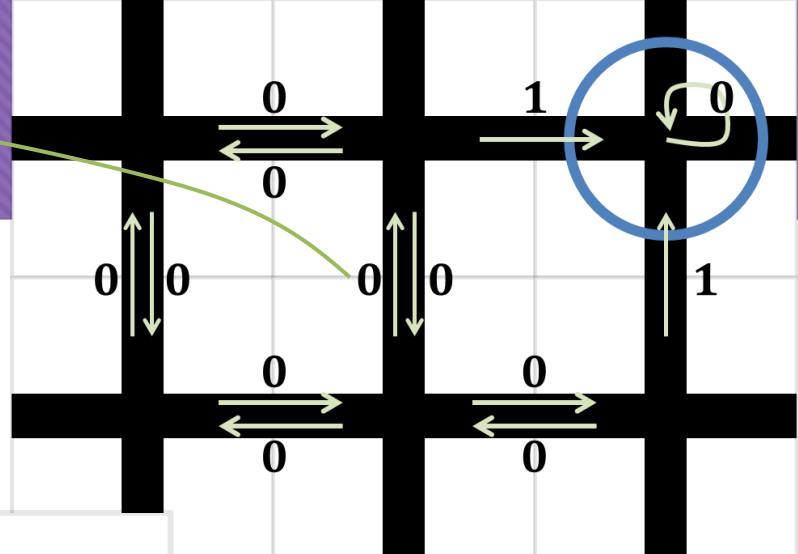
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

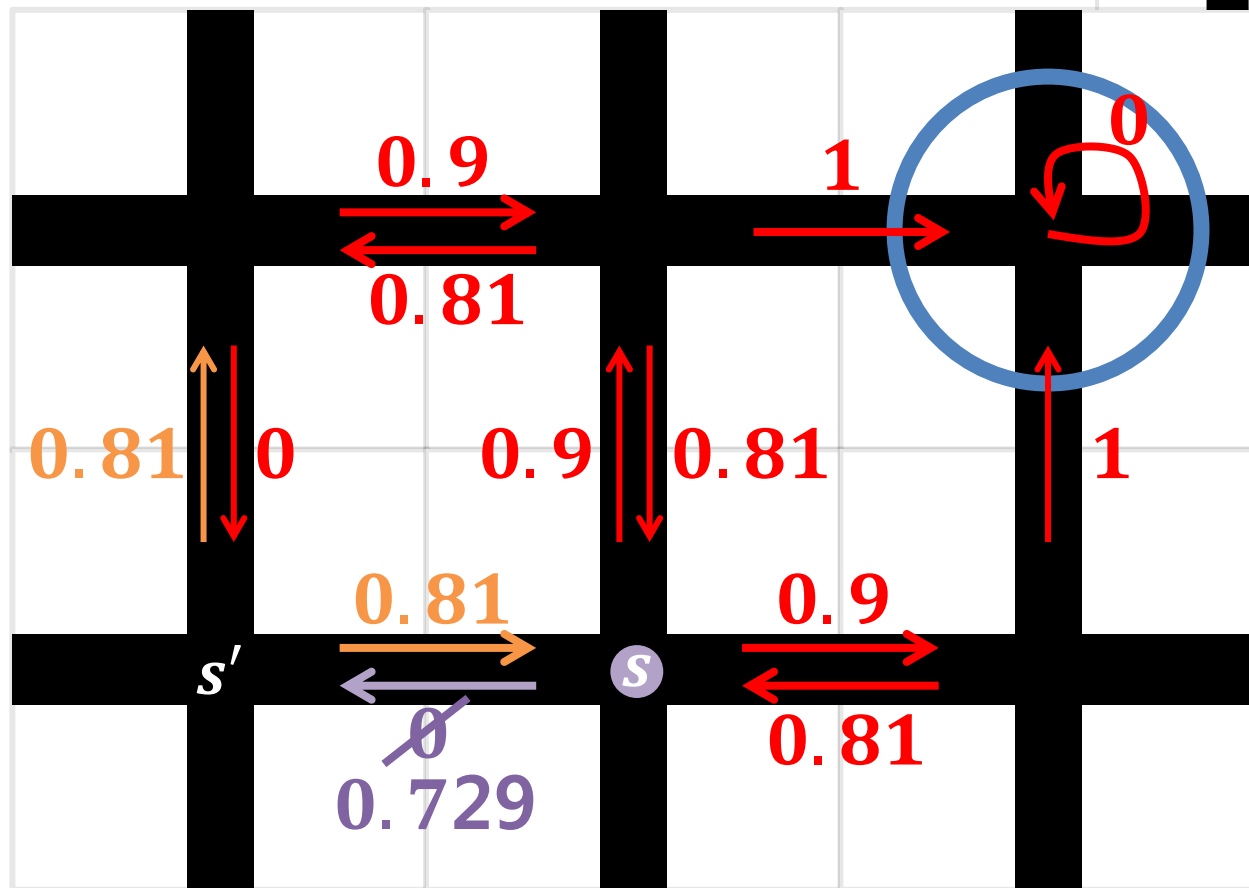
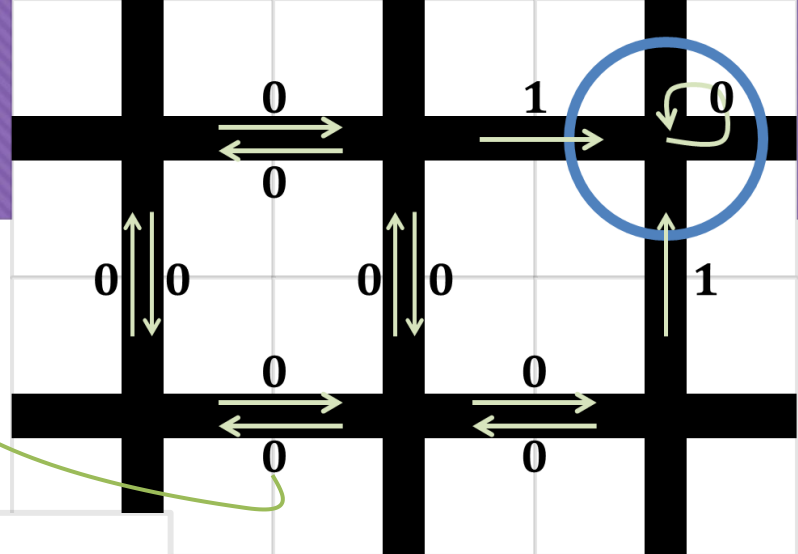
0 1



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

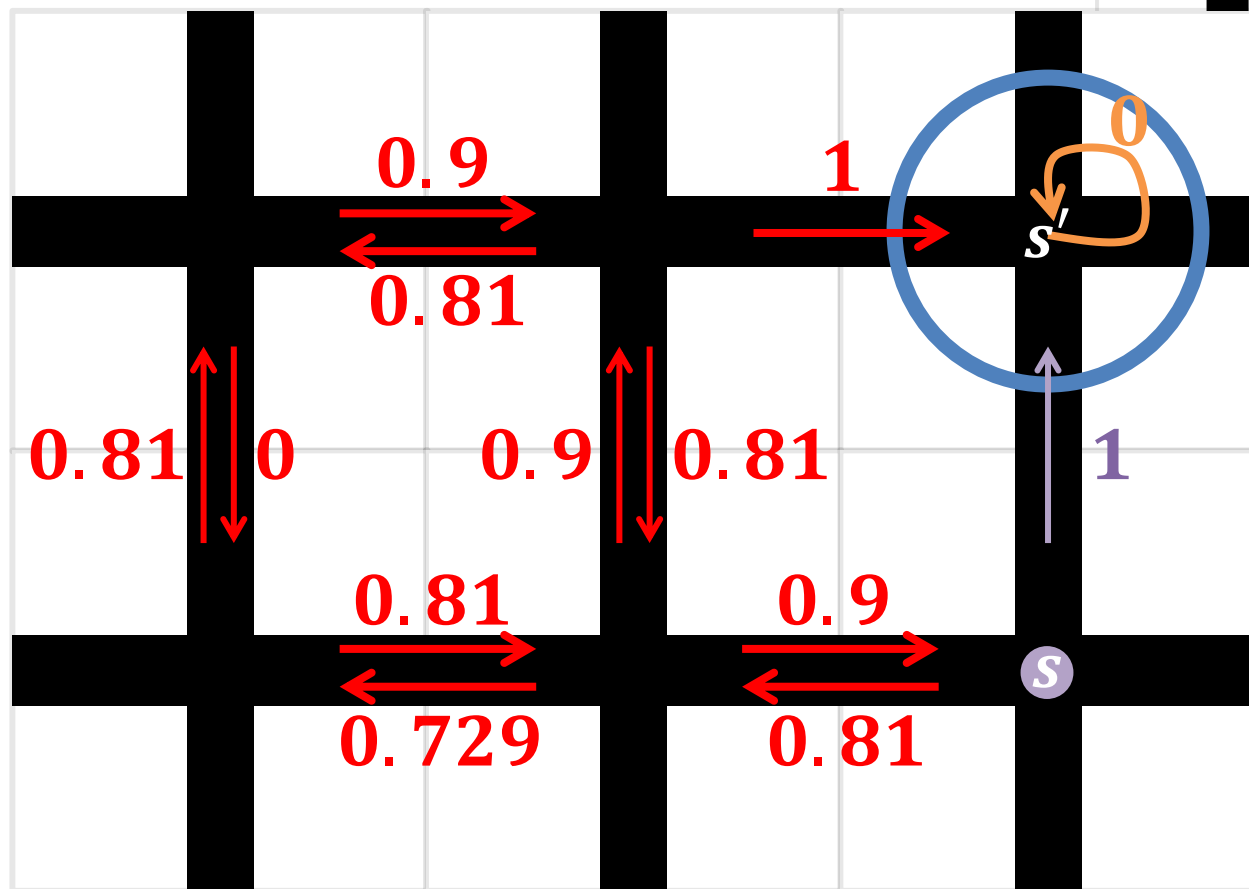
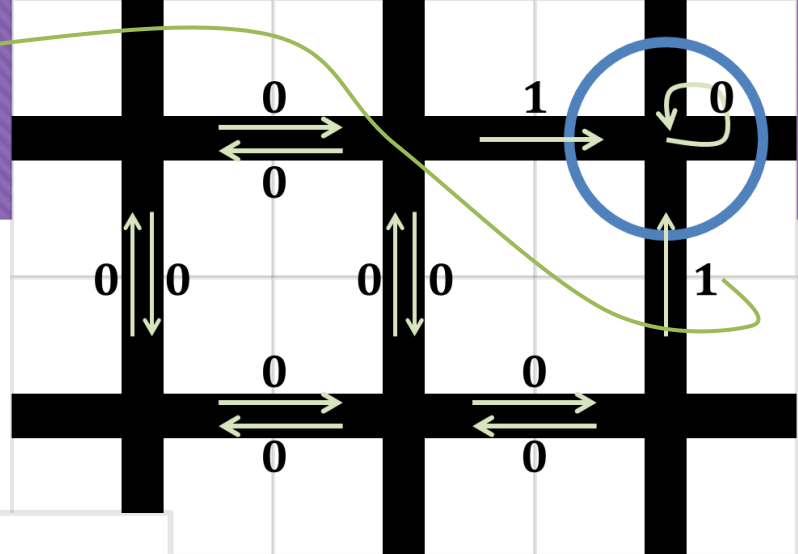
0 0.81



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

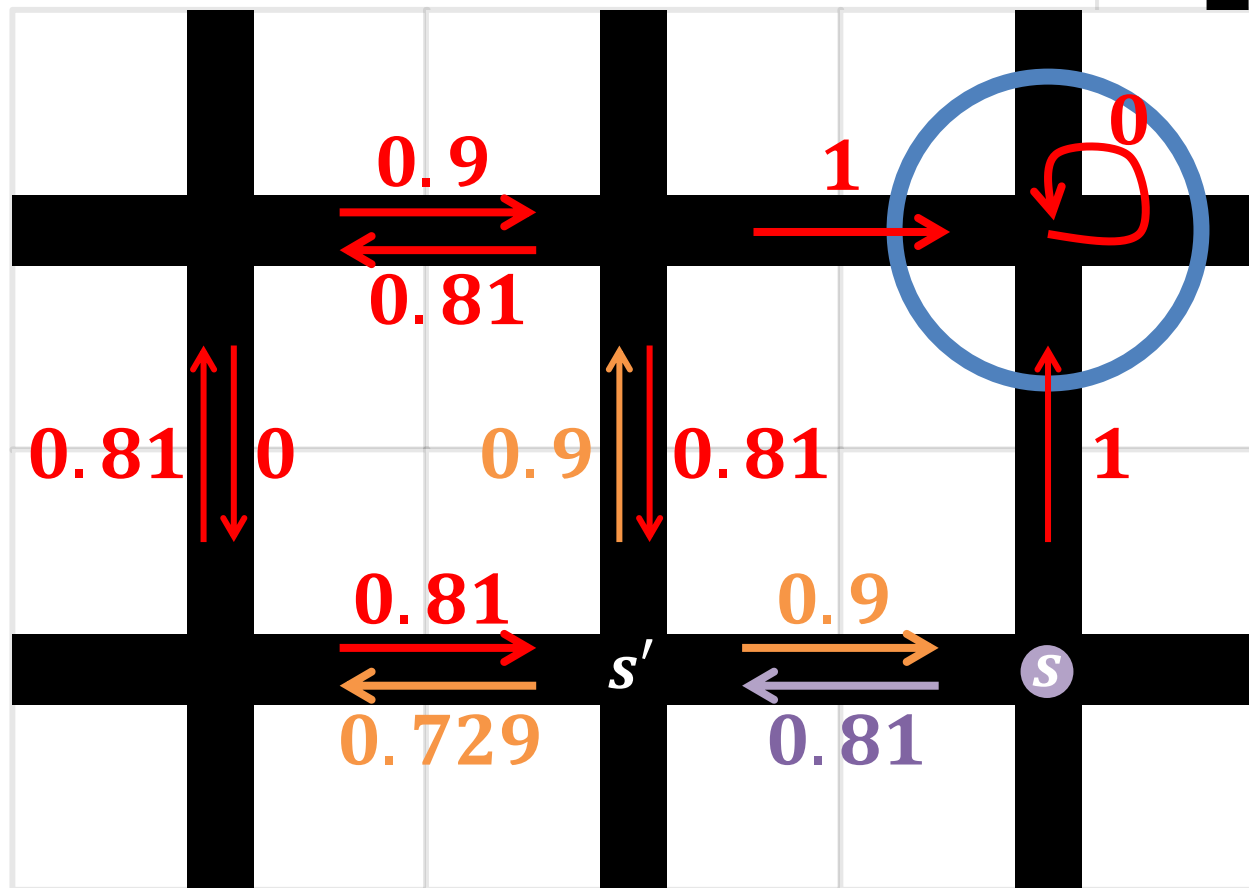
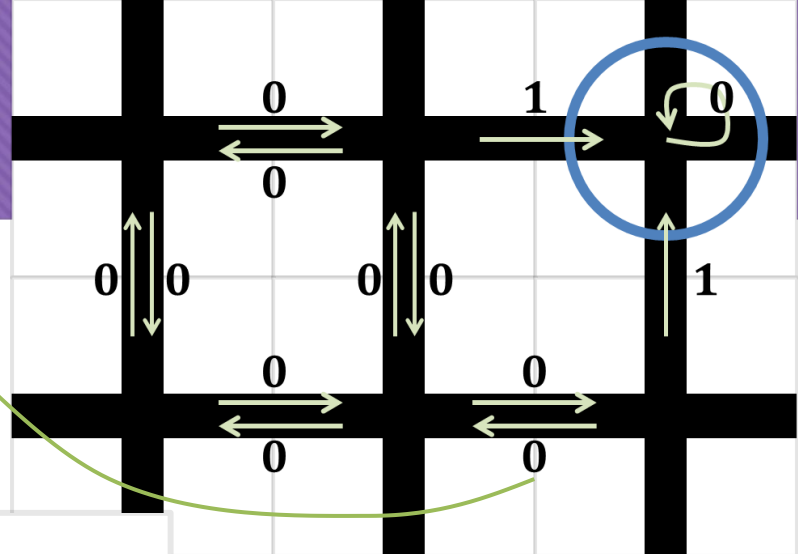
1 0



Q-러닝

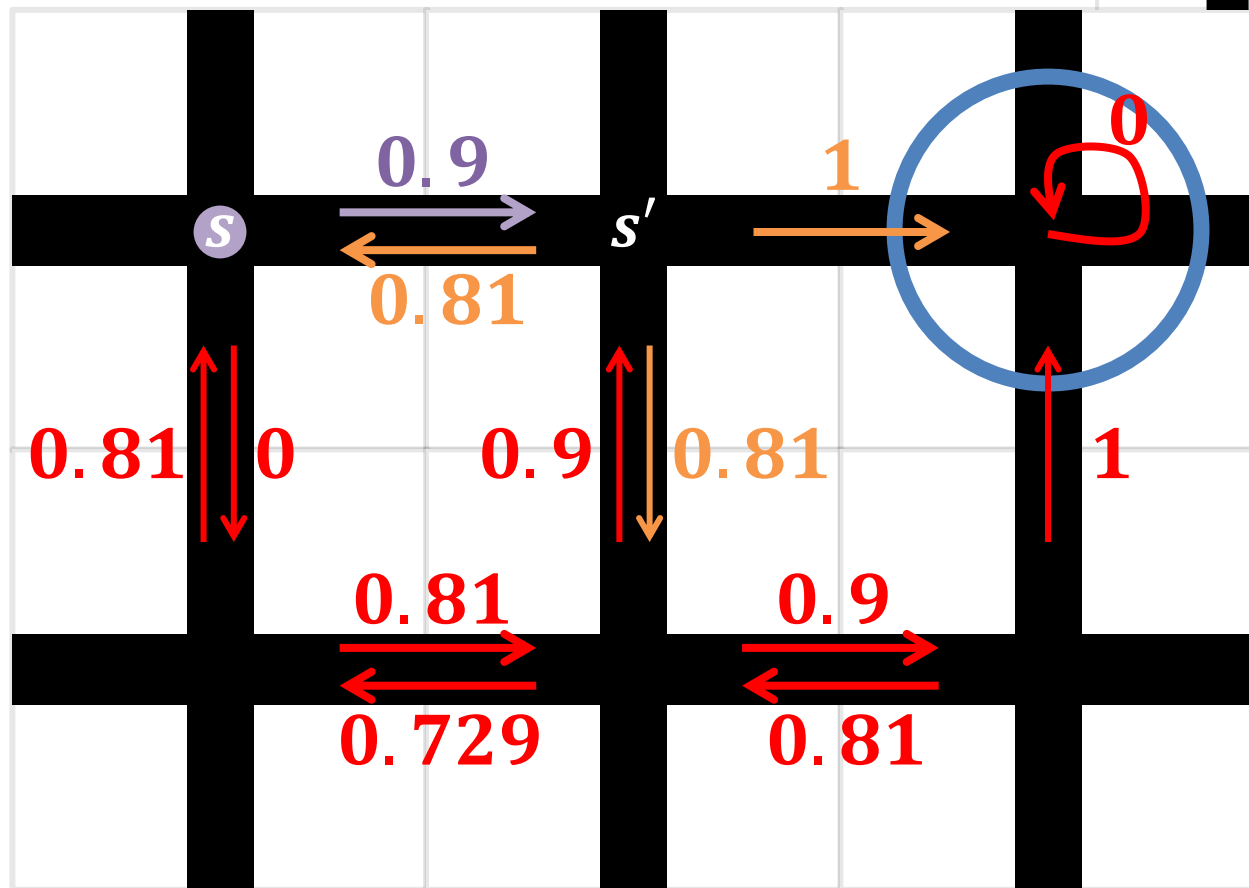
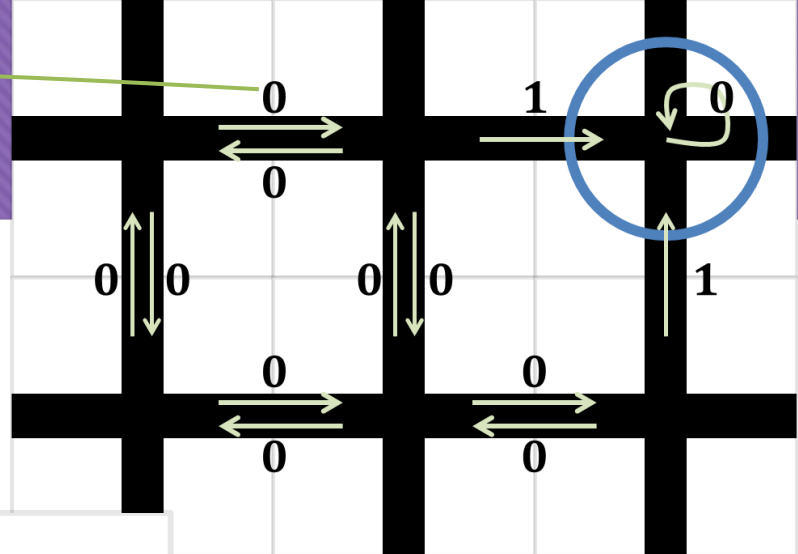
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



Q-러닝

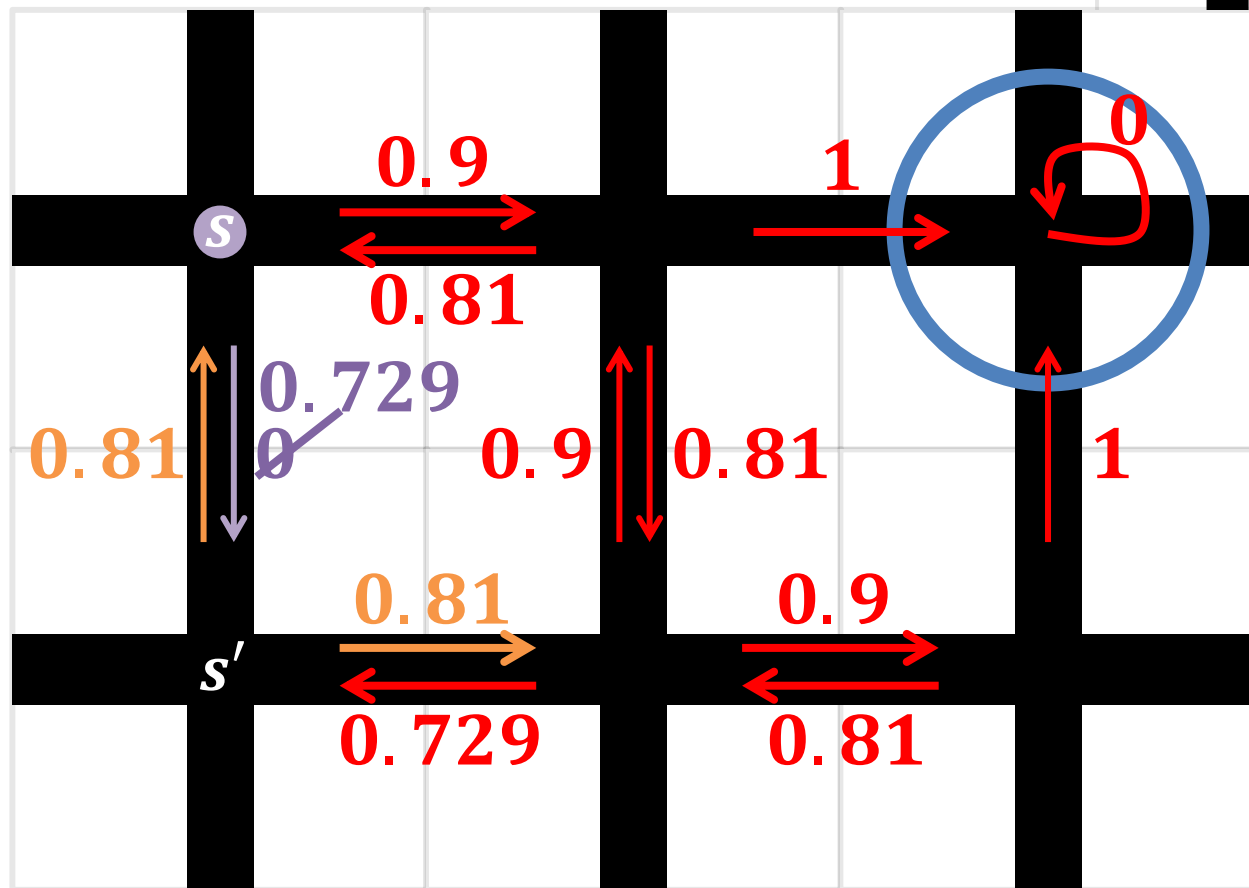
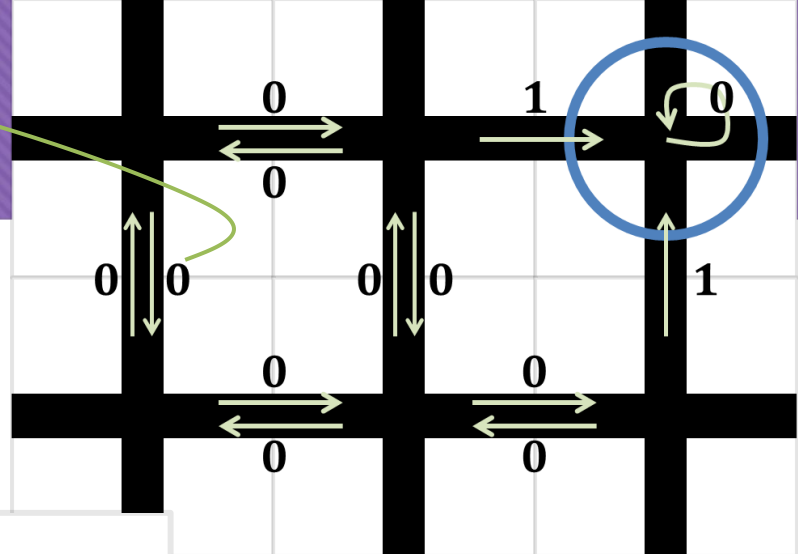
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

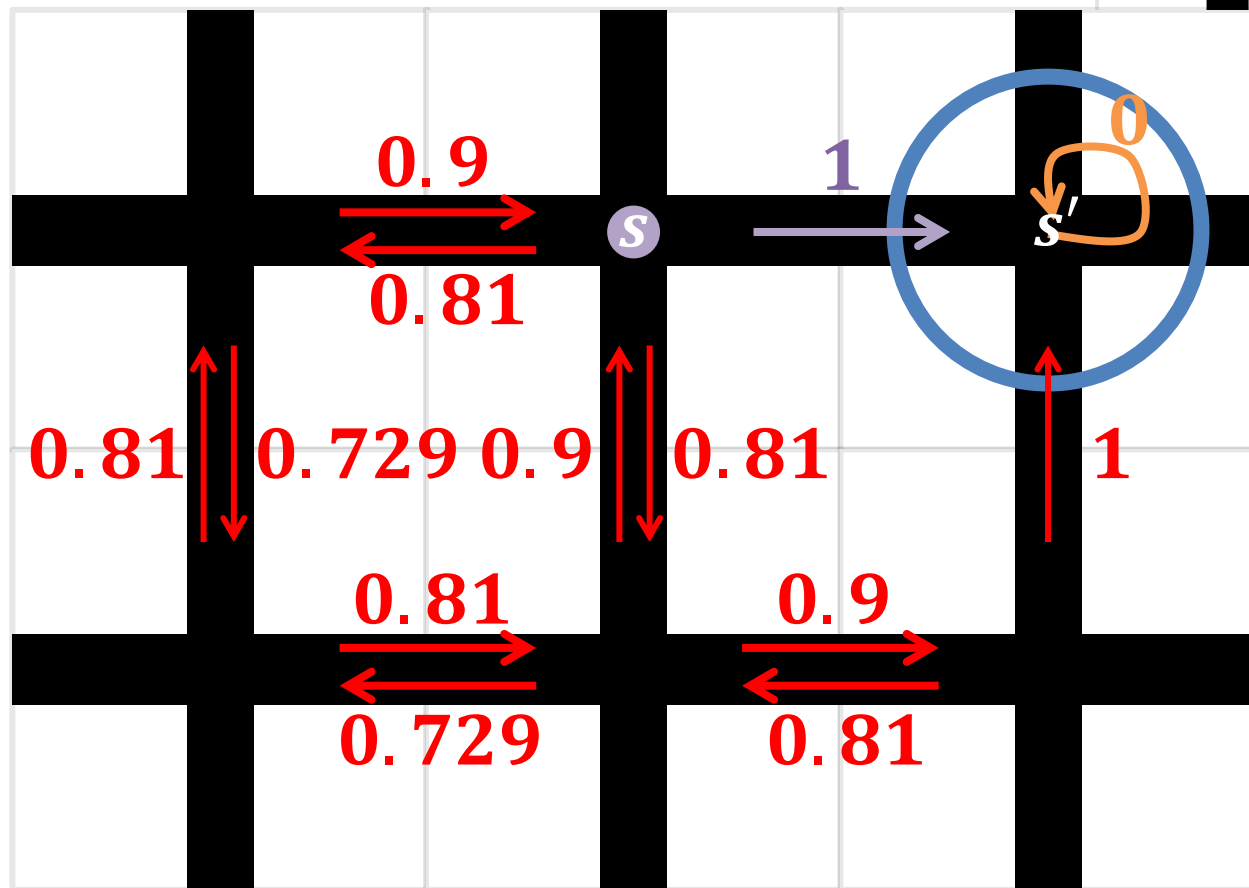
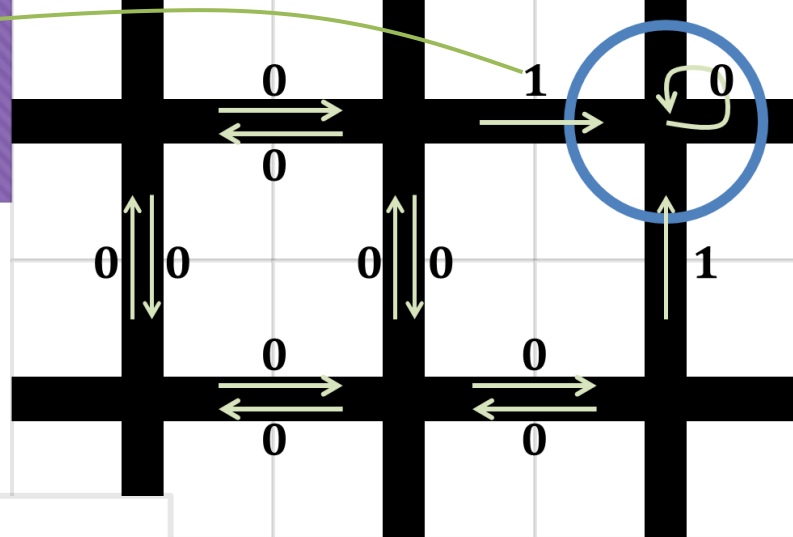
0 0.81



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

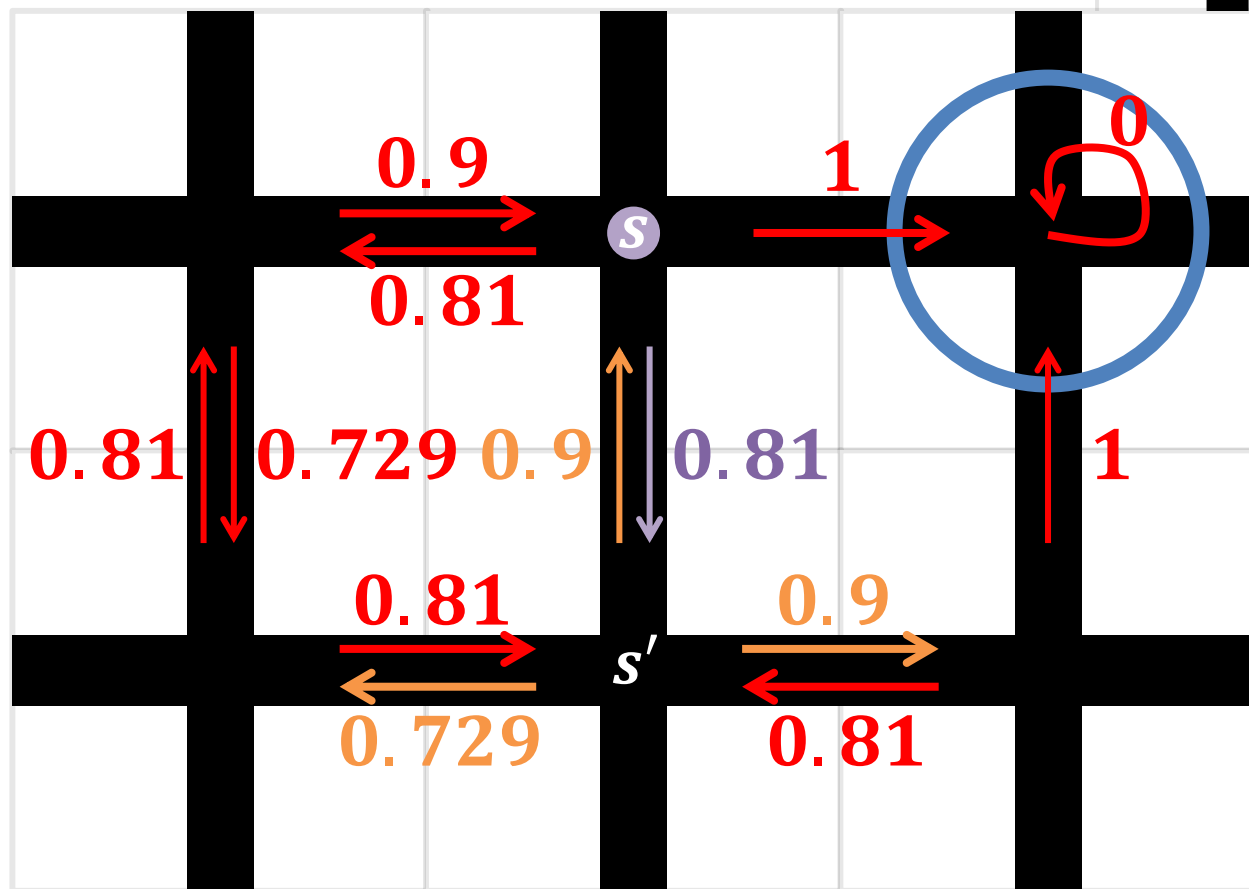
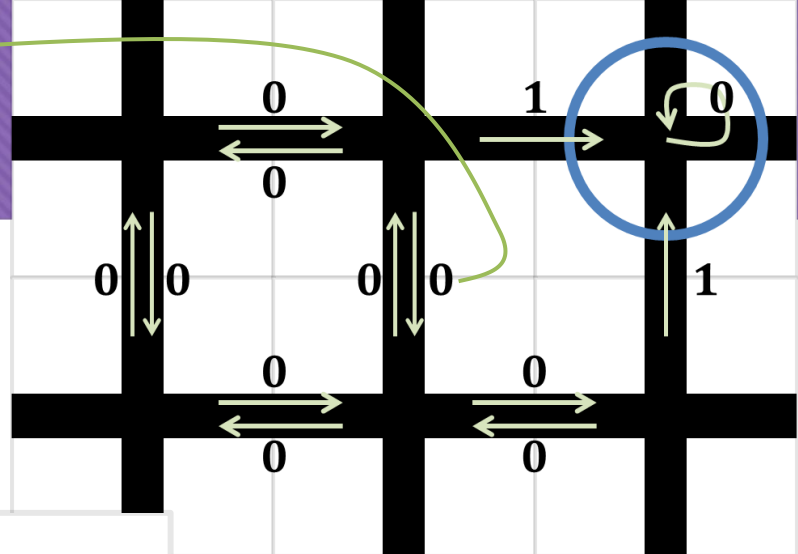
1 0



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

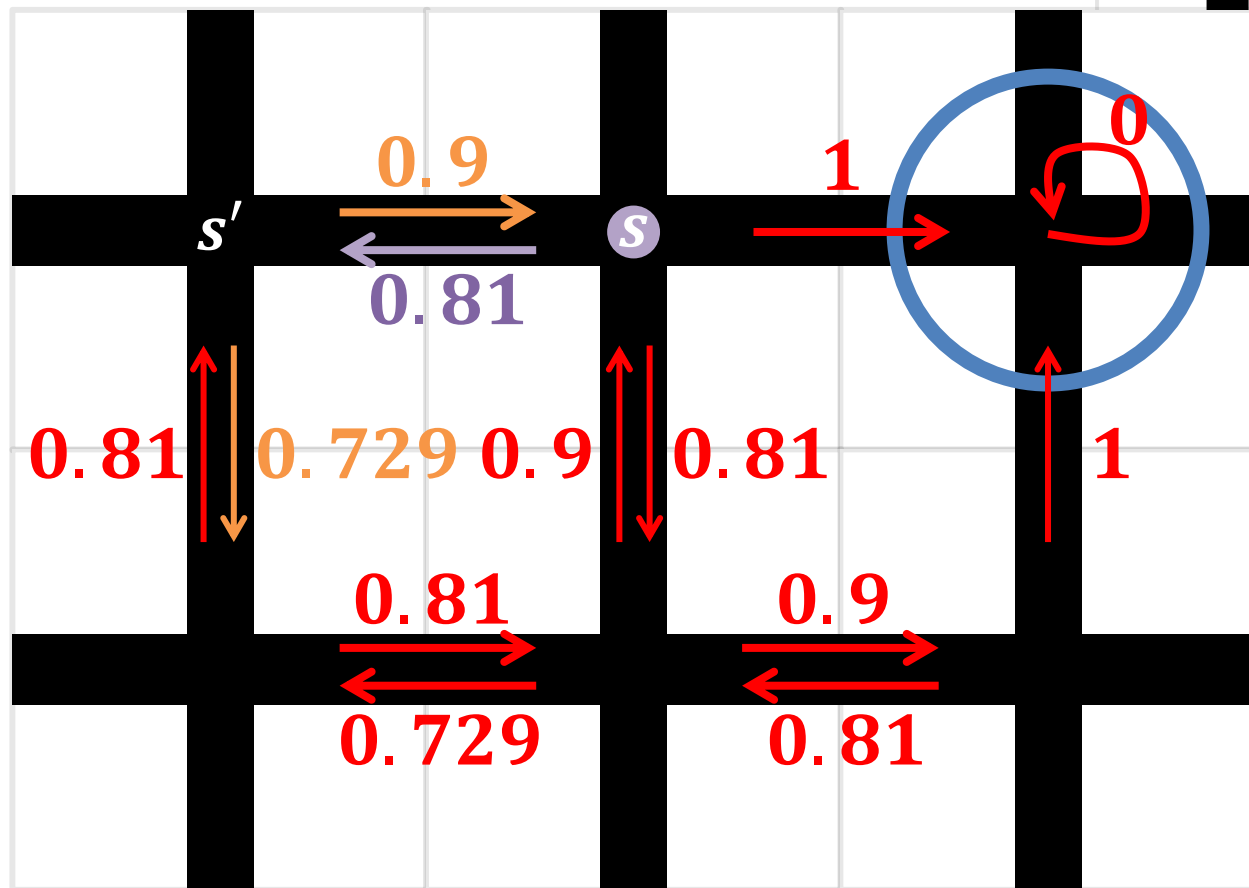
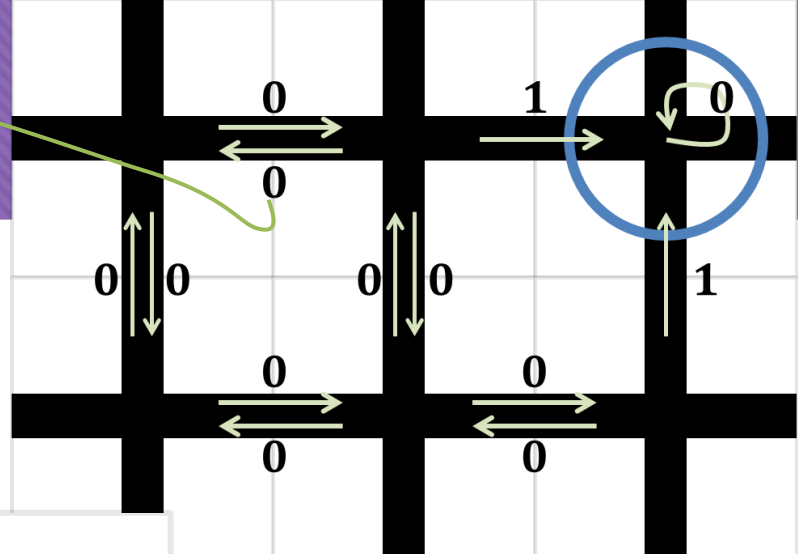
0 0.9



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

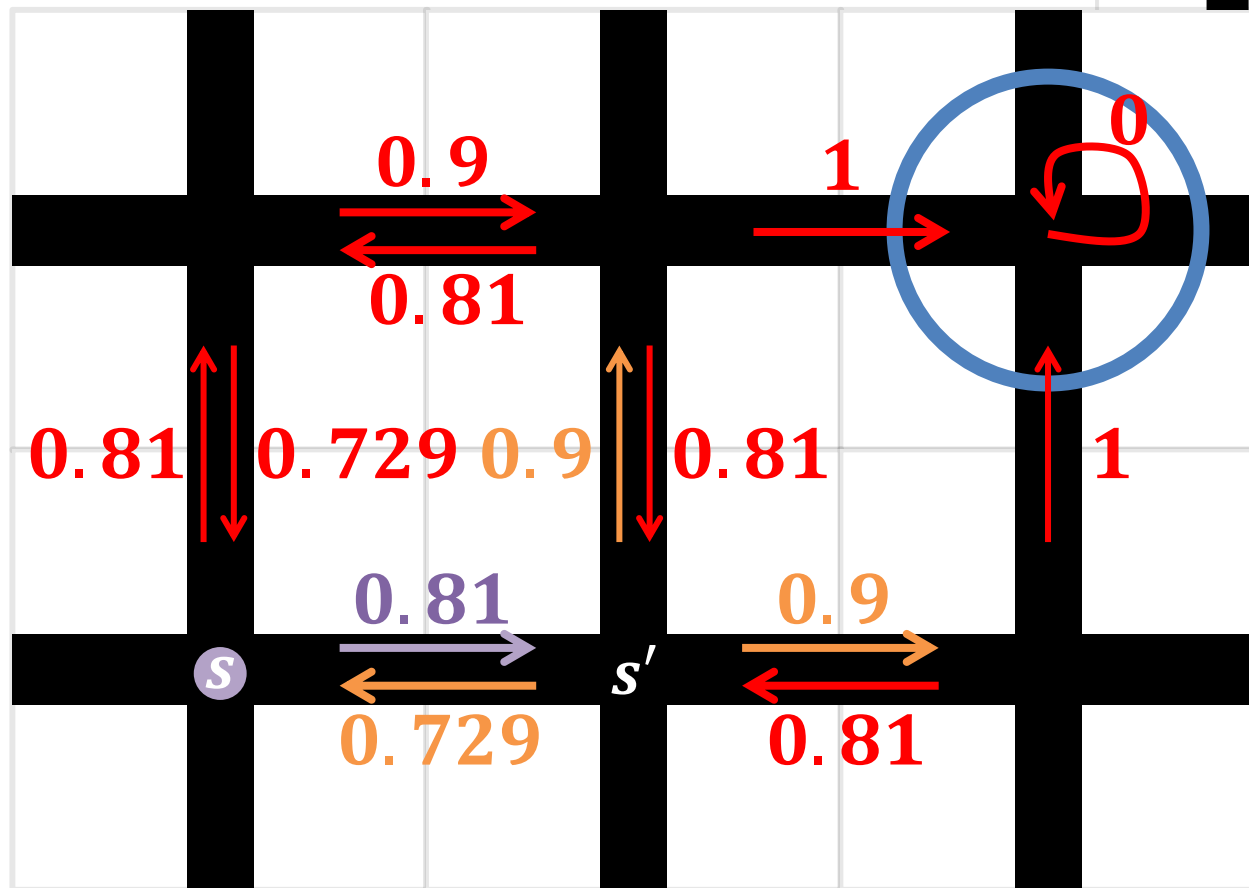
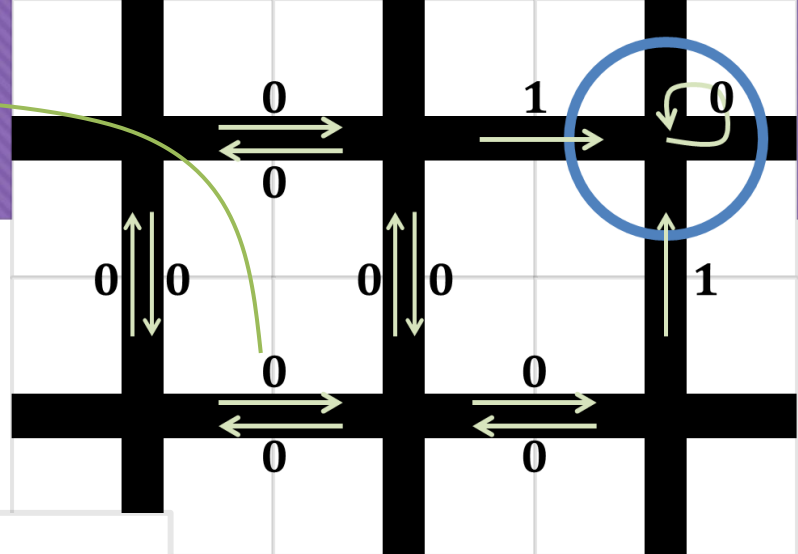
0 0.9



Q-러닝

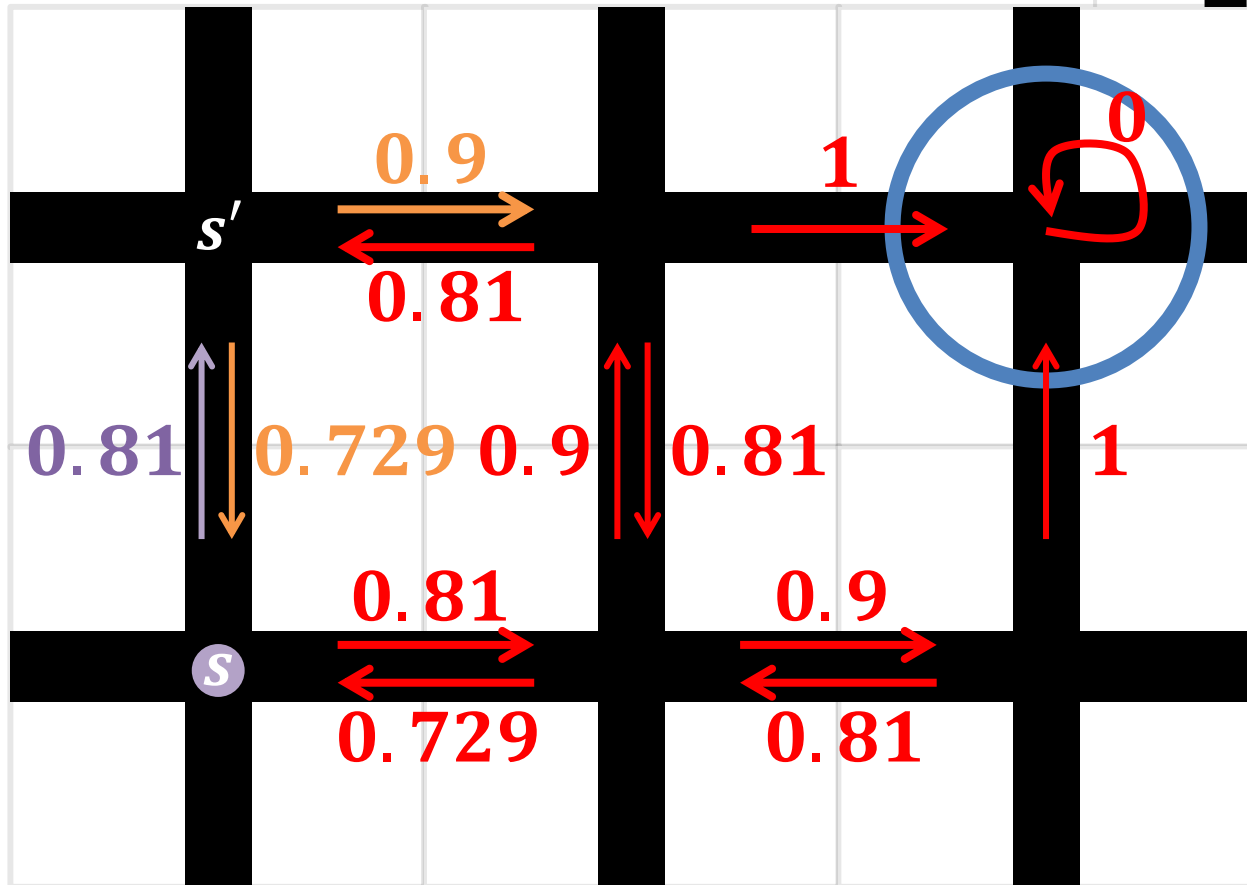
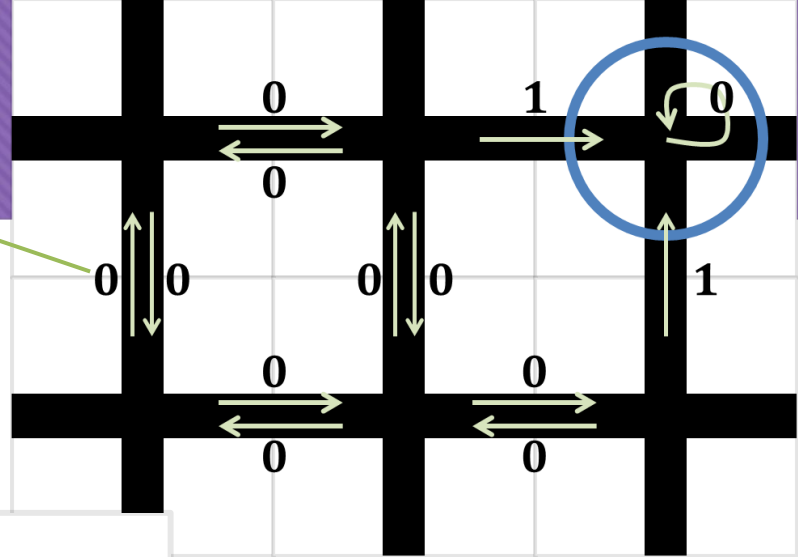
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



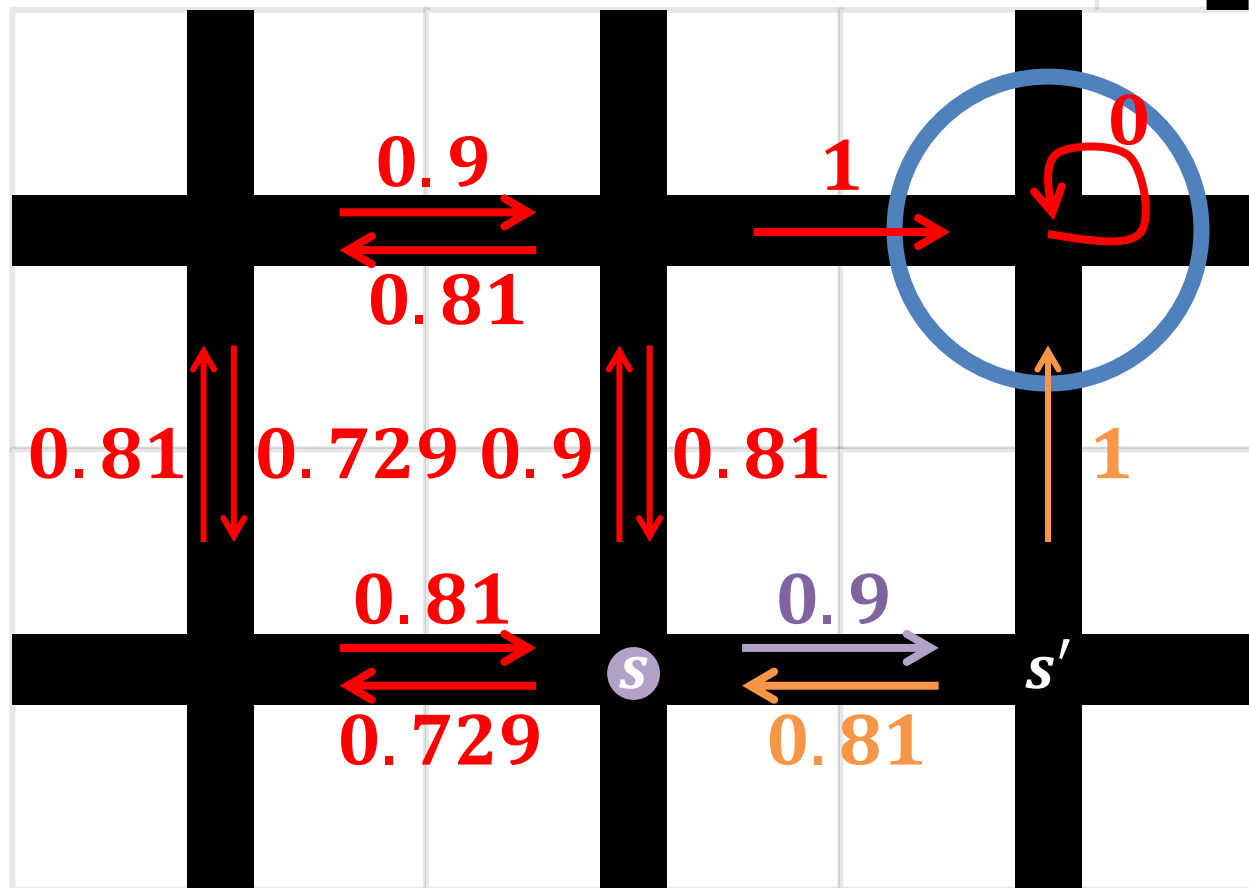
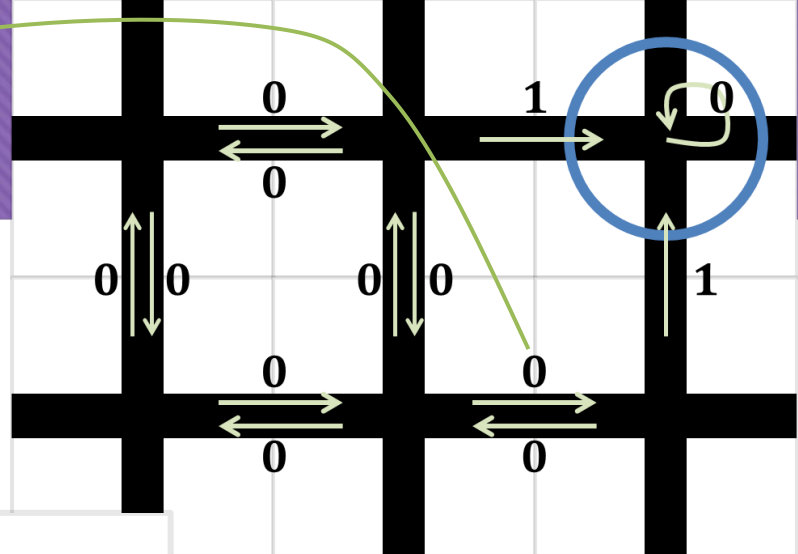
Q-러닝

$$\widehat{Q}(s, a) \leftarrow \underset{0}{\underset{0.9}{\mathbf{r} + \mathbf{0.9} \times \max_{a'} \widehat{Q}(s', a')}} \quad \text{with } \mathbf{0.9} \leftarrow \gamma$$



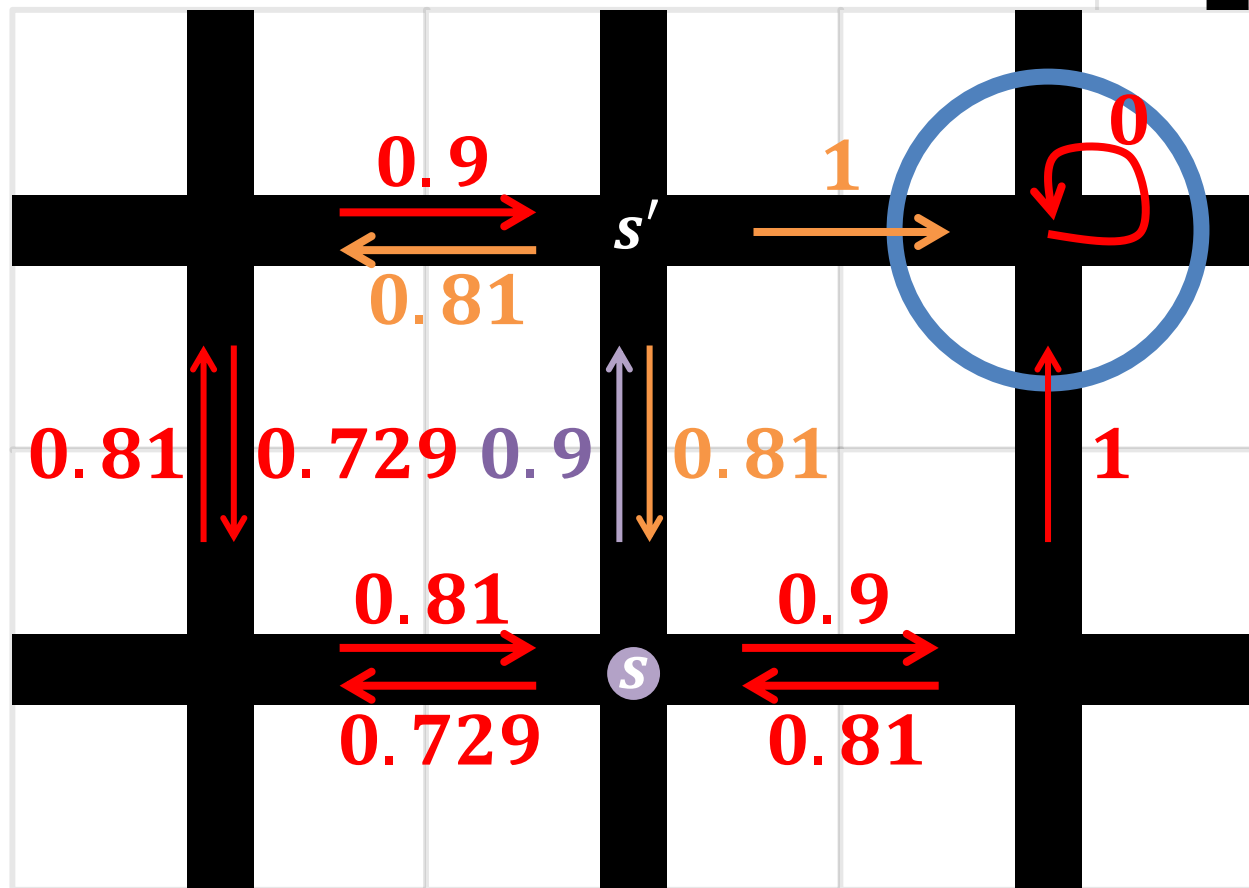
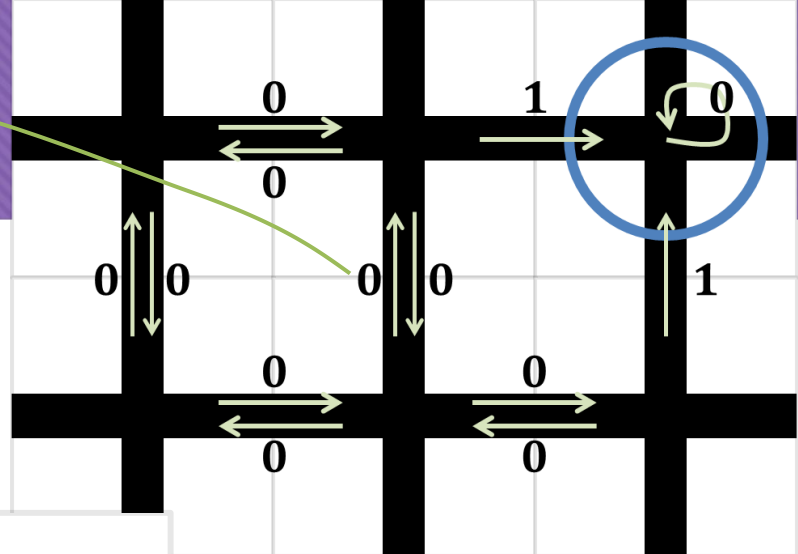
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

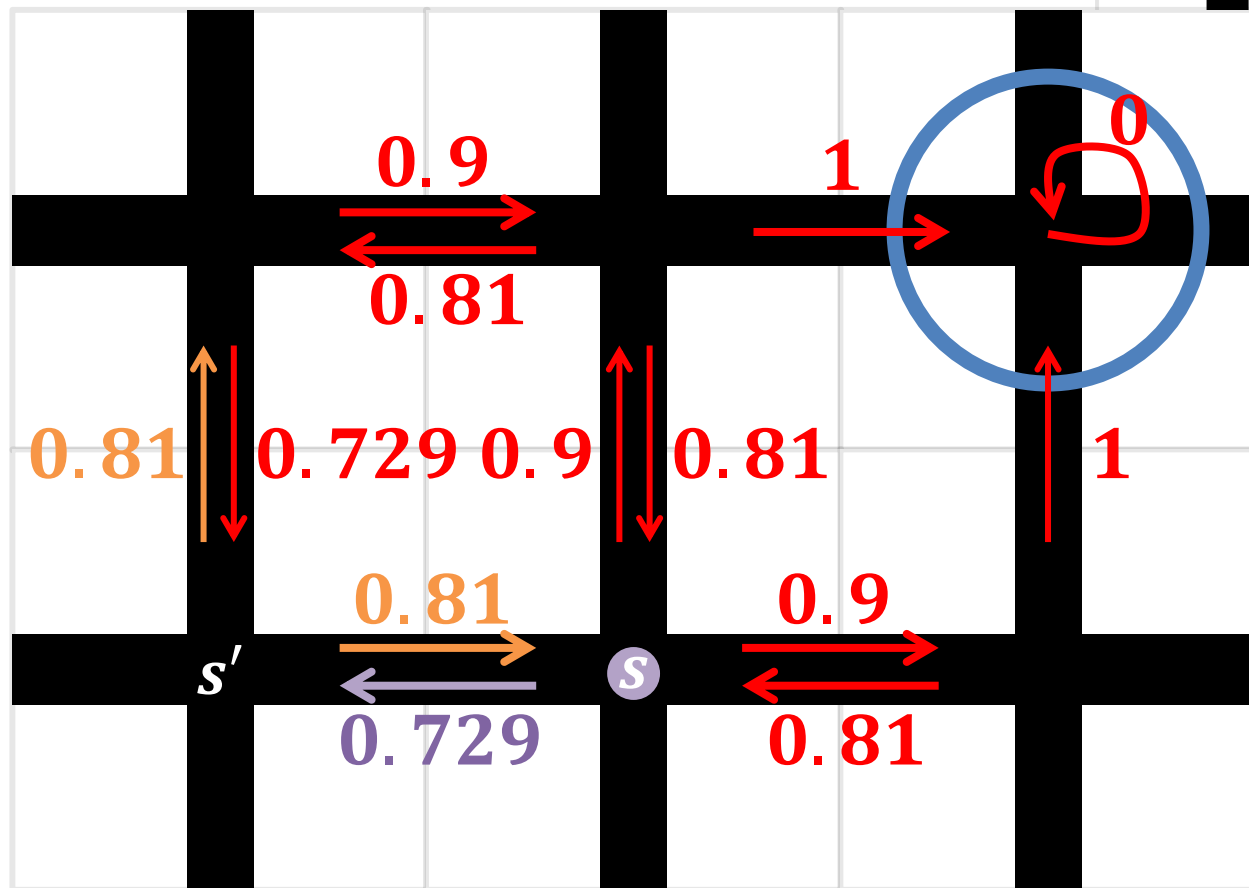
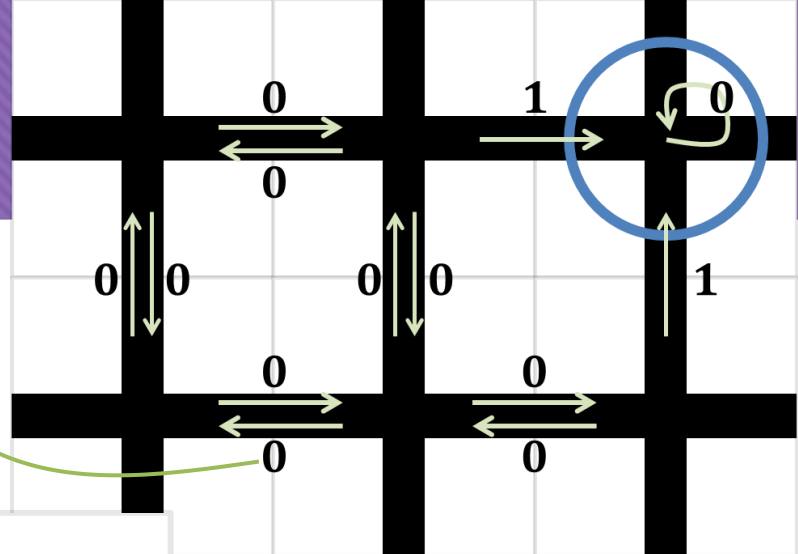
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

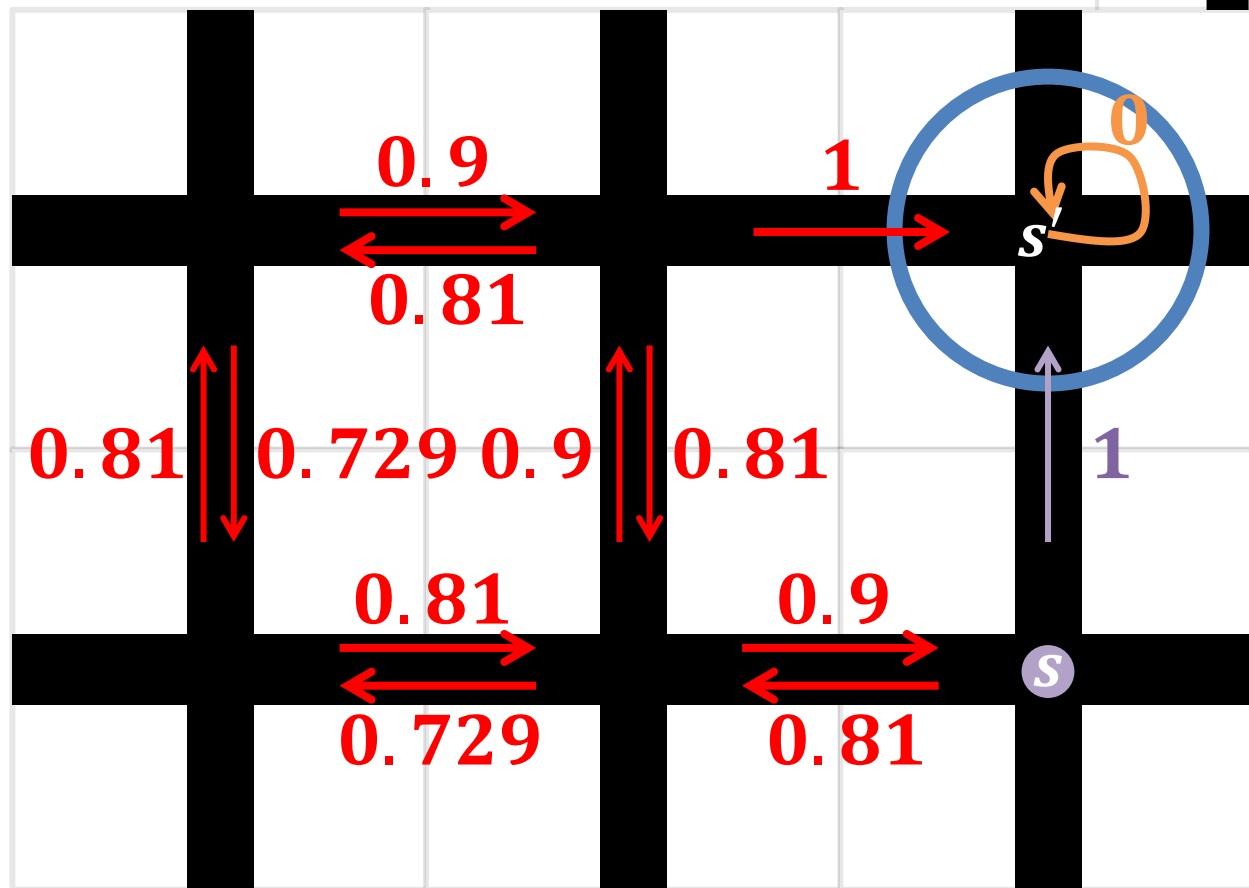
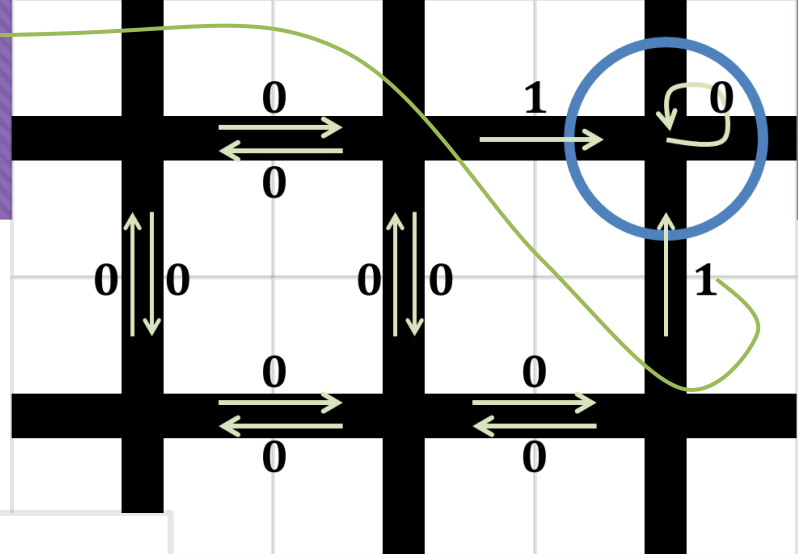
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.81



Q-러닝

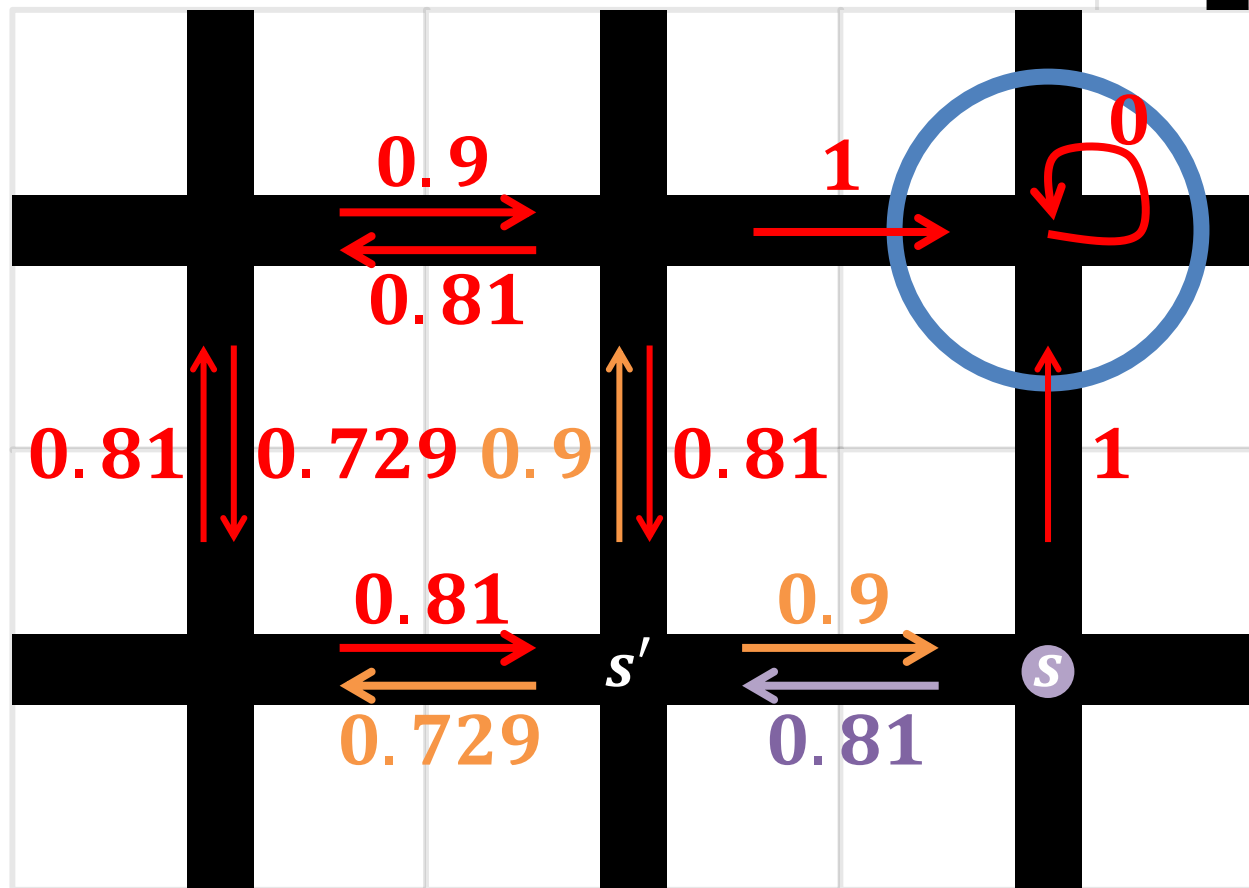
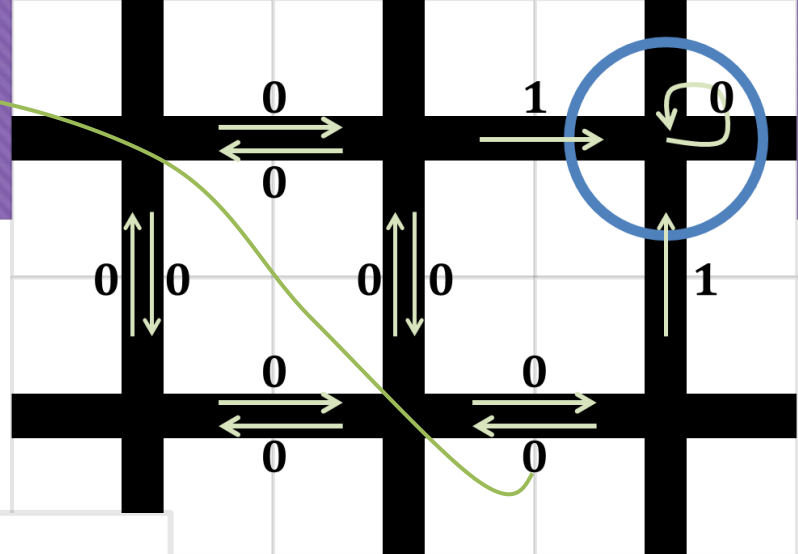
$$\widehat{Q}(s, a) \leftarrow \underset{1}{\mathbf{r}} + \underset{0}{\mathbf{0.9}} \times \max_{a'} \widehat{Q}(s', a')$$



Q-러닝

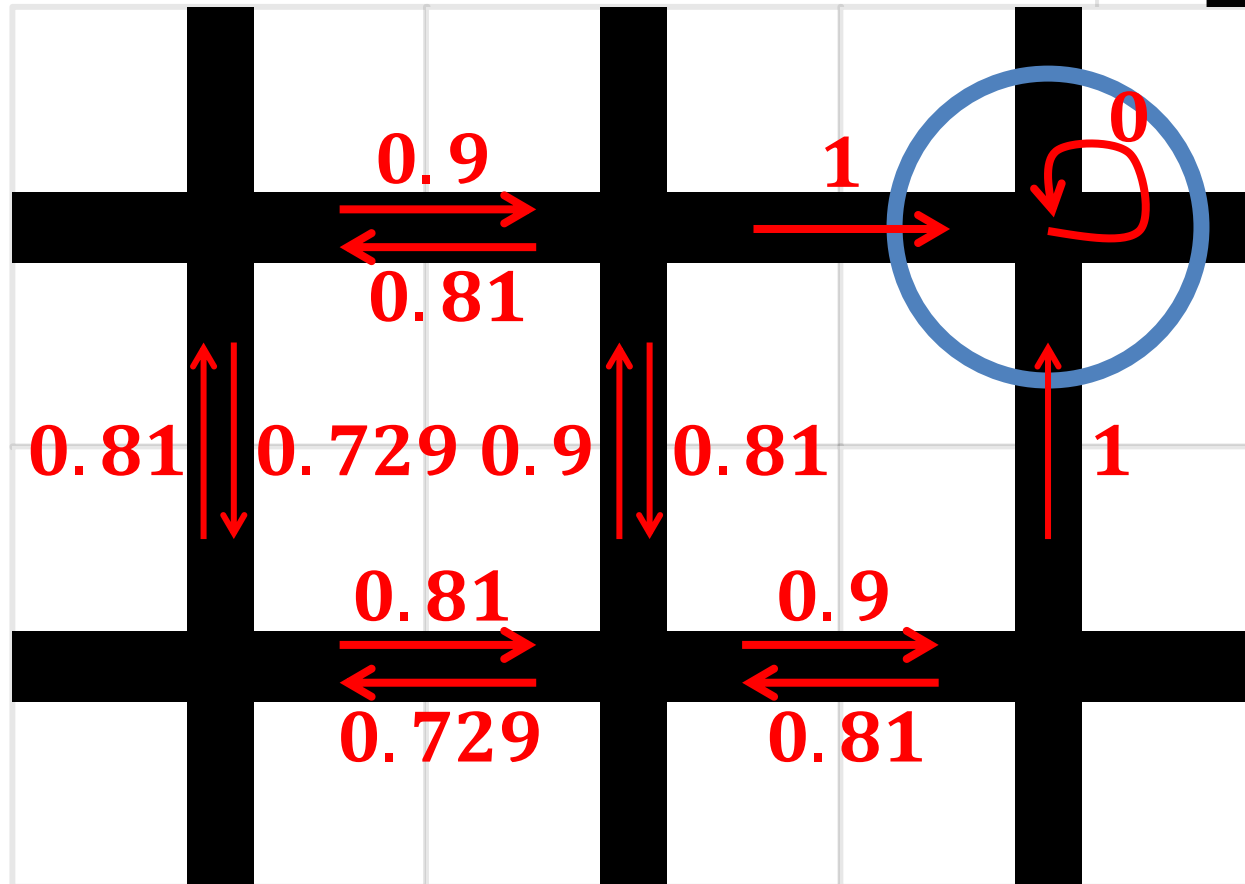
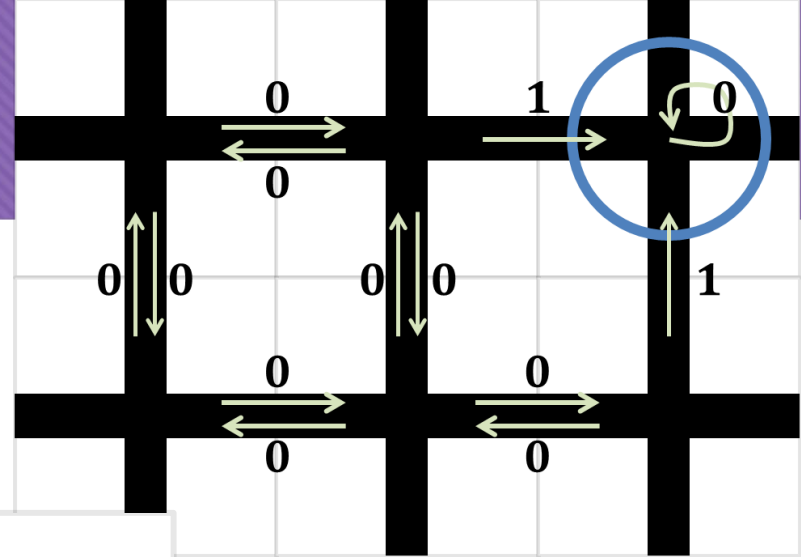
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



Q-러닝

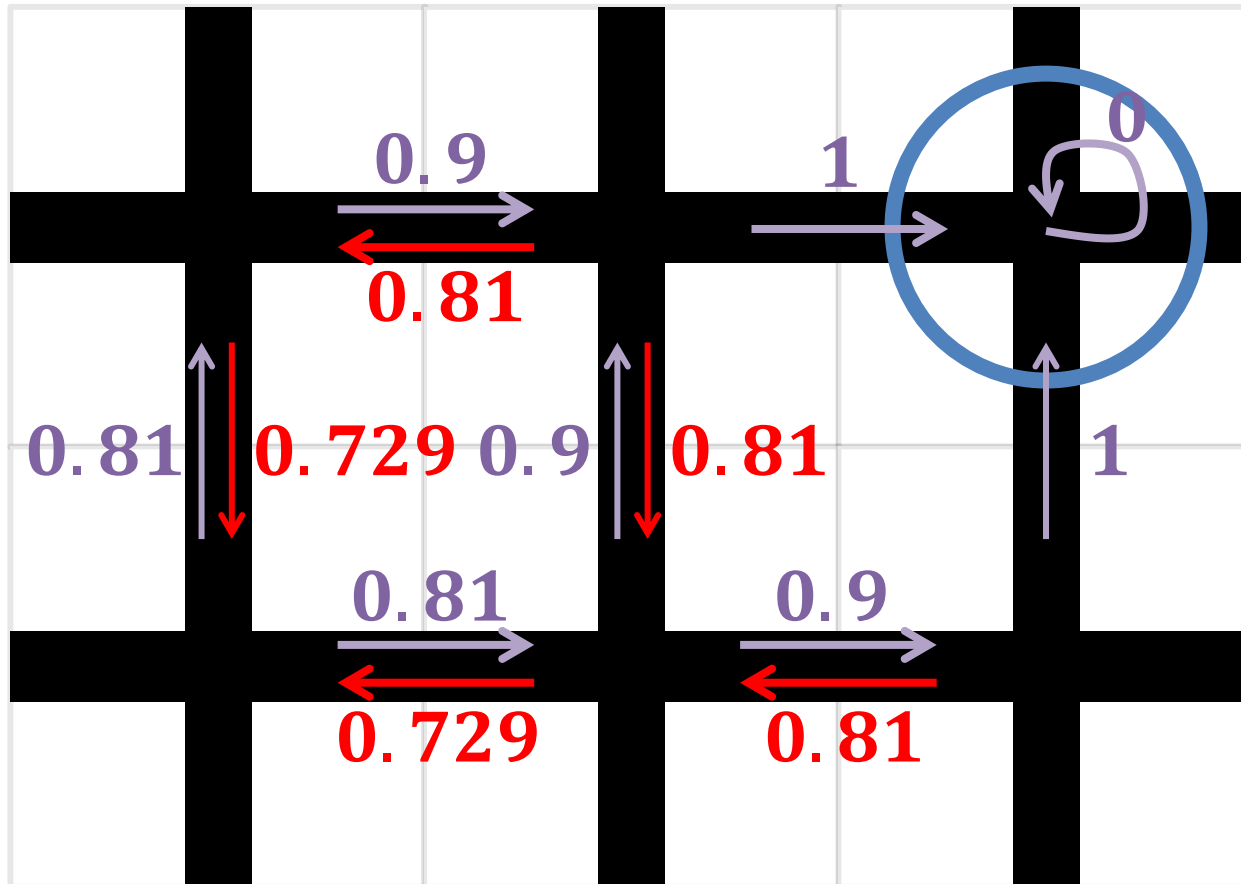
변화가 없을 때까지 계속 반복



$$\pi^*(s) = \operatorname{argmax}_a Q(s, a)$$

최적의 정책(policy) $\pi^*(s)$

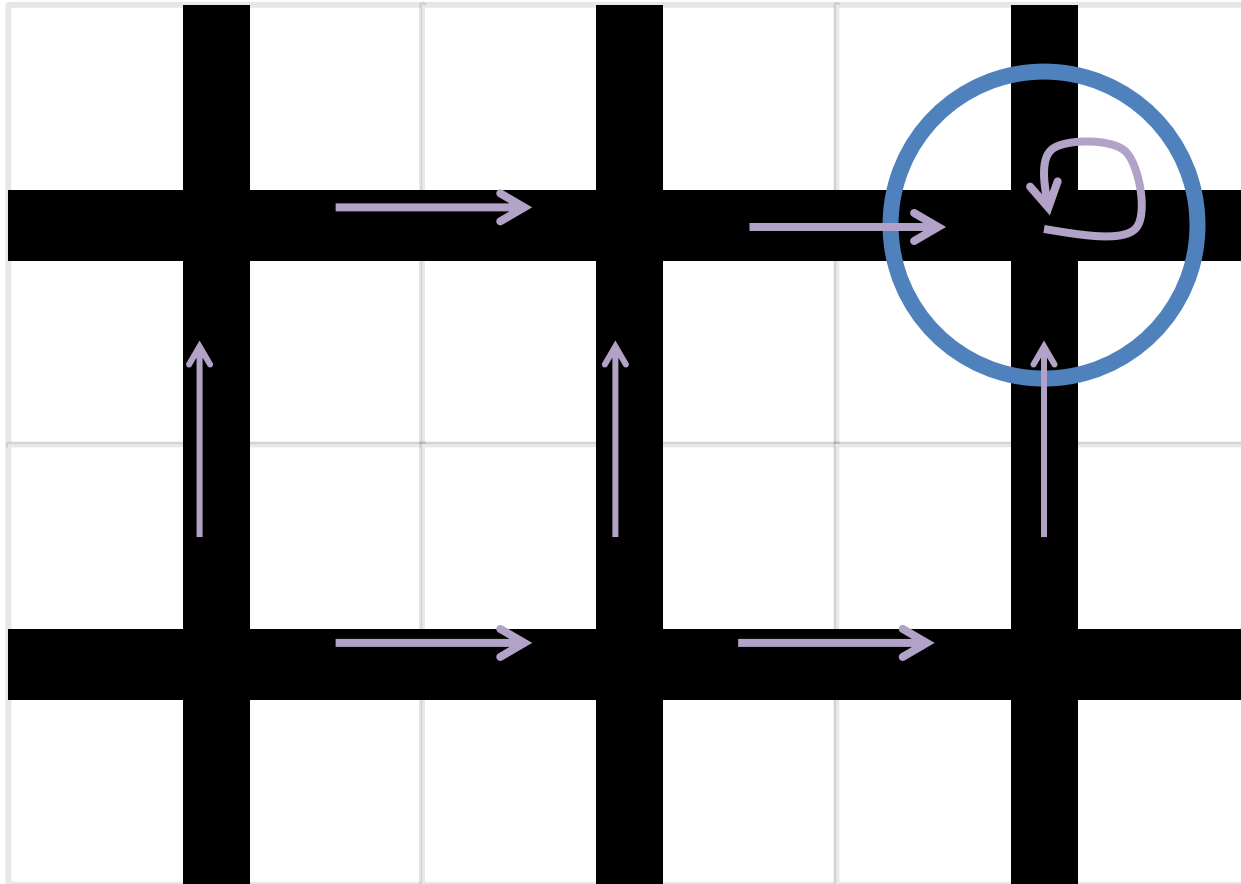
각 상태에서 Q값이 가장 큰 행동



최적의 정책(policy) $\pi^*(s)$

각 상태에서 Q값이 가장 큰 행동

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} Q(s, a)$$



실제 상황에서는 학습을 완료한 후
행동하는 것이 아닐 수 있다.


$Q(s, a)$ 가 아니라 $\hat{Q}(s, a)$ 가 최대인 행동 선택

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

0.9



모든 상태에서의 모든 행동을 계속 경험하면
 $\hat{Q}(s, a) \rightarrow Q(s, a)$ 로 수렴



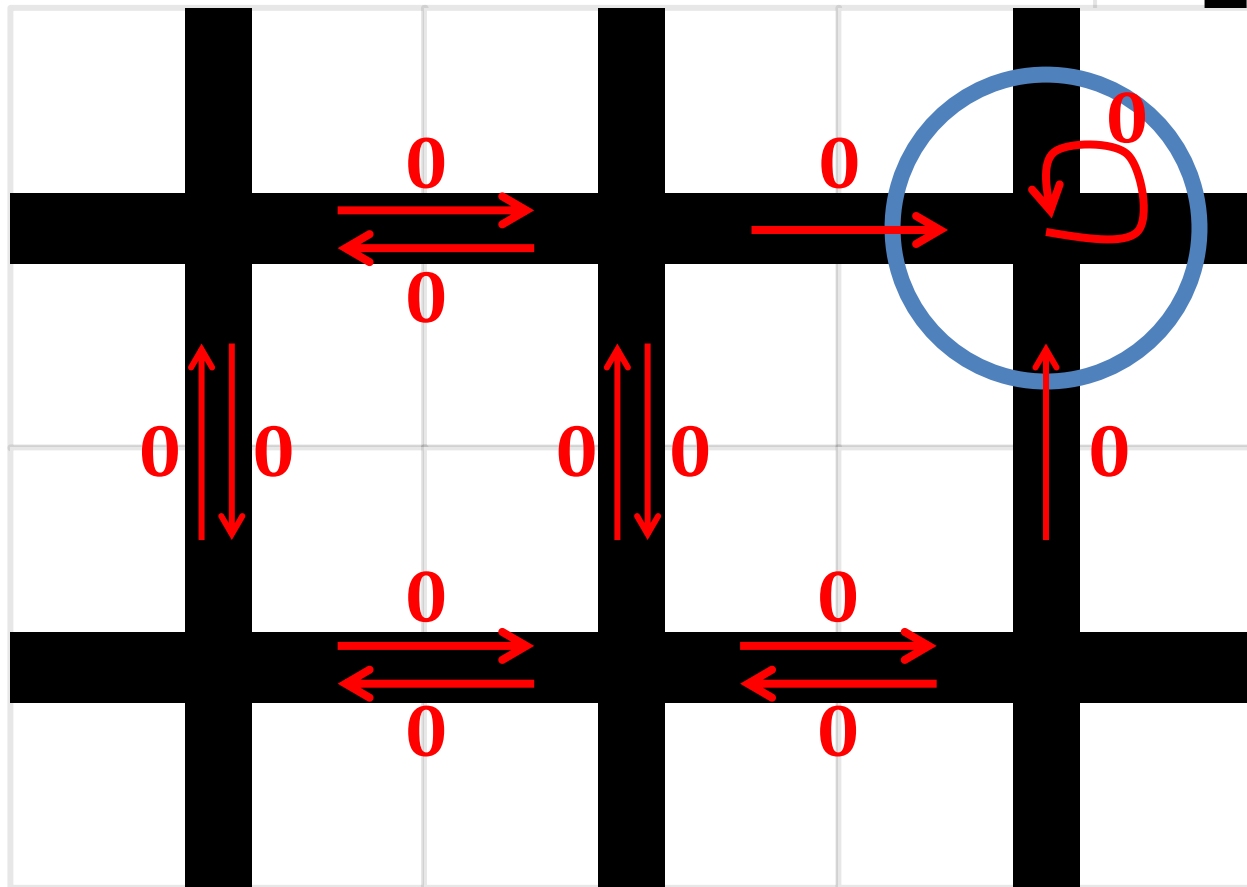
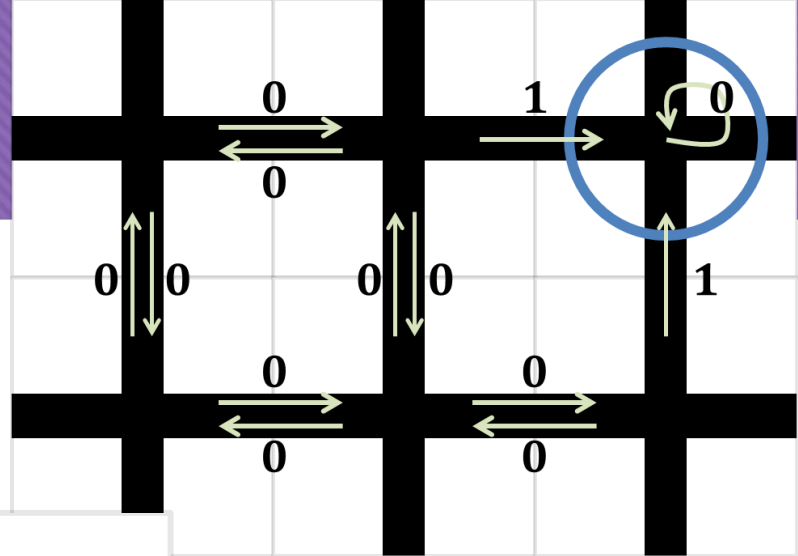
$$\pi^*(s) = \operatorname{argmax}_a Q(s, a)$$

각 상태에서 $Q(s, a)$ 가 최대인 행동(a)을
취하면 됨

Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

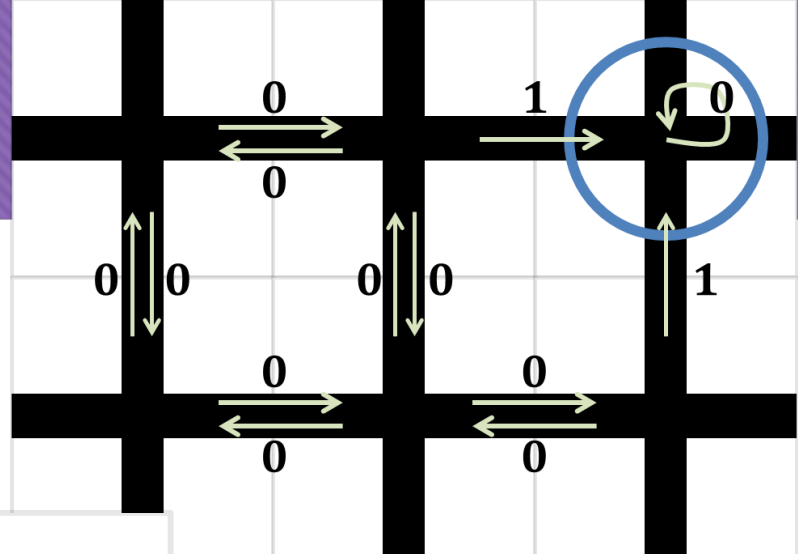
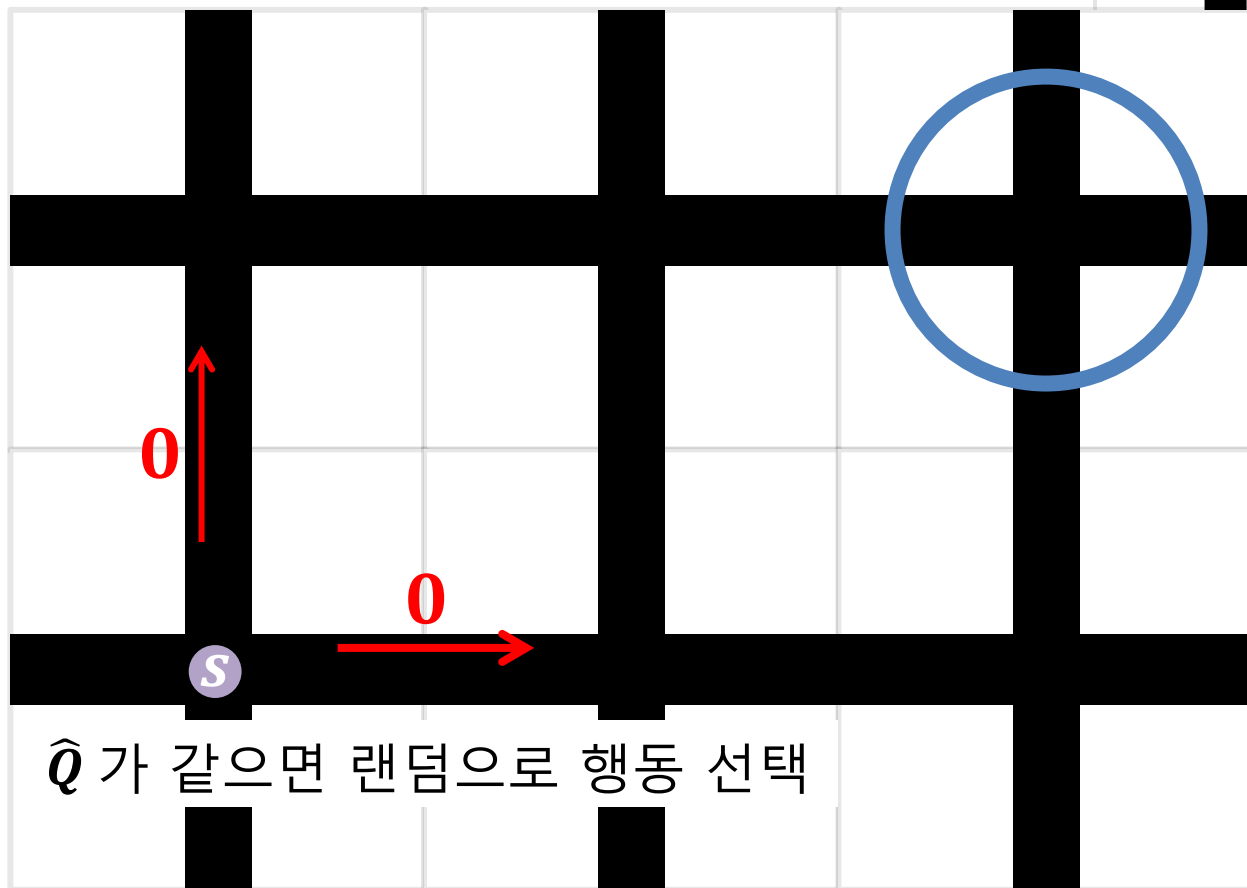
\hat{Q} 초기 값 : 모두 0



Q-러닝

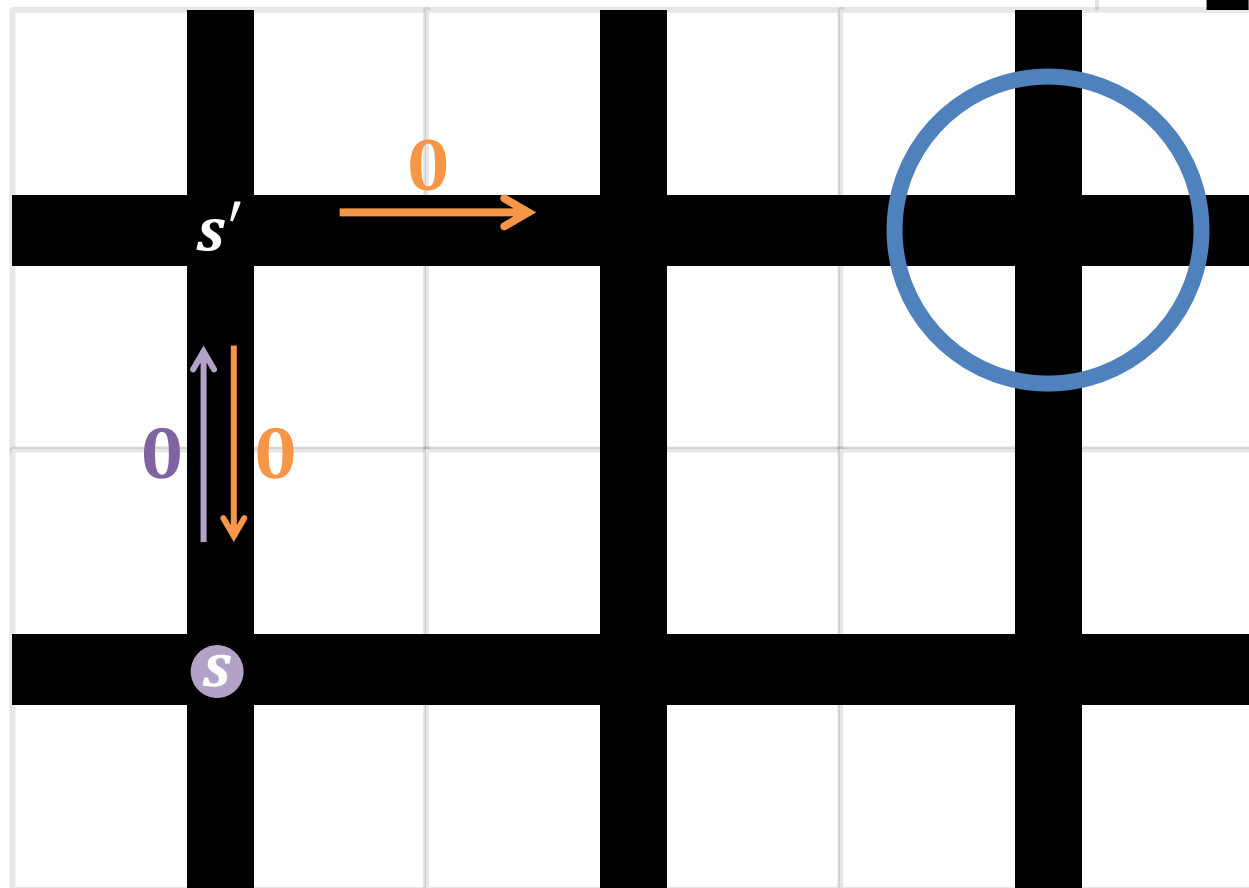
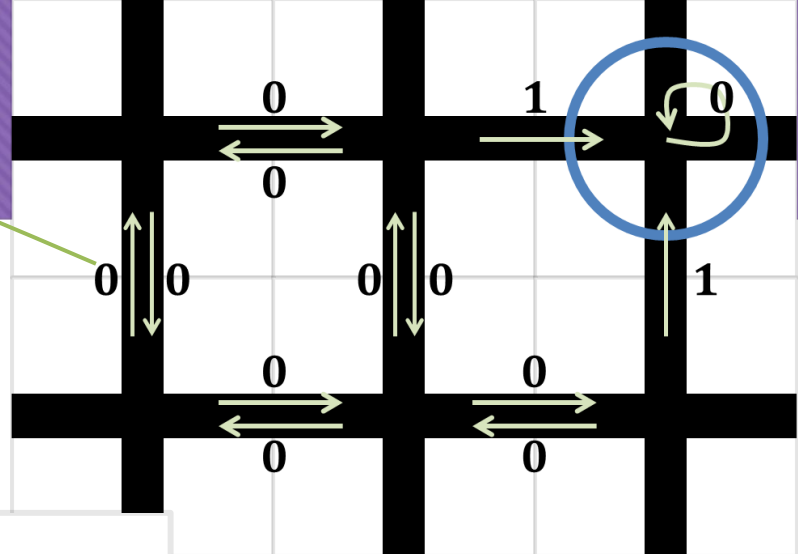
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

출발 위치: 왼쪽 아래



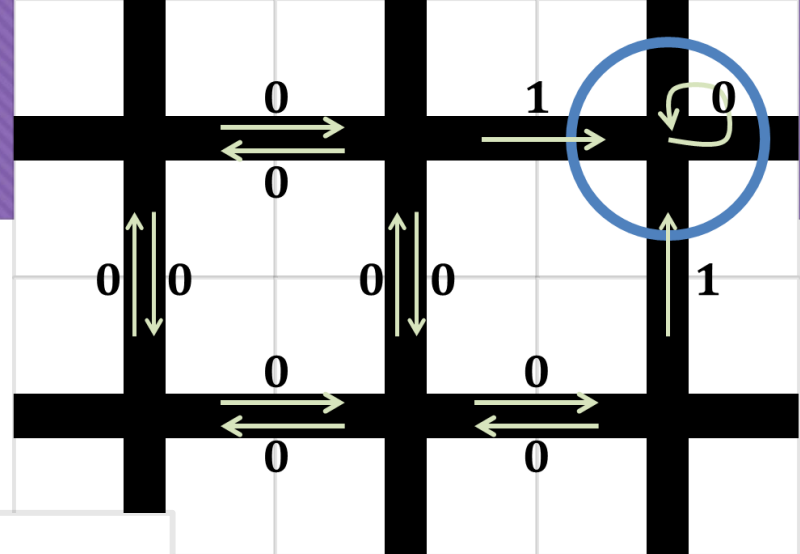
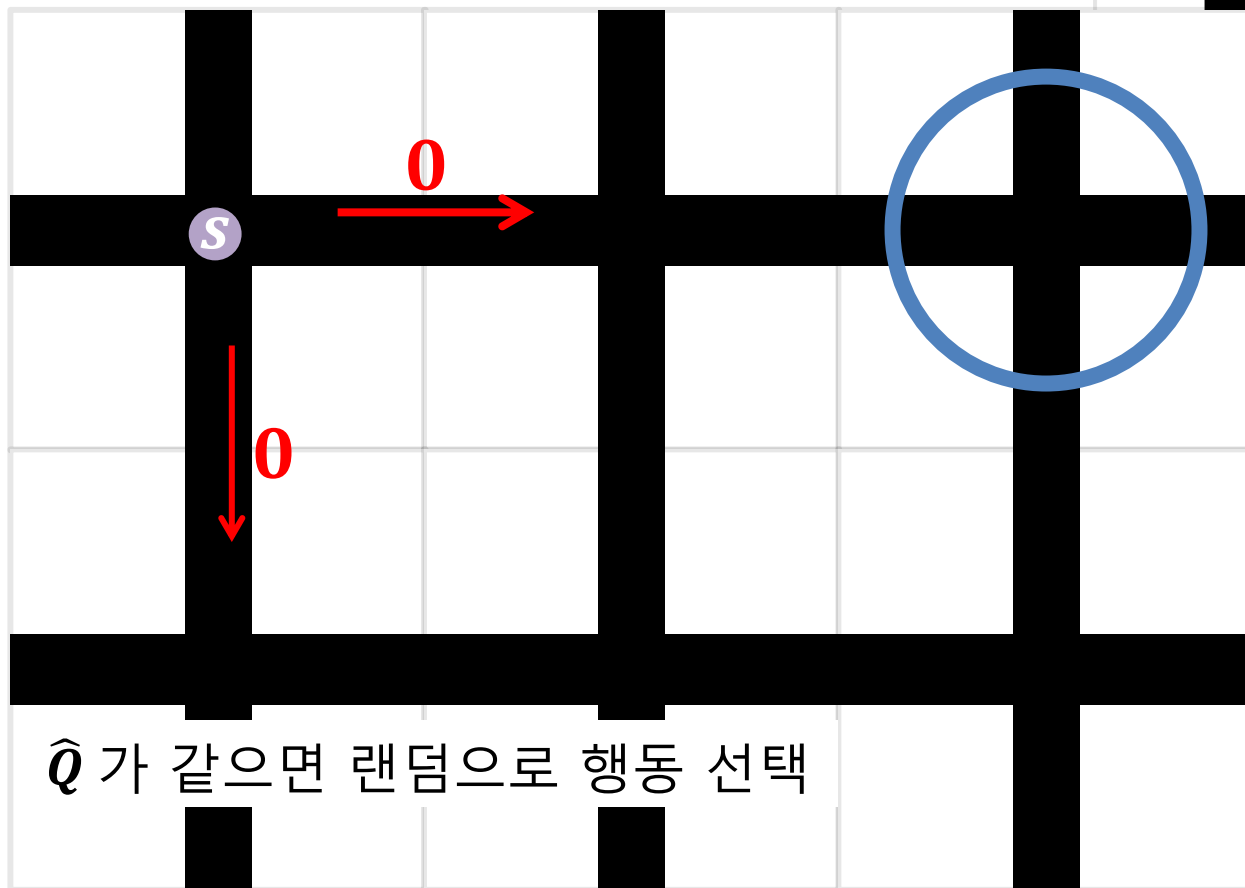
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

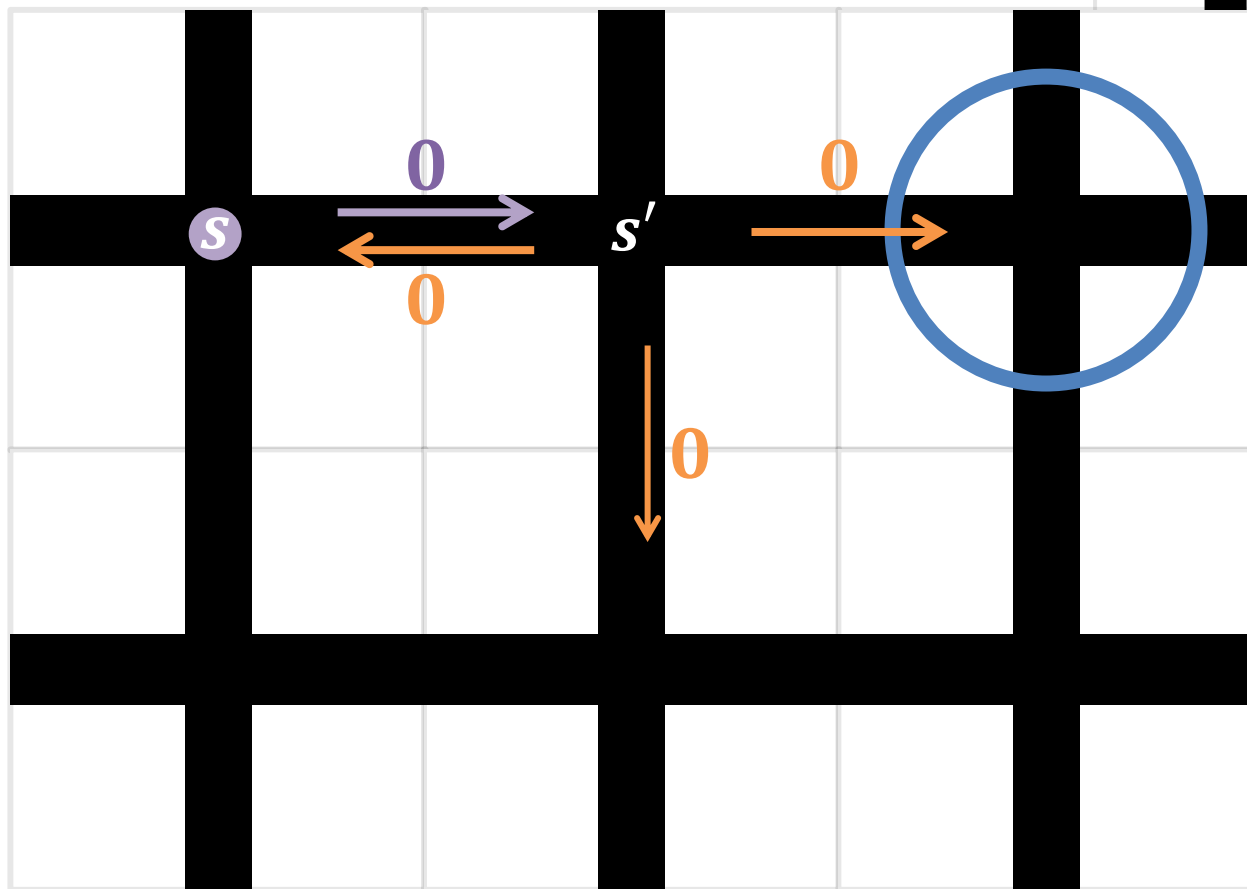
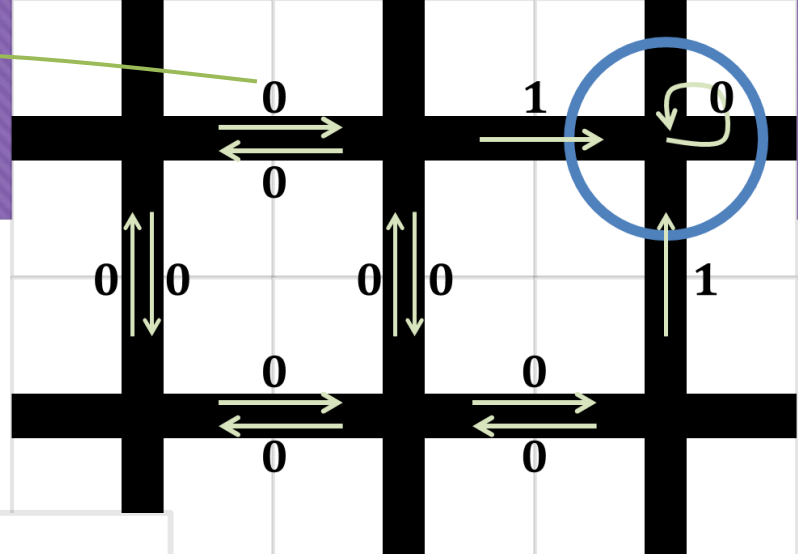
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

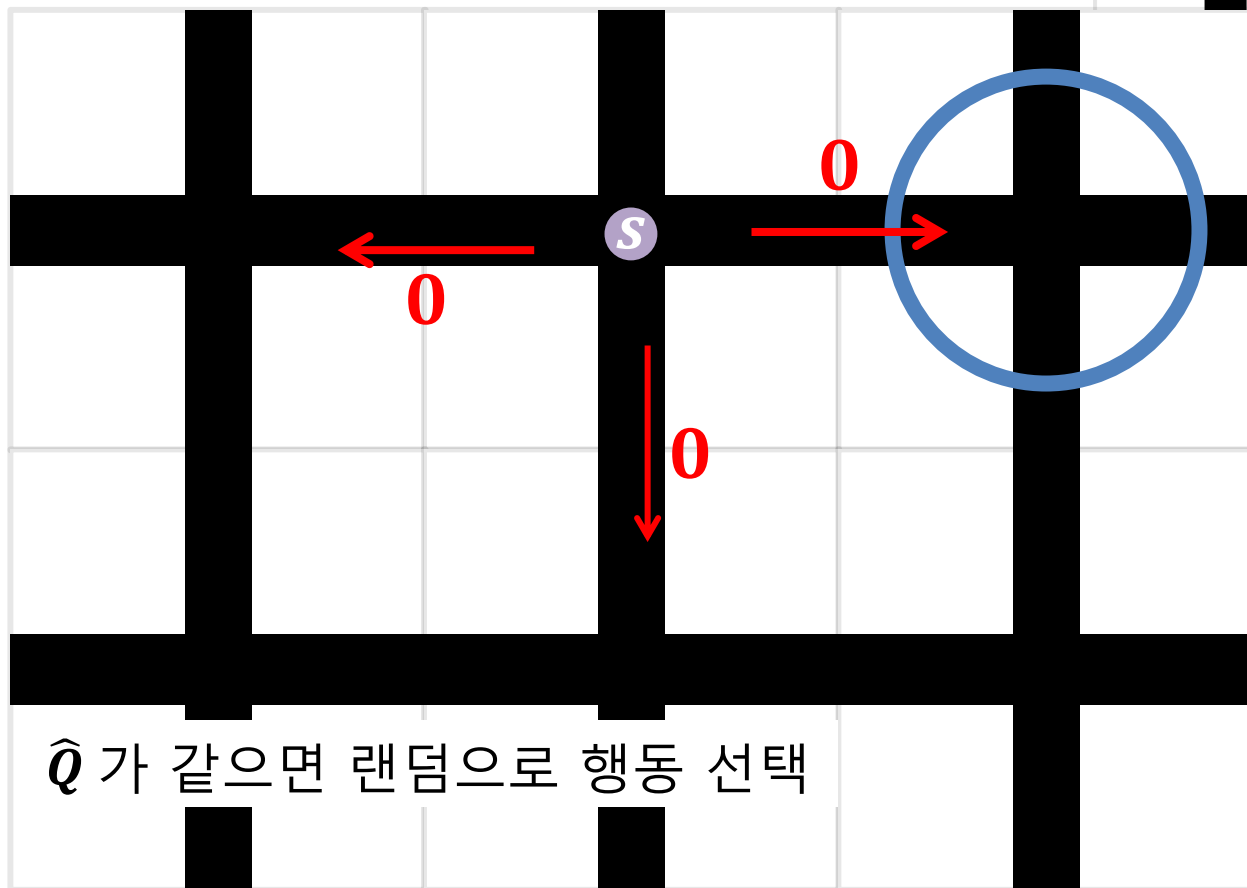
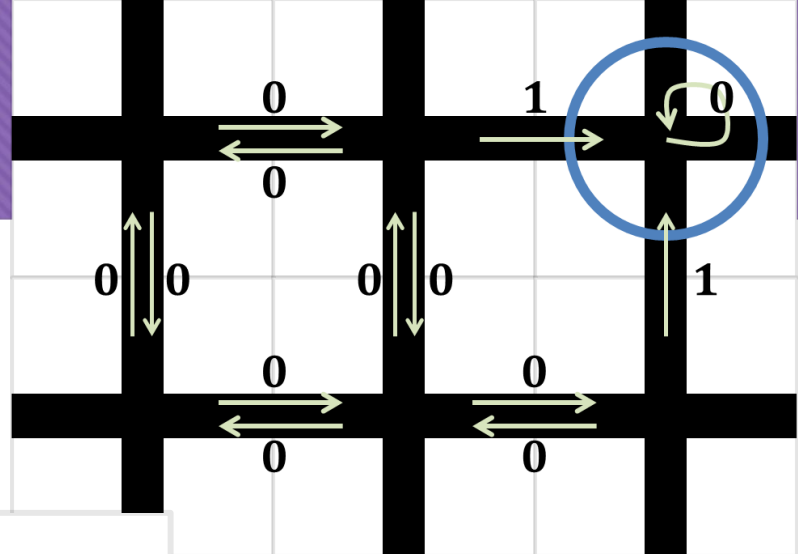
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

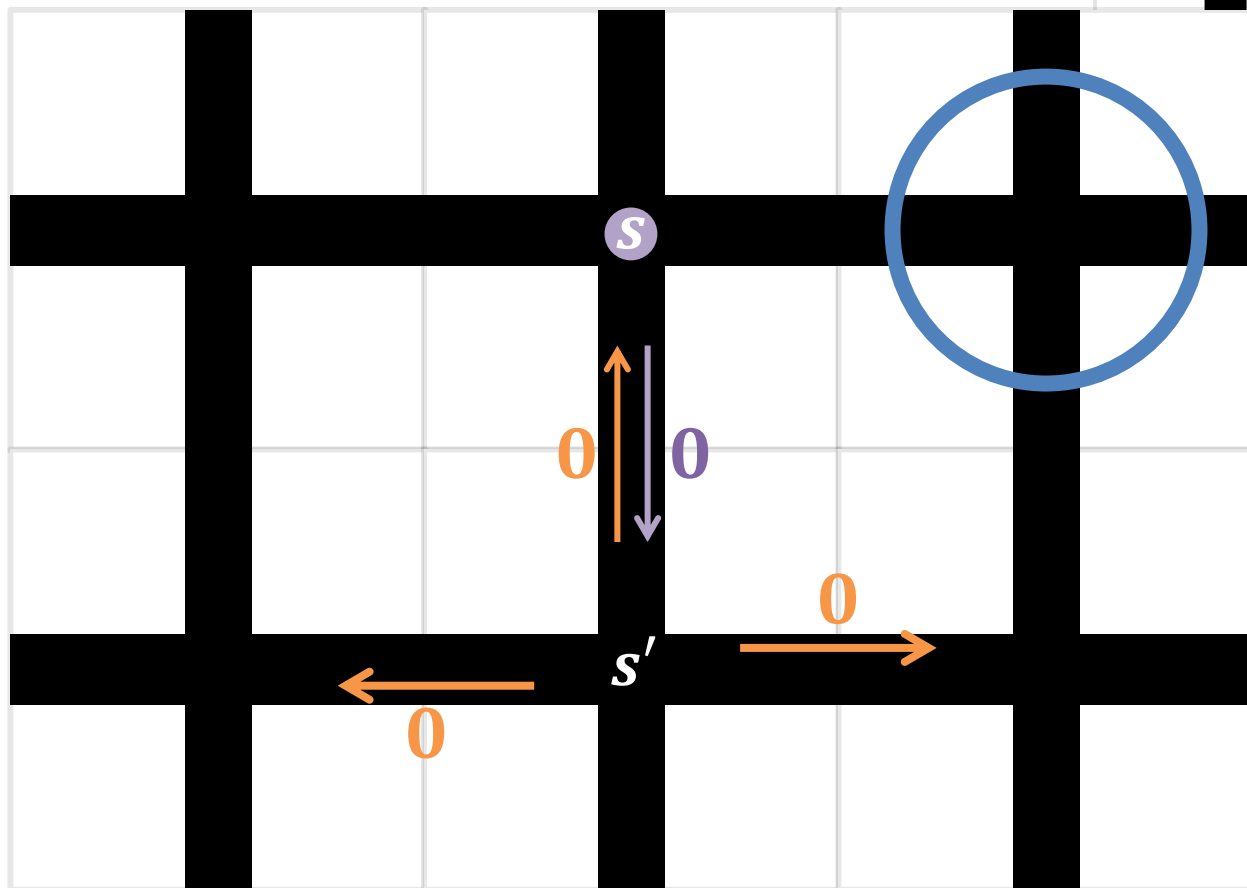
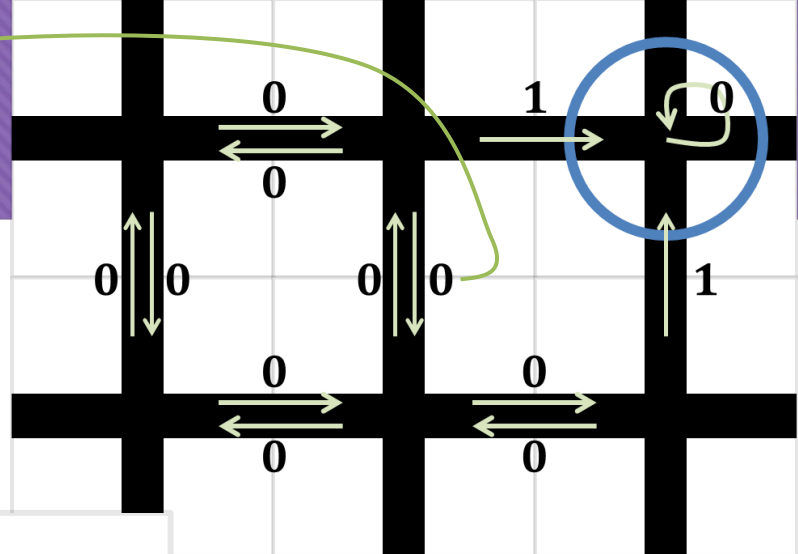
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

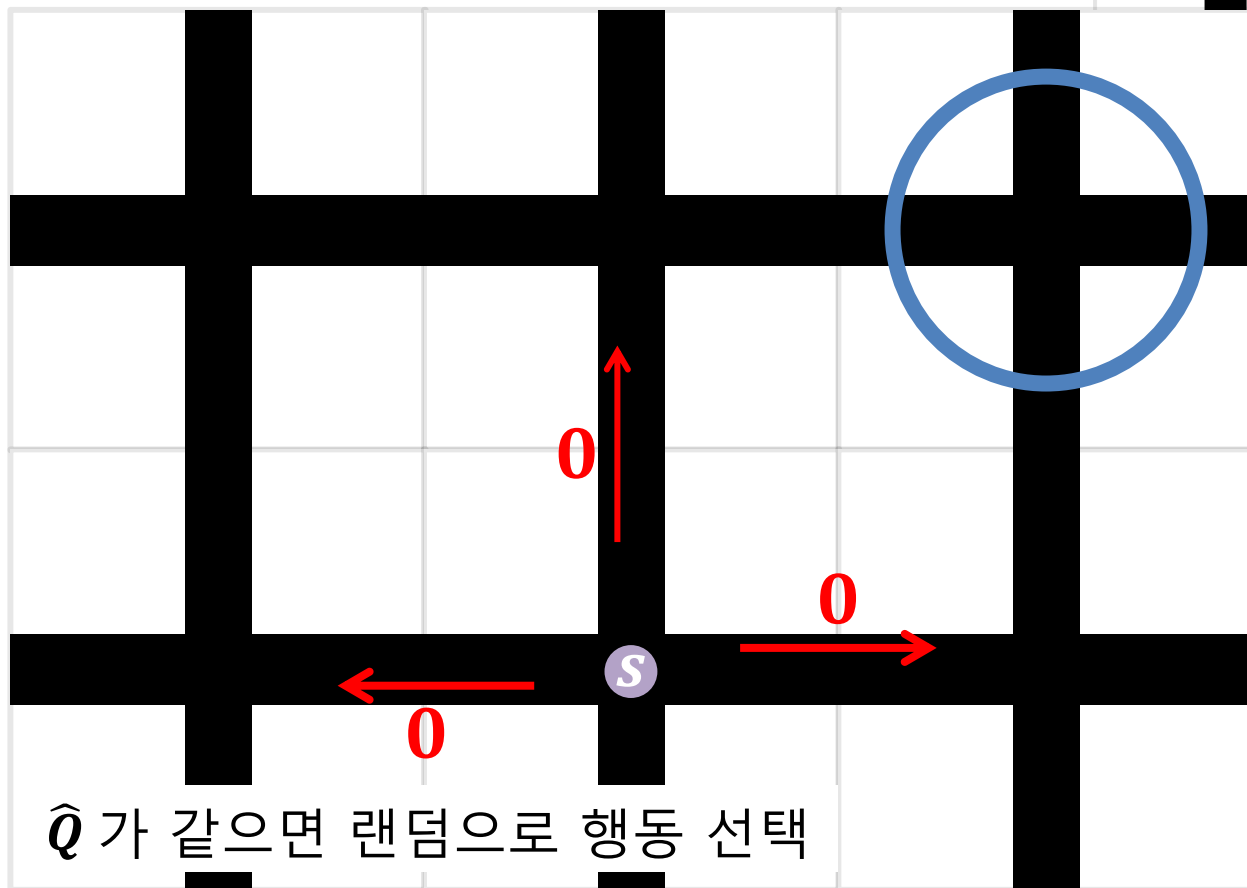
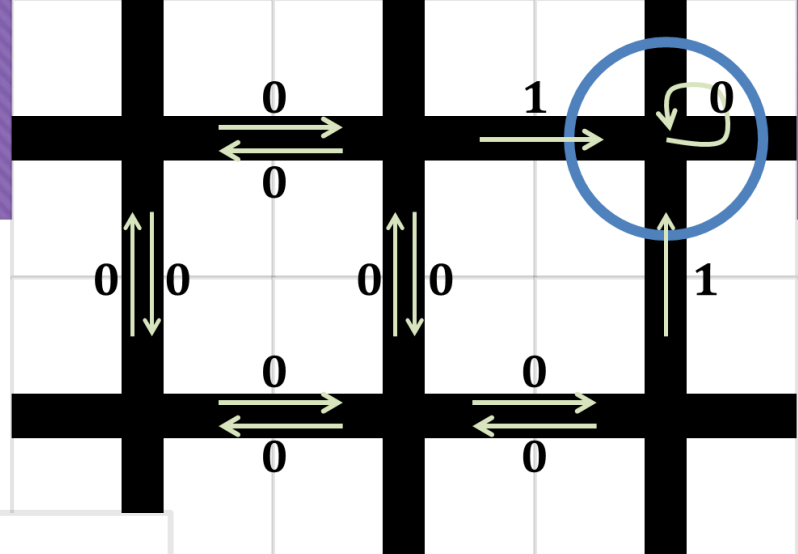
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

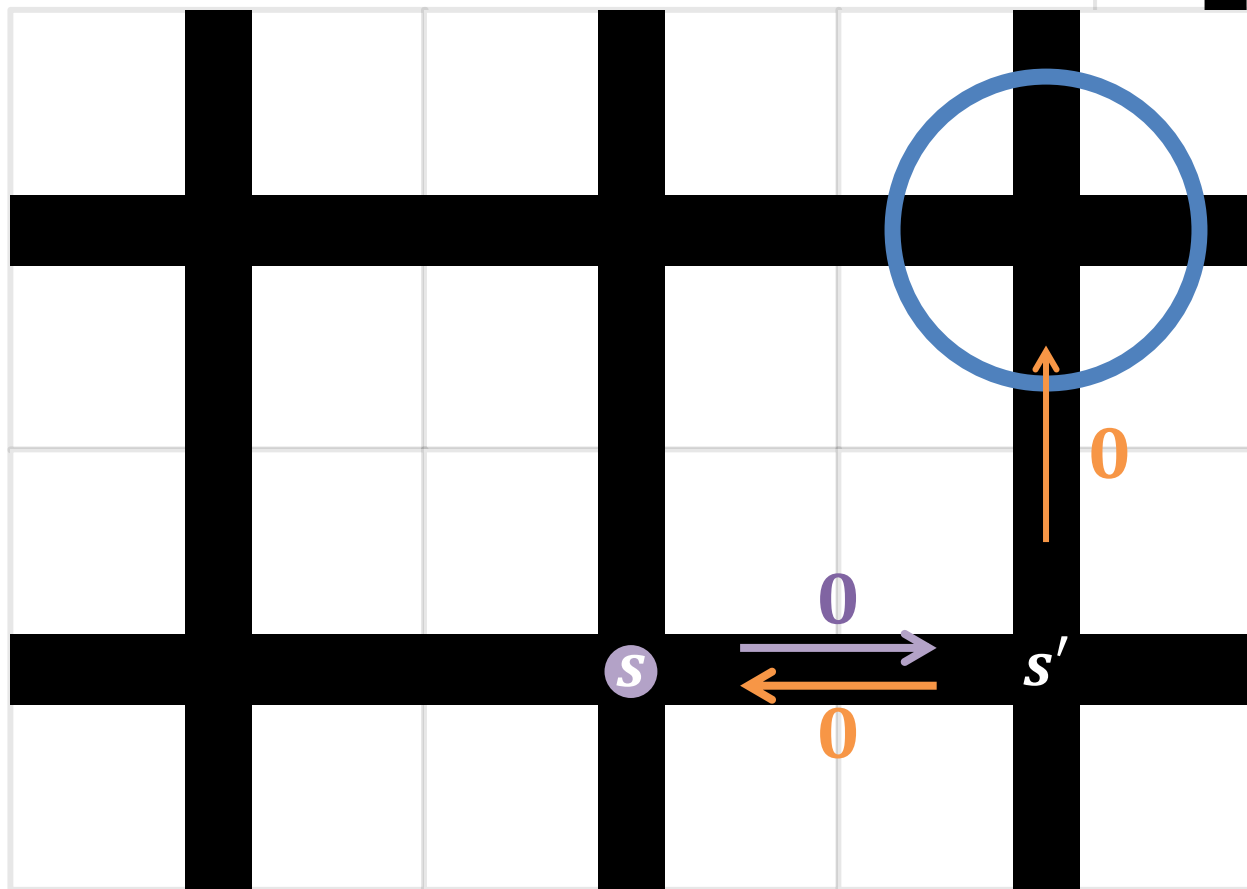
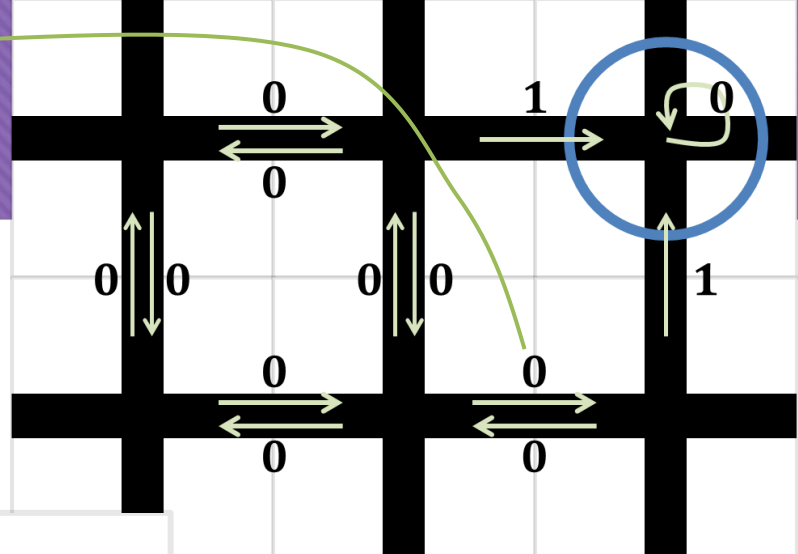
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

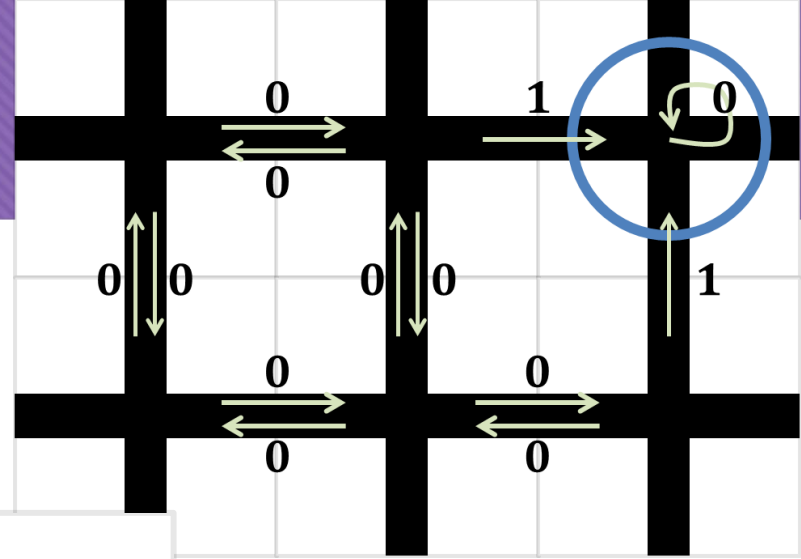
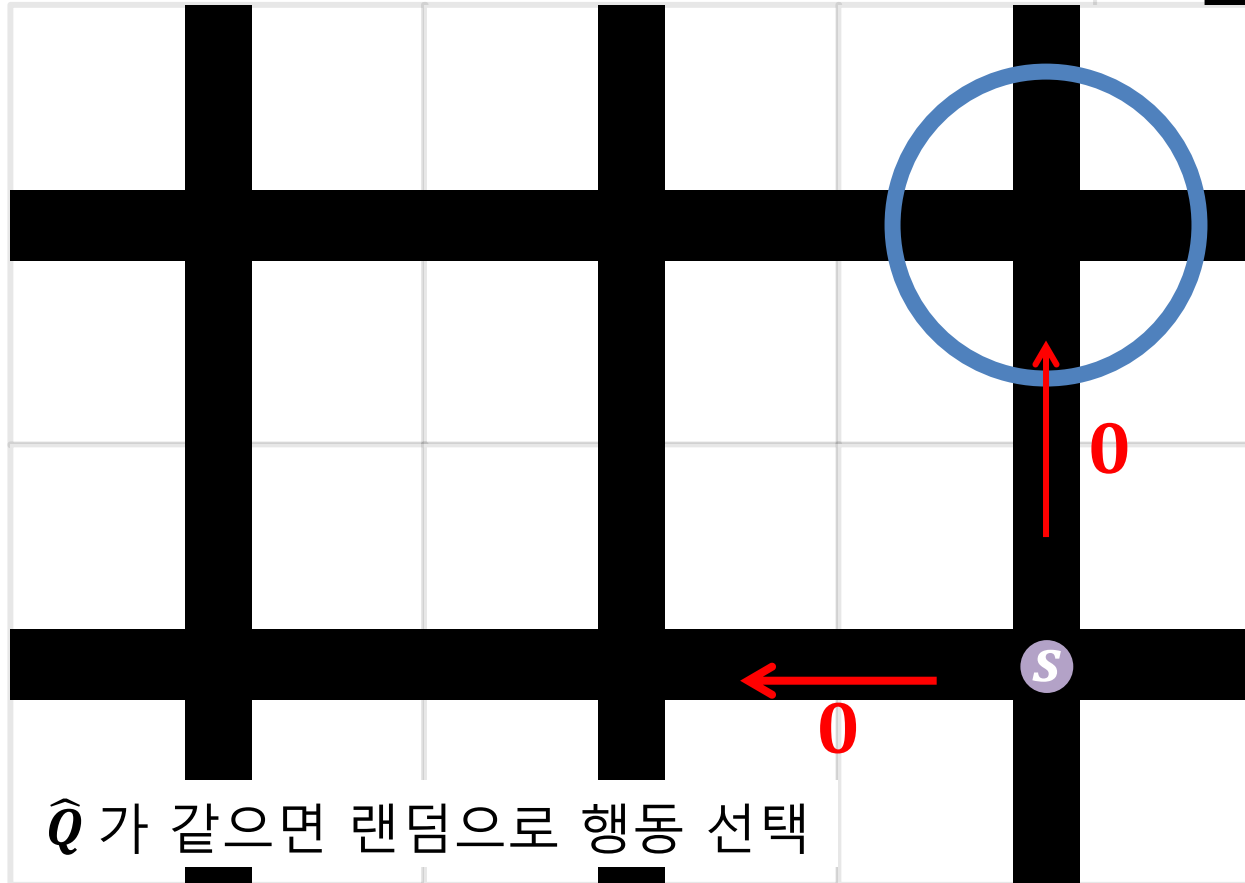
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

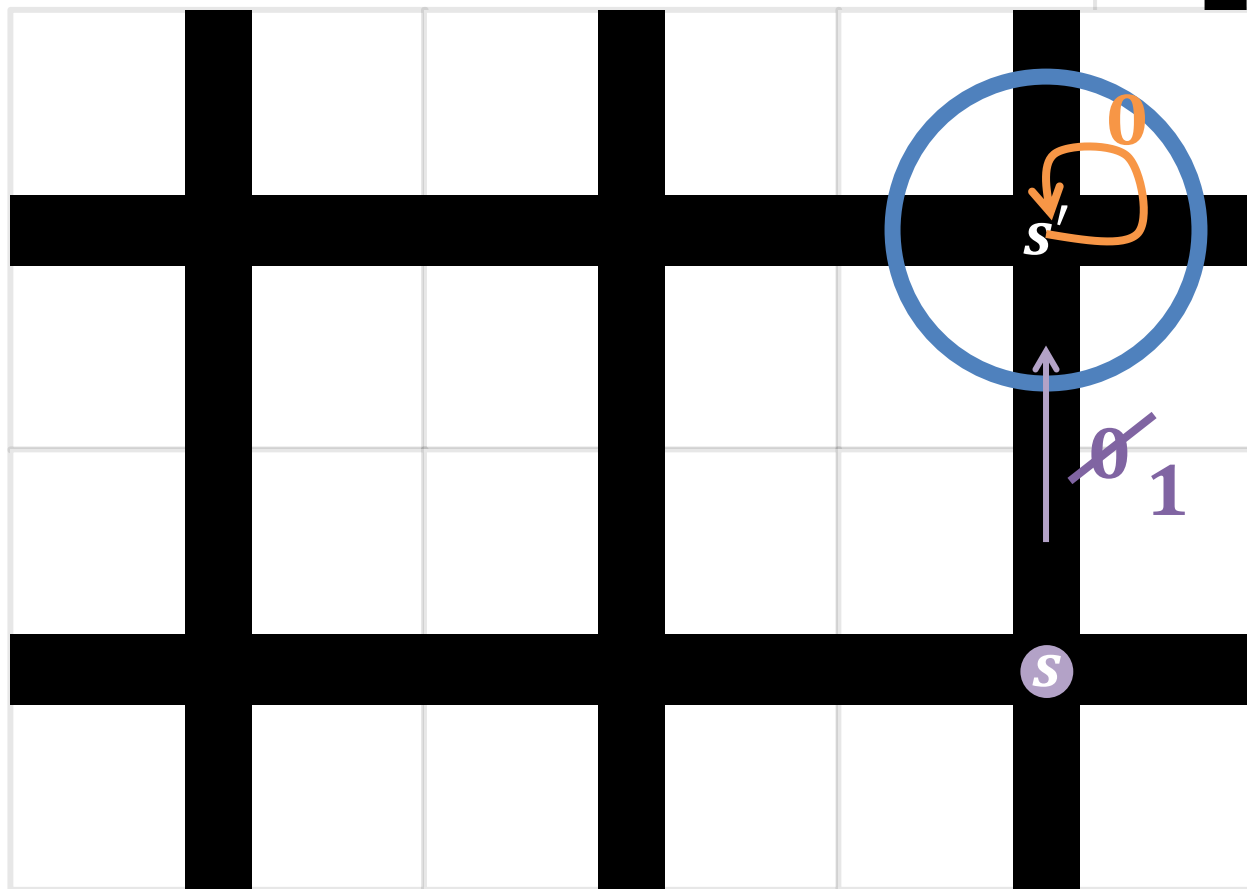
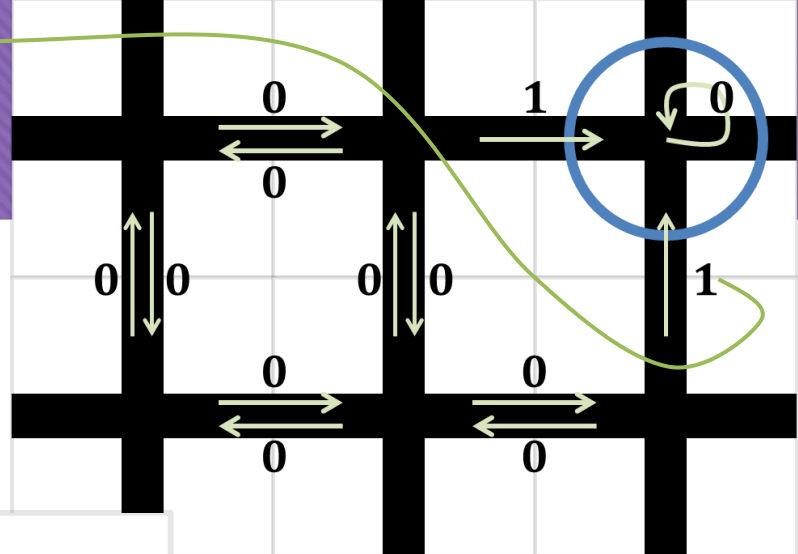
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

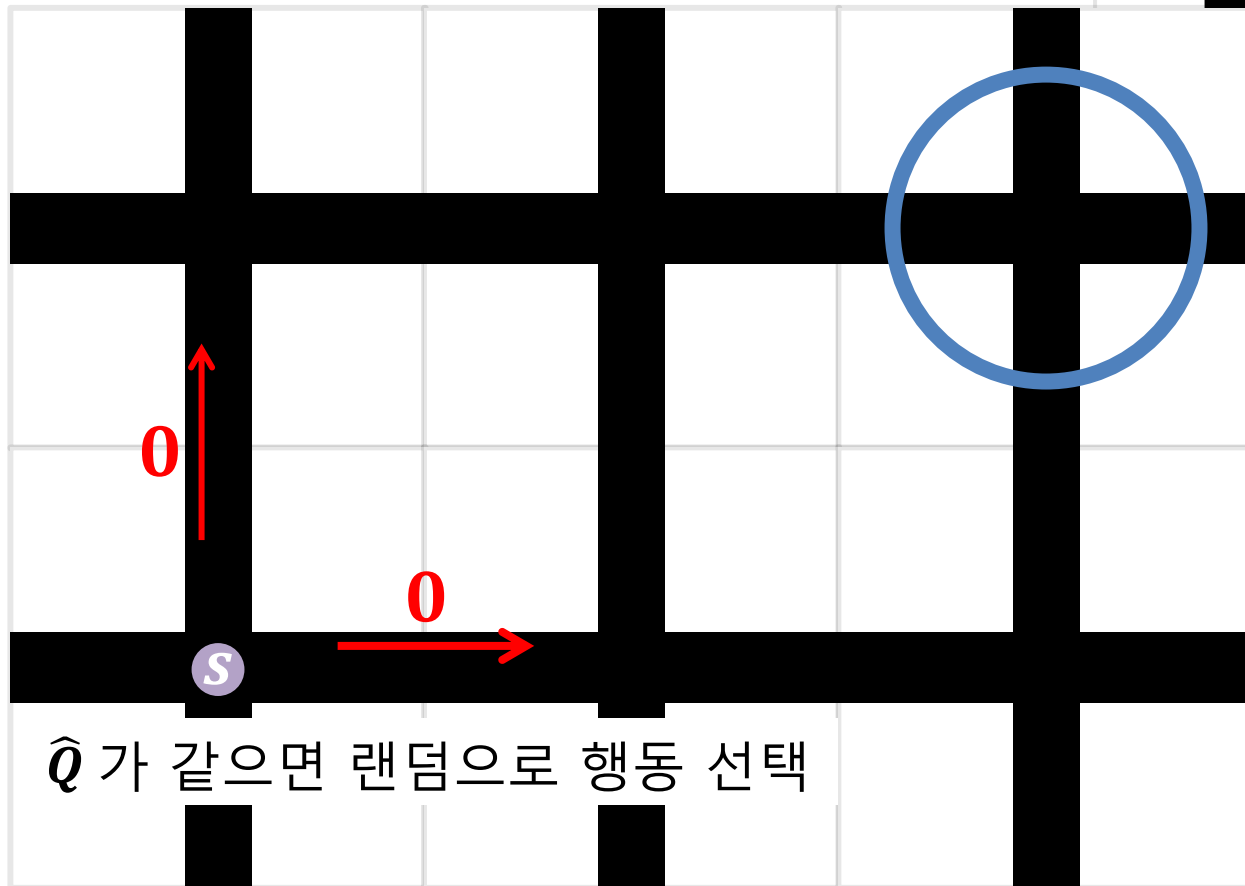
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

1 0

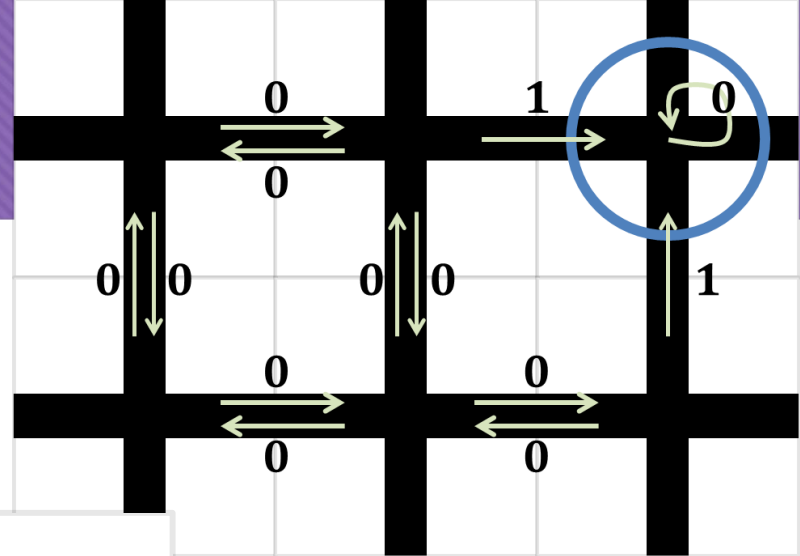


Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

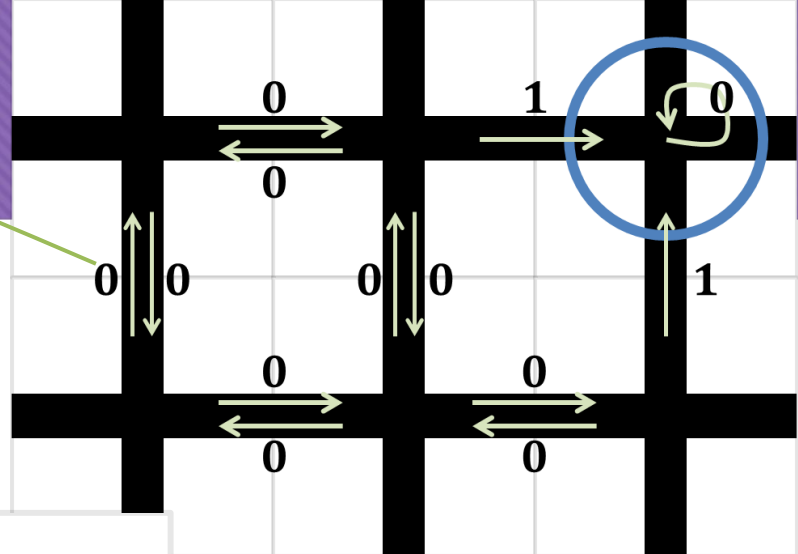
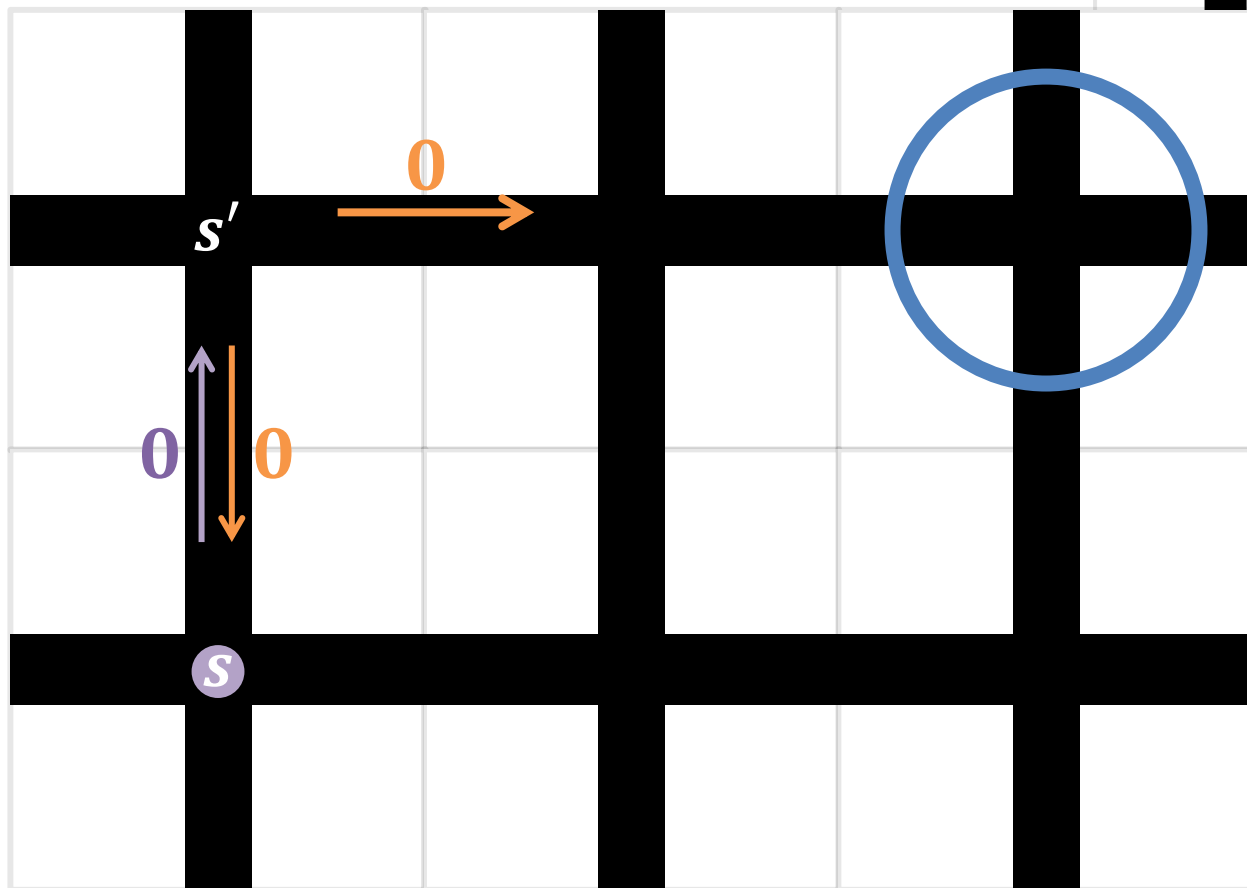


\hat{Q} 가 같으면 랜덤으로 행동 선택



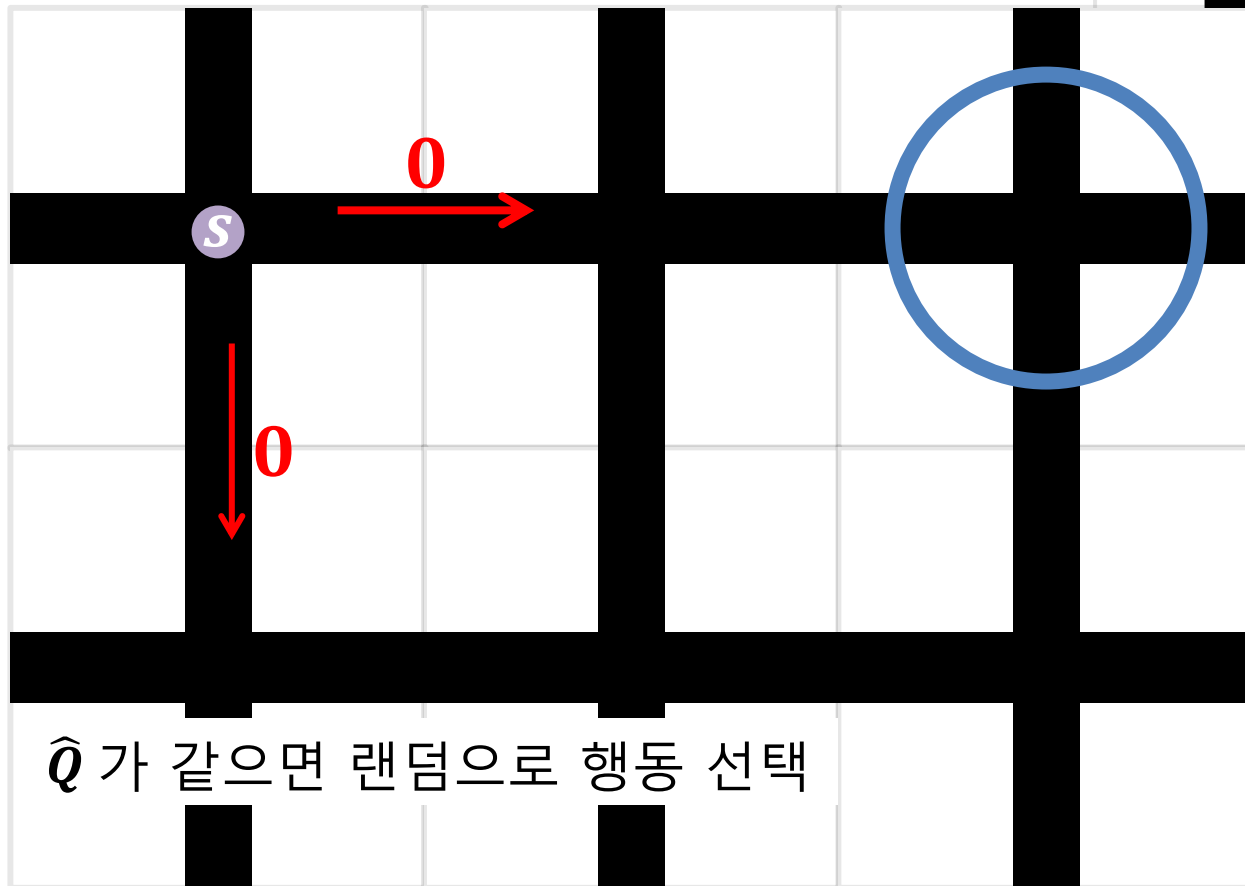
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

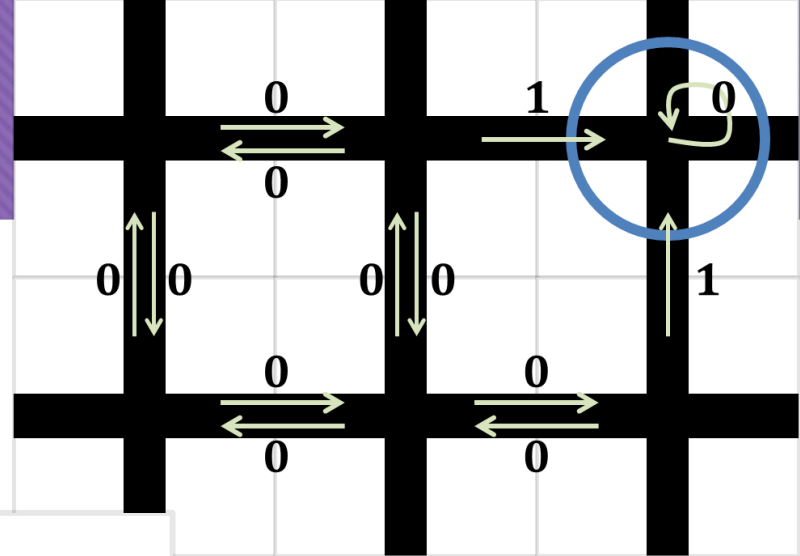


Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



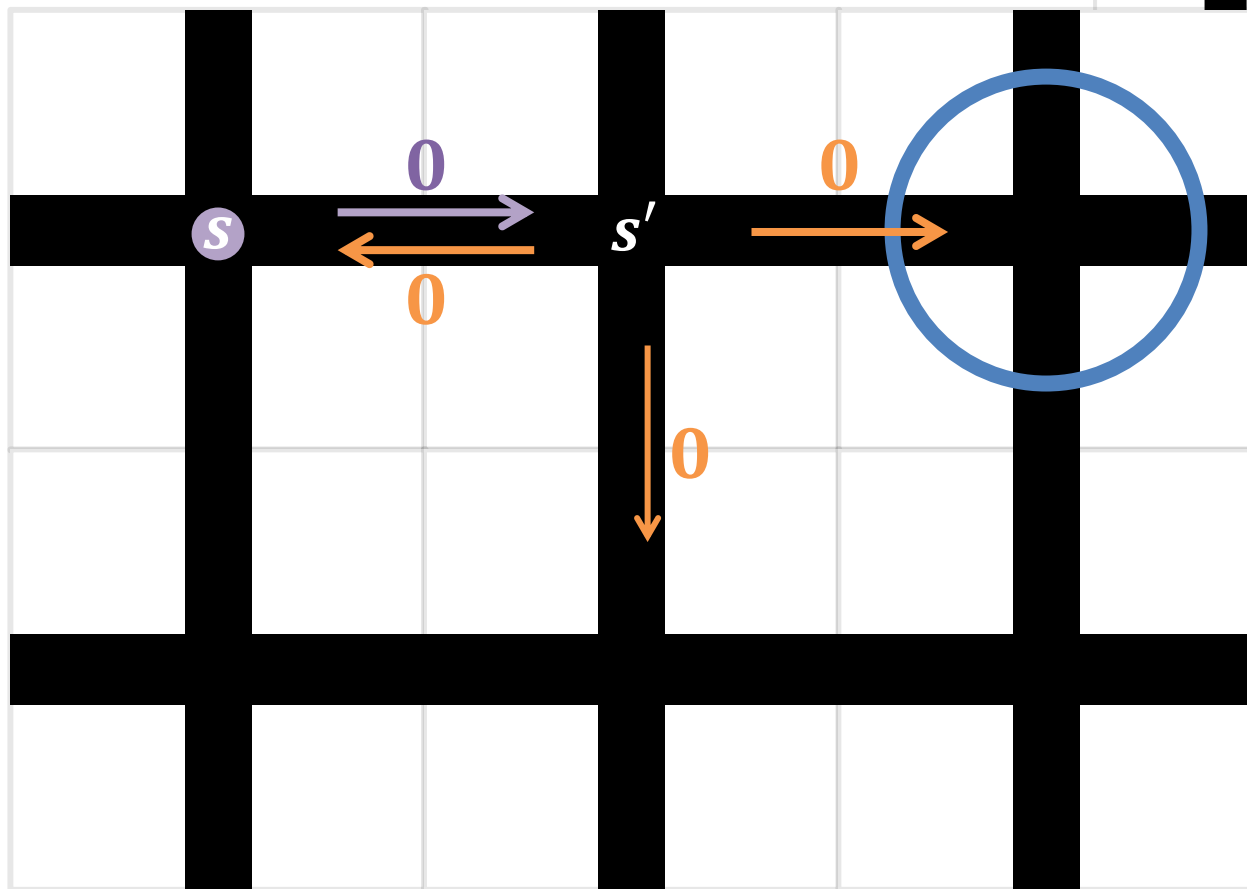
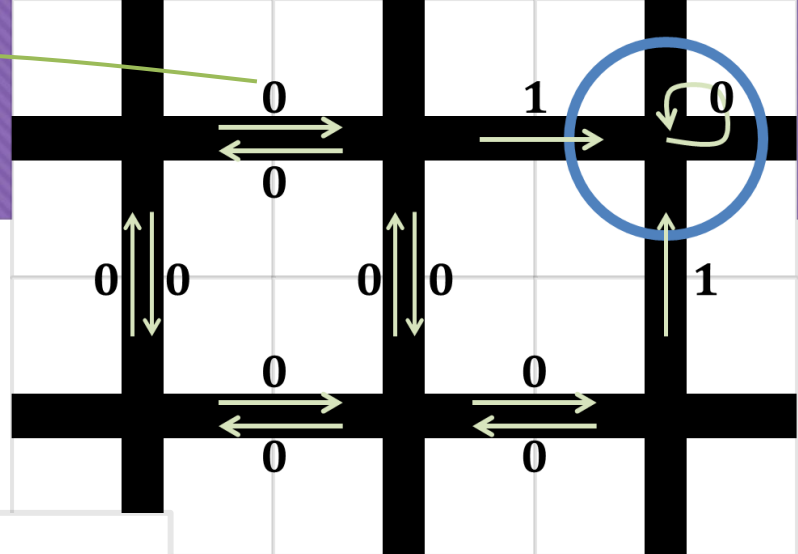
\hat{Q} 가 같으면 랜덤으로 행동 선택



Q-러닝

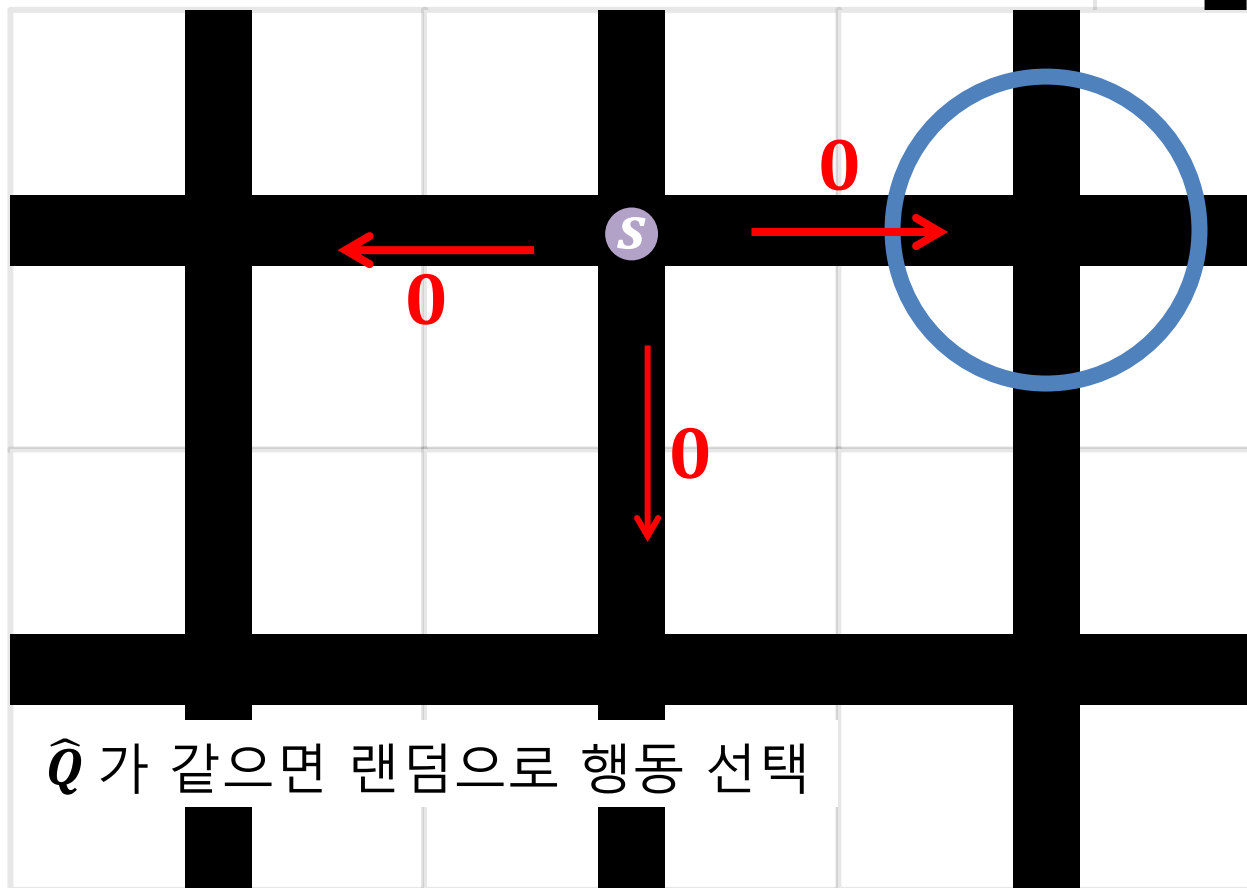
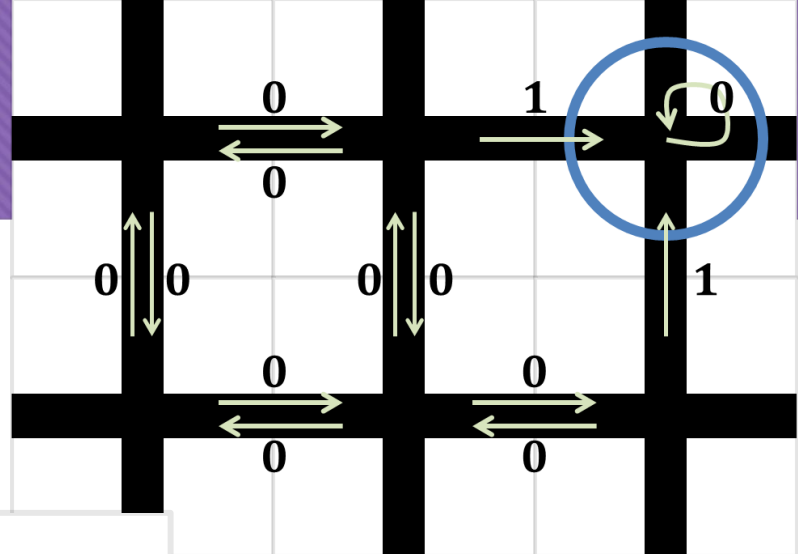
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

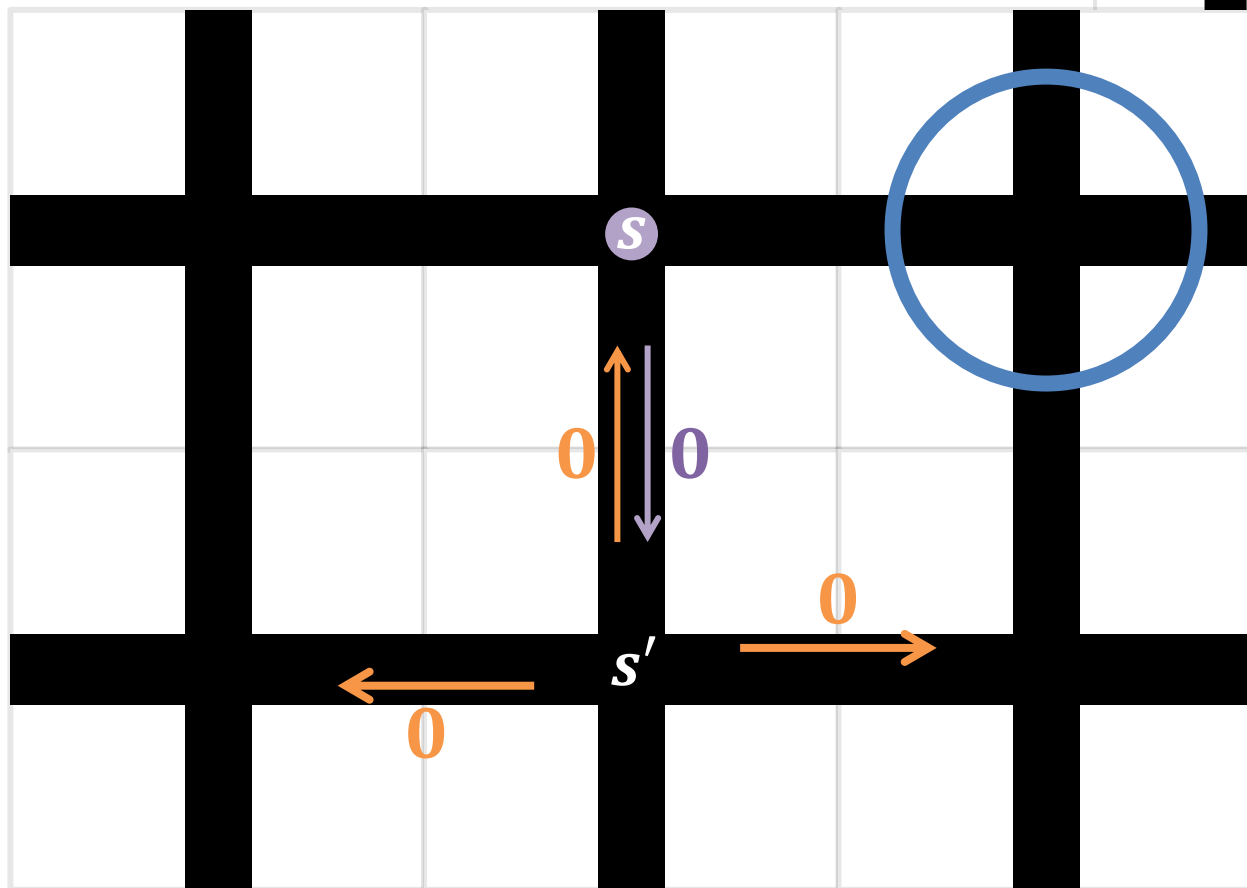
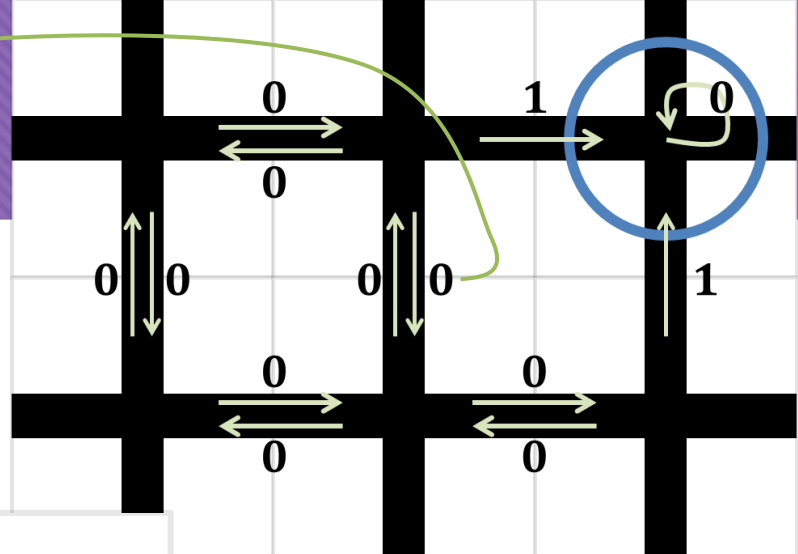


\hat{Q} 가 같으면 랜덤으로 행동 선택

Q-러닝

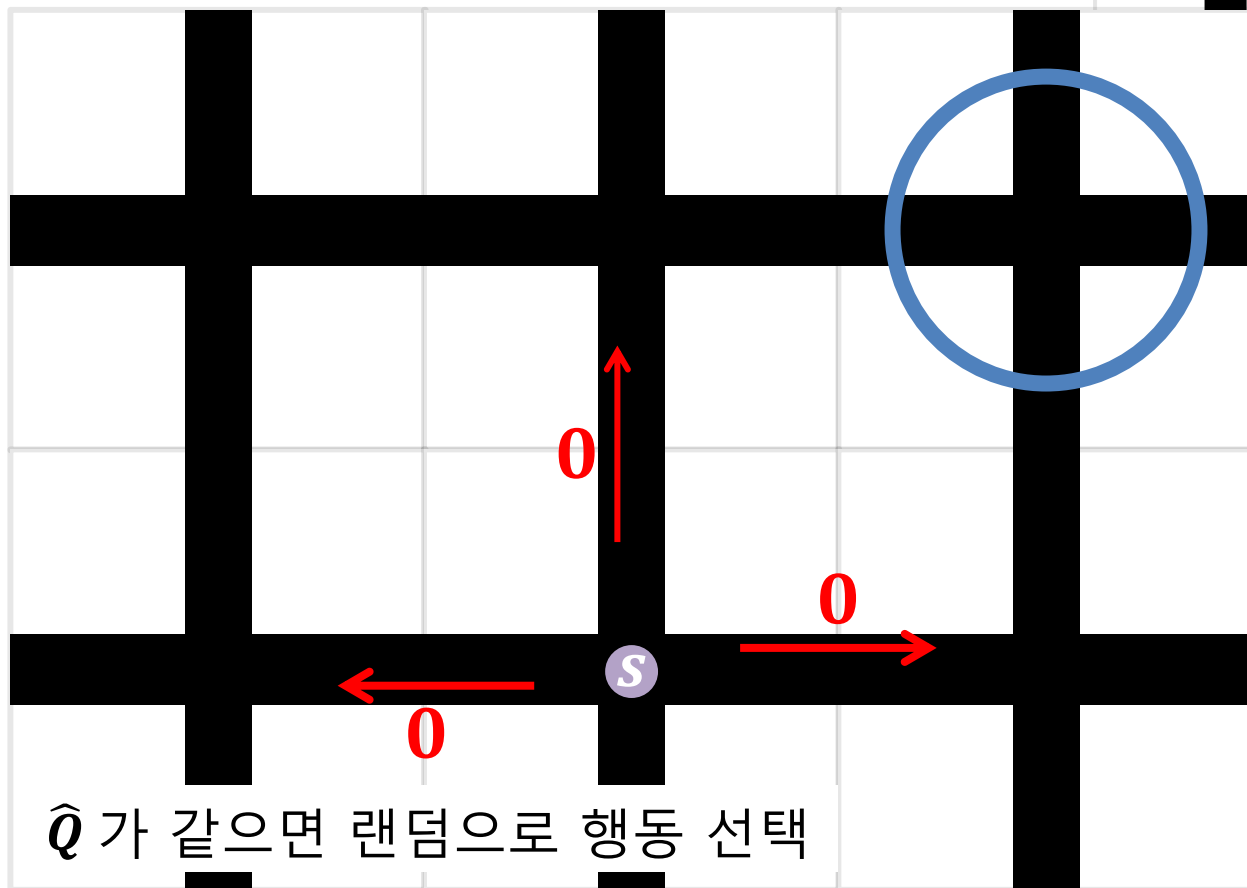
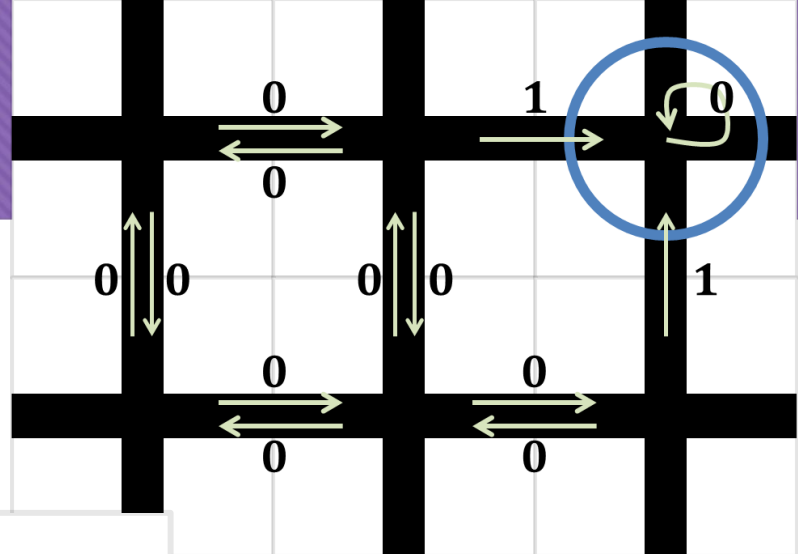
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

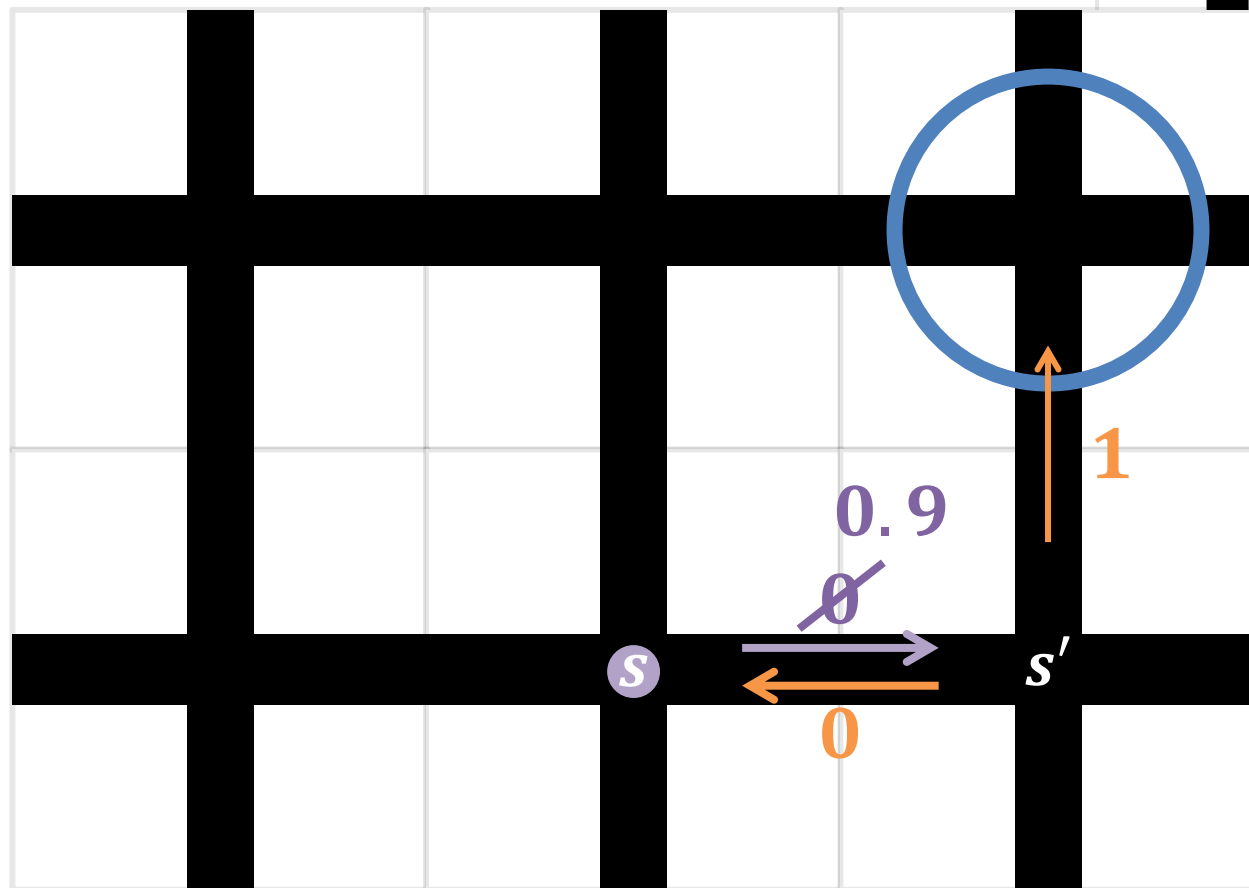
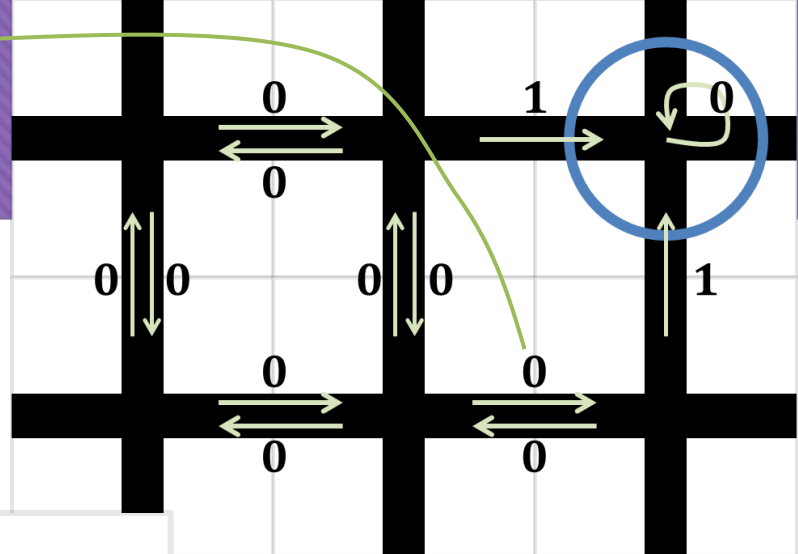
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

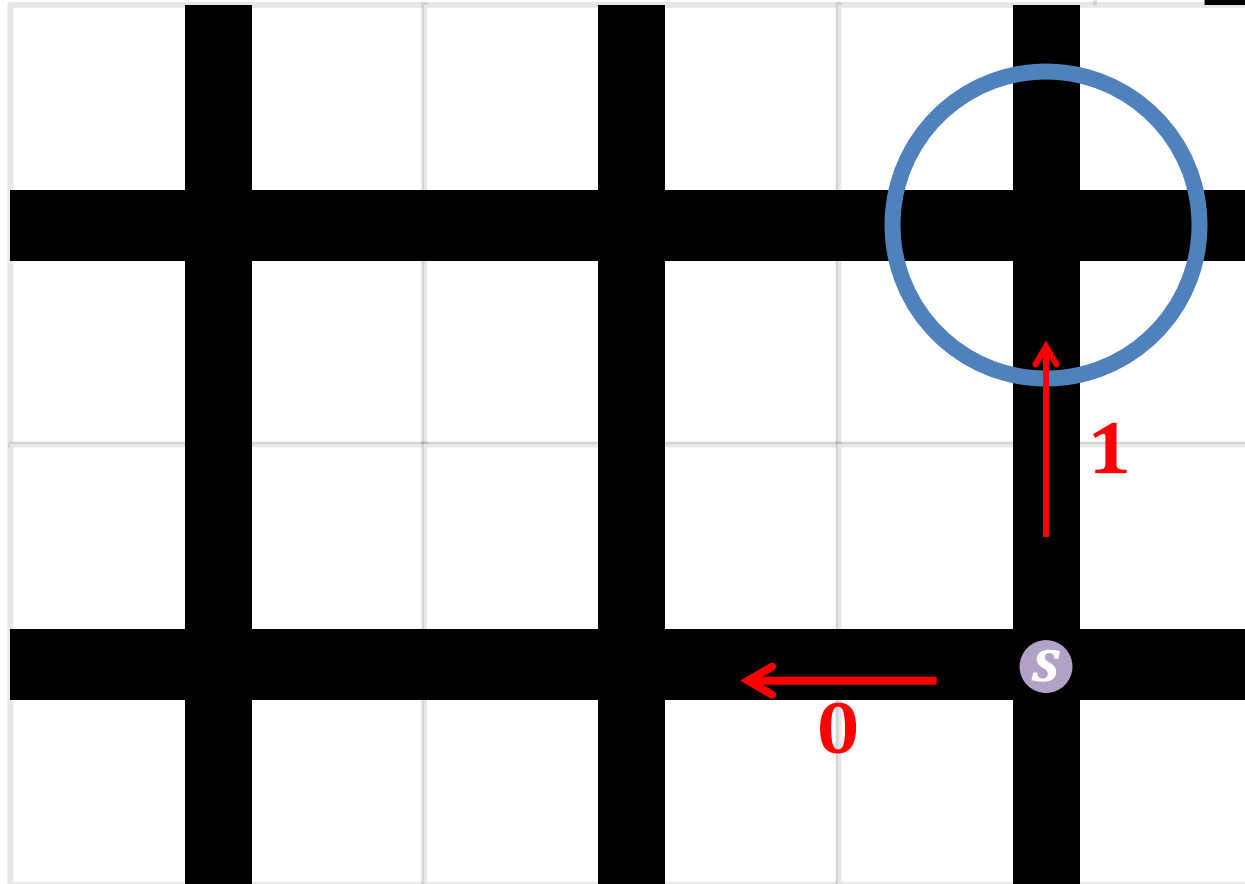
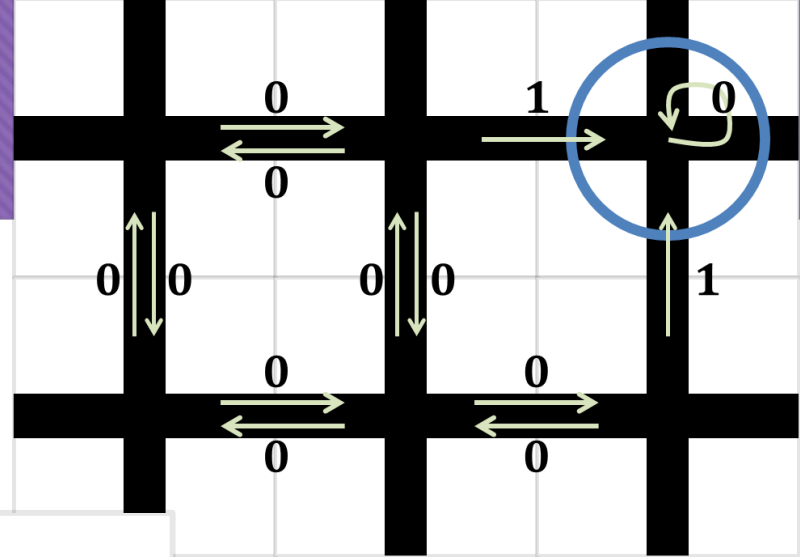
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 1



Q-러닝

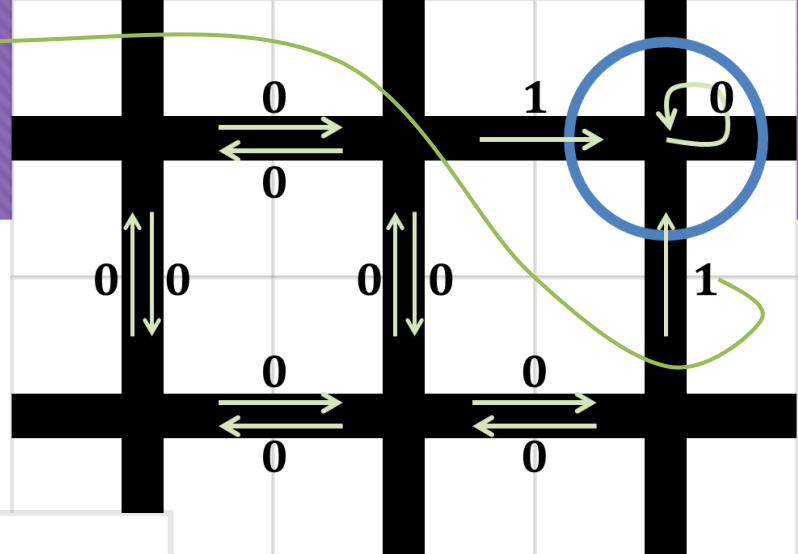
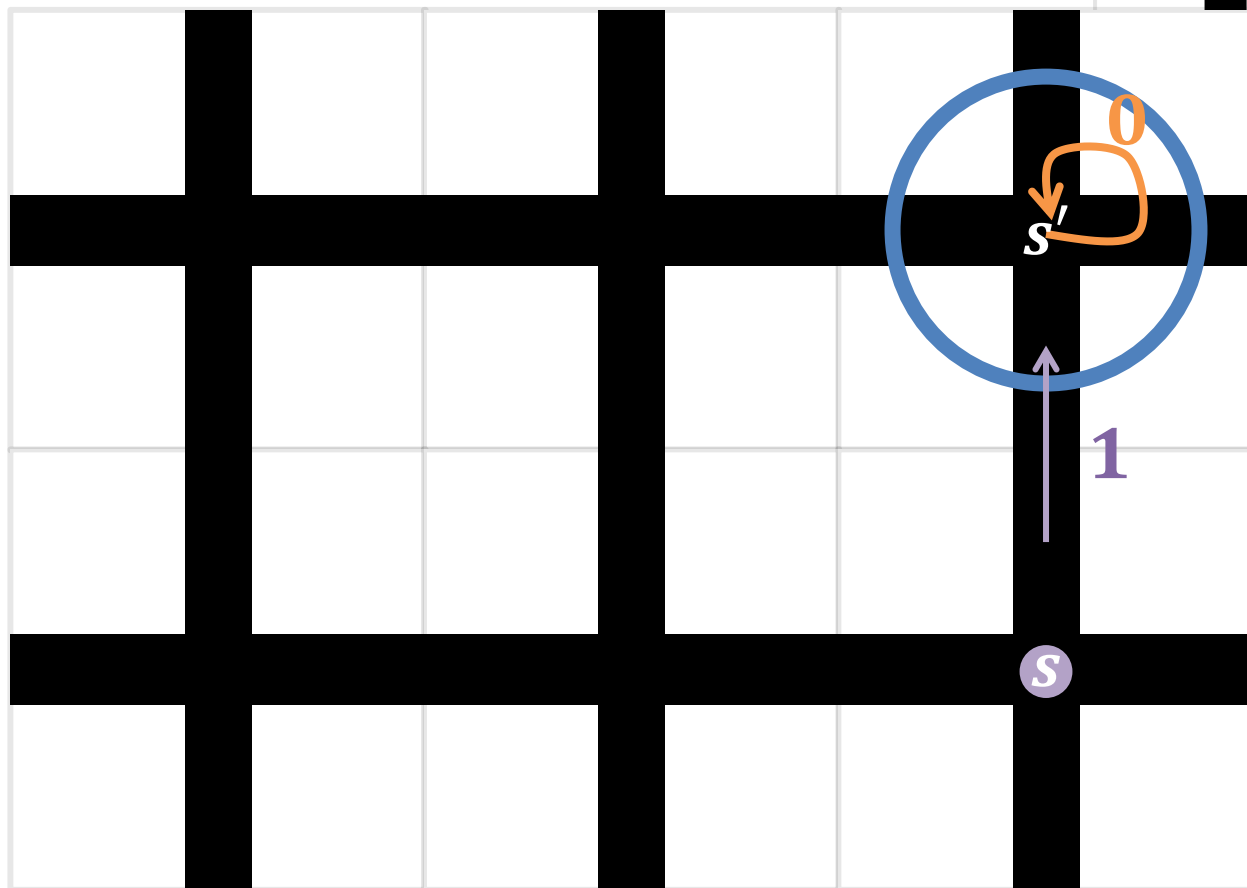
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

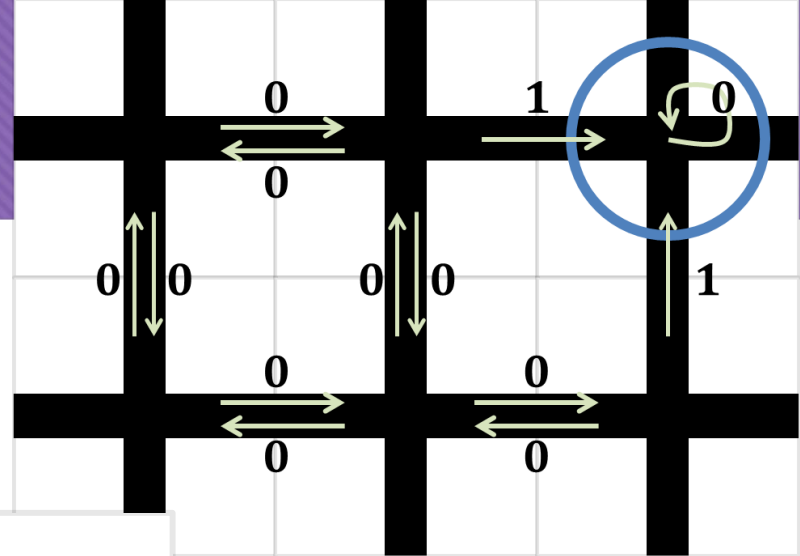
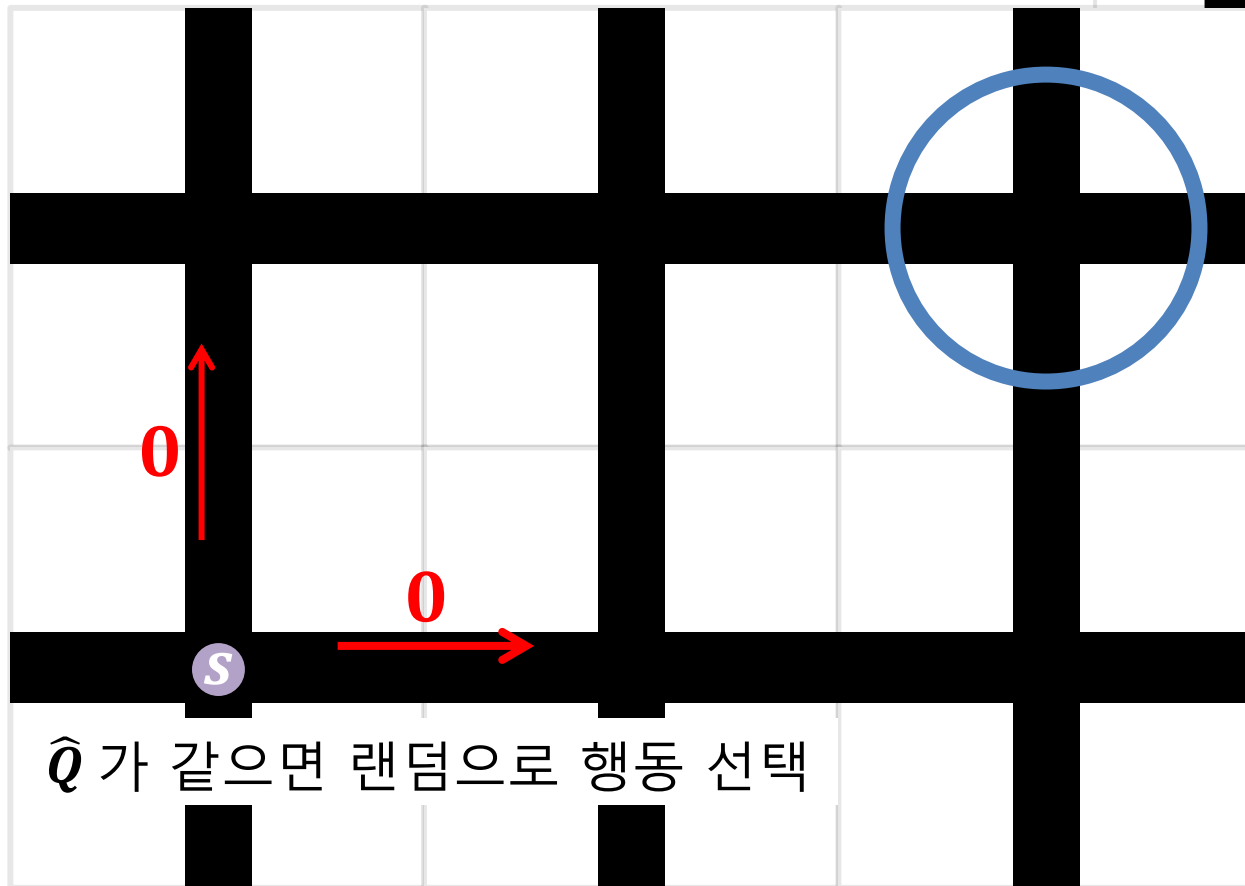
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

1 0



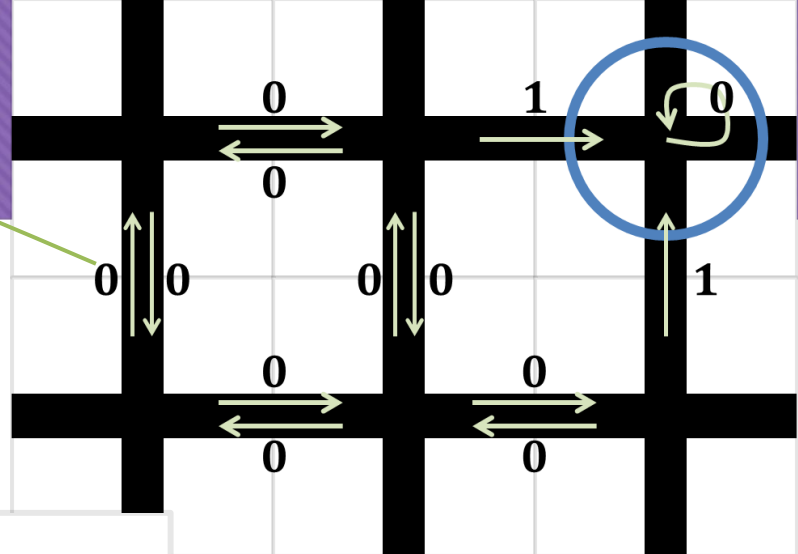
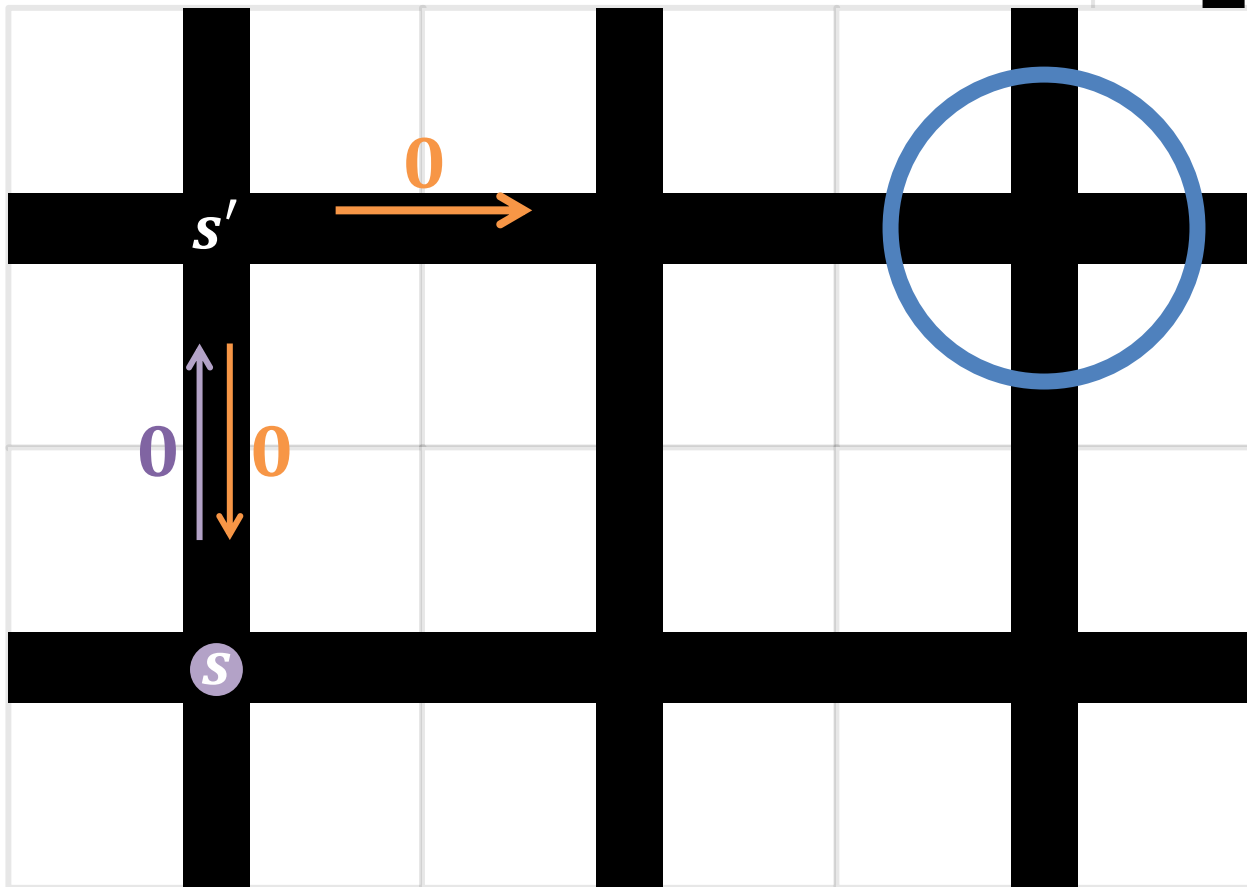
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



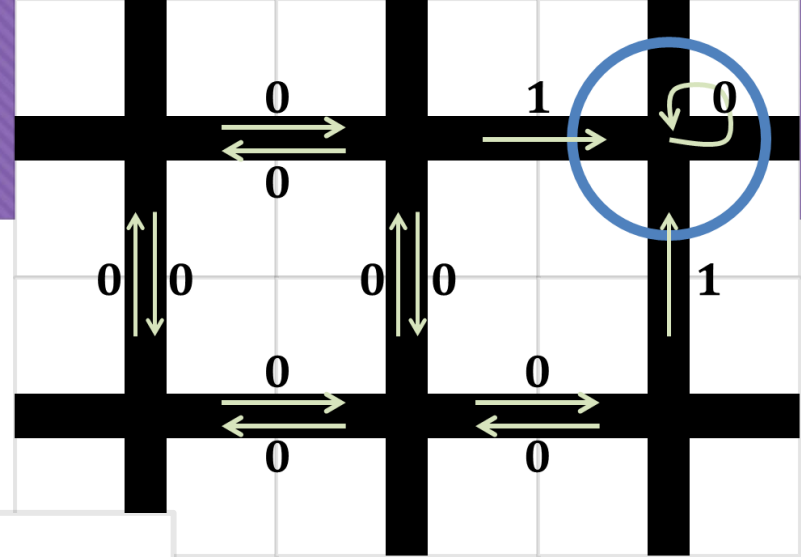
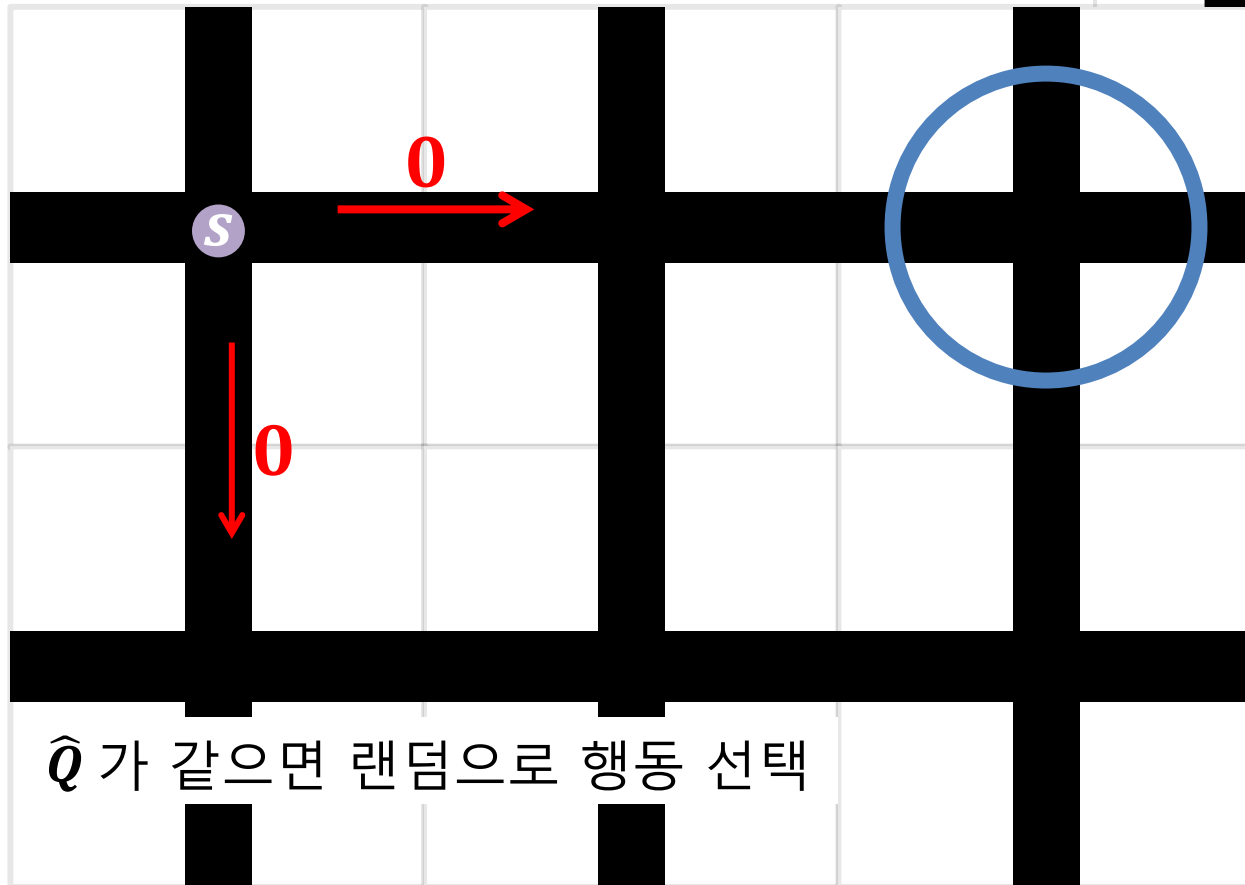
Q-러닝

$$\hat{Q}(s, a) \leftarrow \underset{0}{\underset{0}{\mathbf{r}}} + \mathbf{0.9} \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

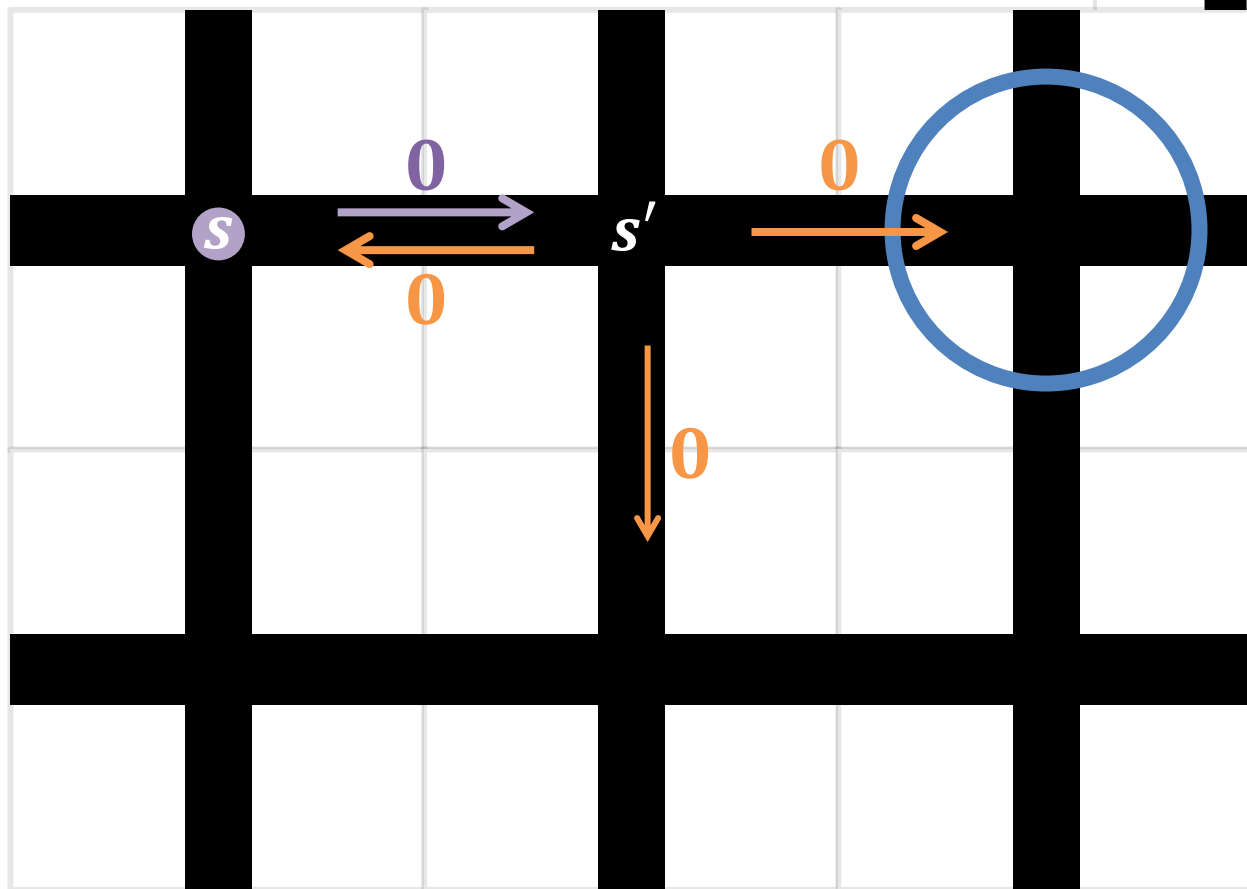
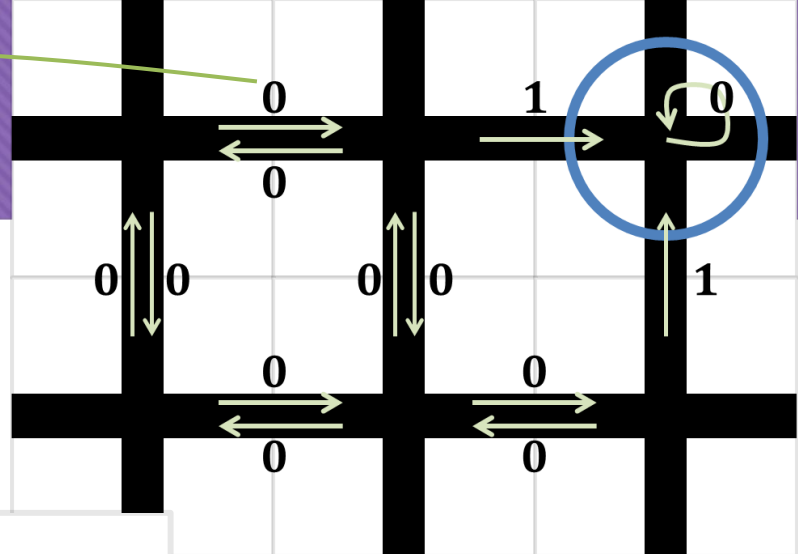
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

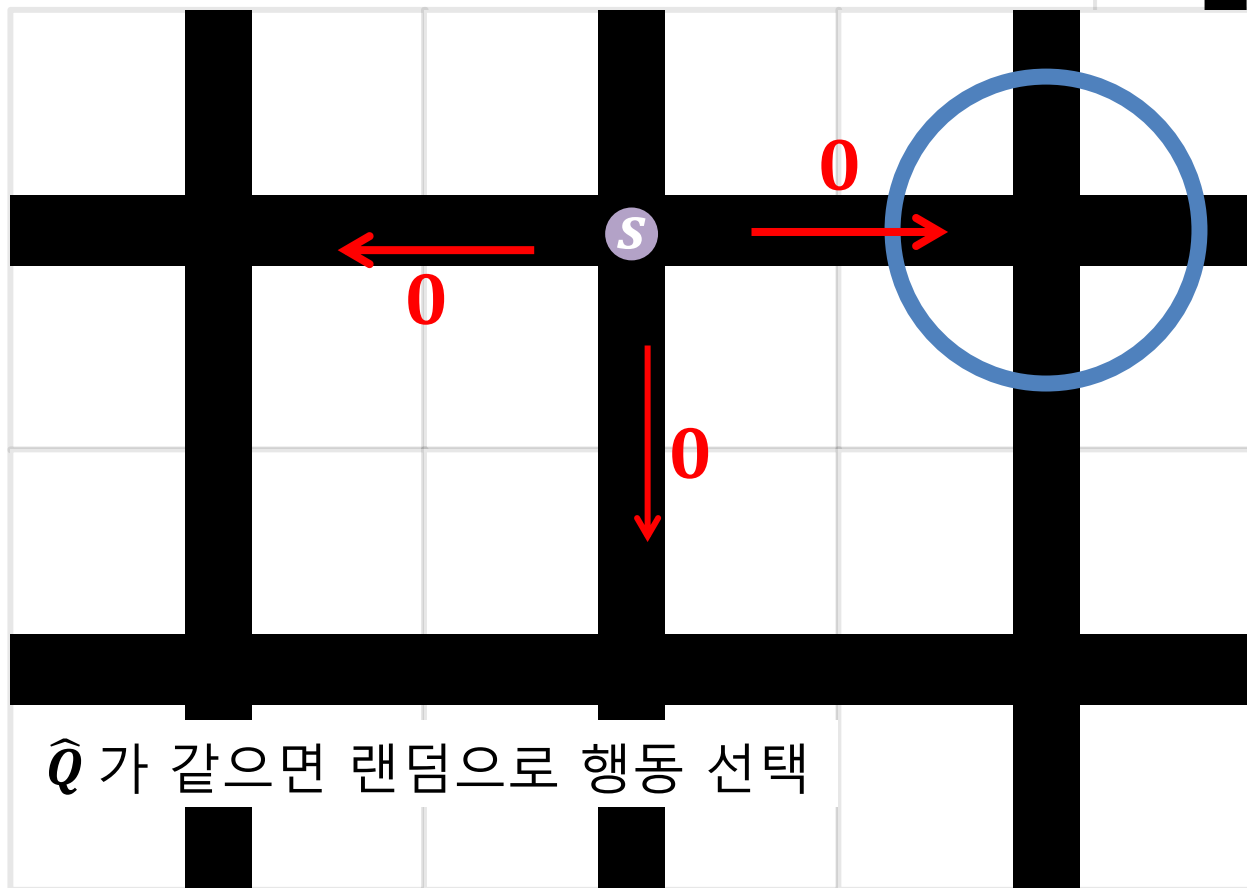
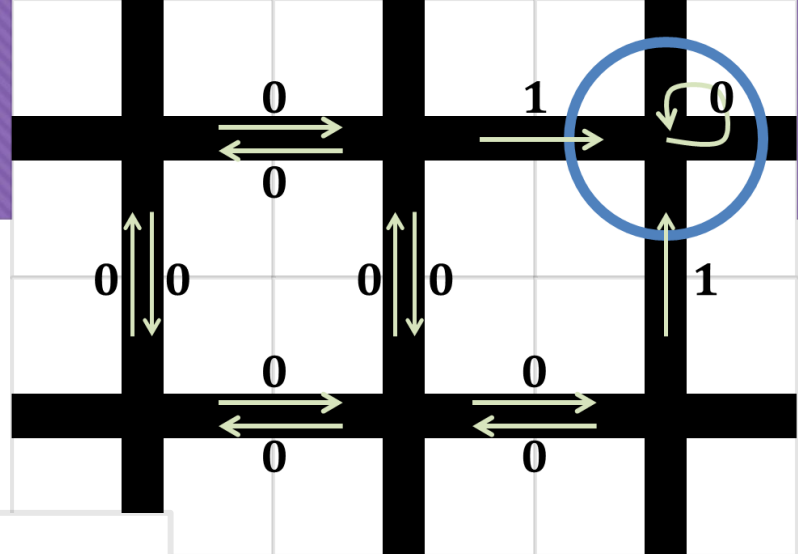
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

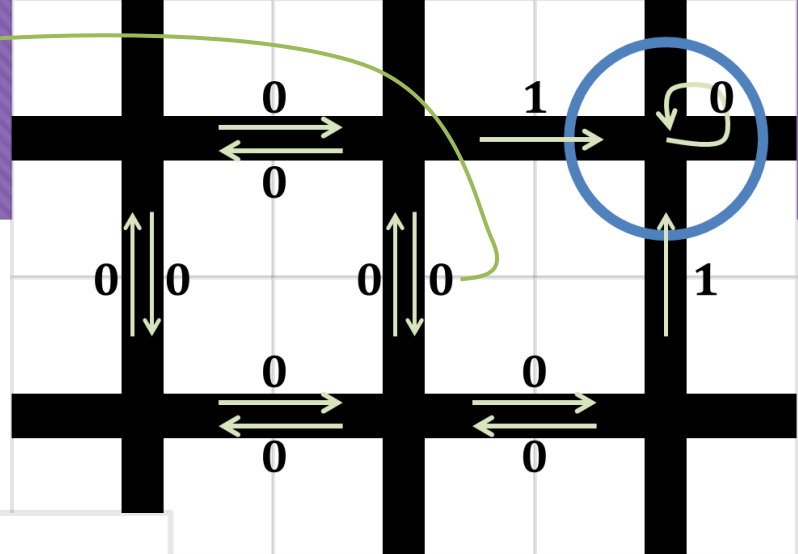
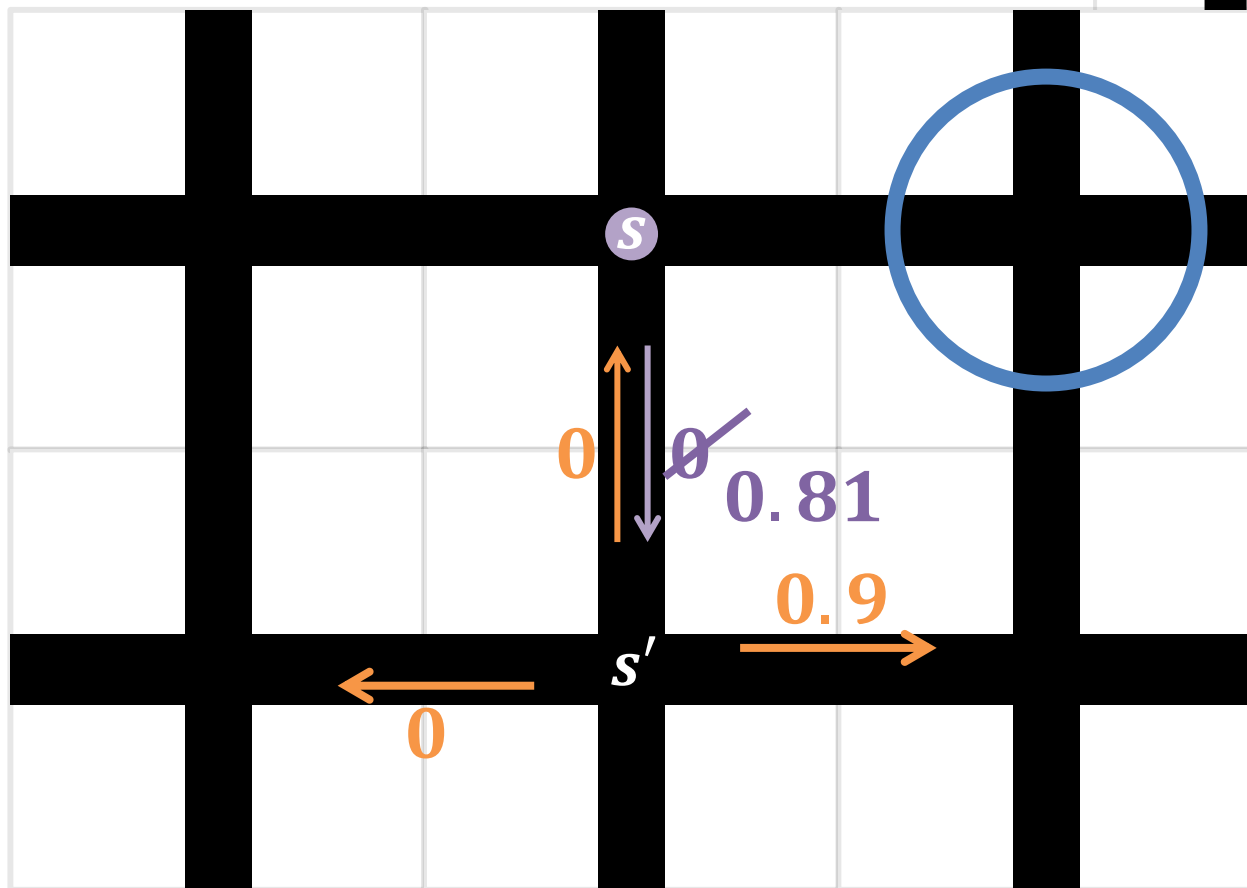


\hat{Q} 가 같으면 랜덤으로 행동 선택

Q-러닝

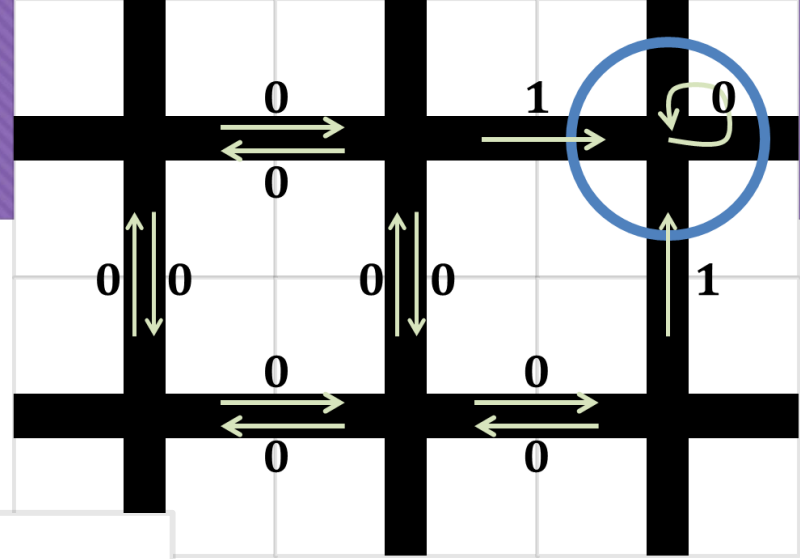
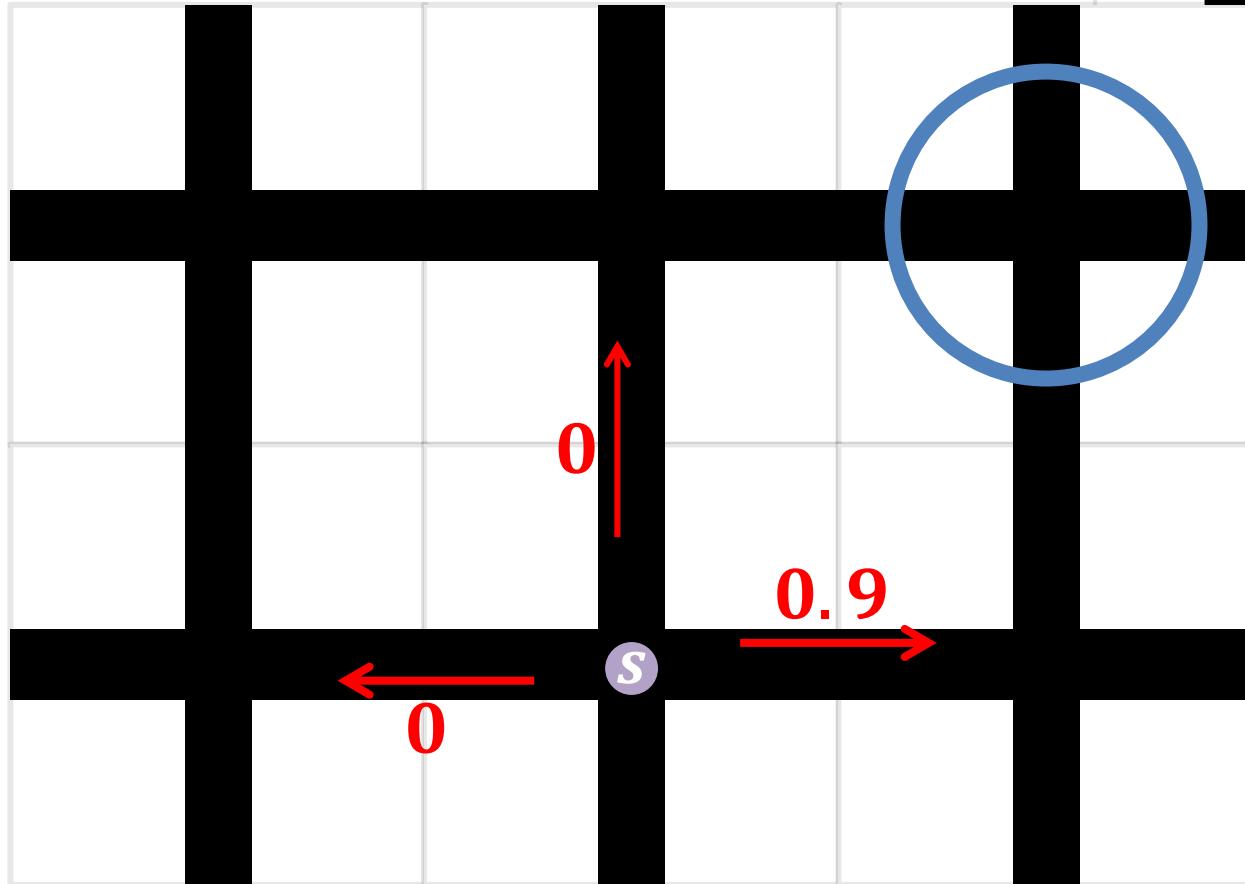
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



Q-러닝

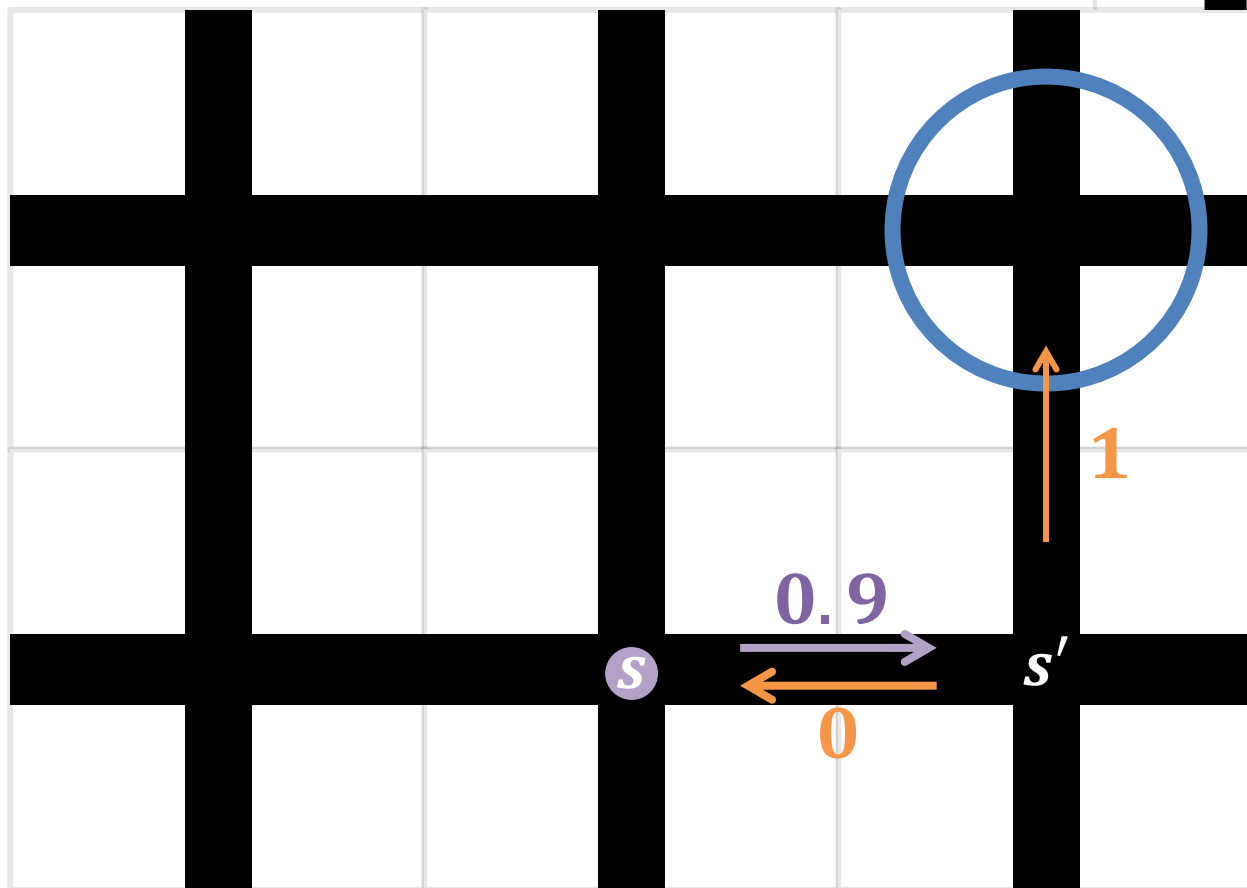
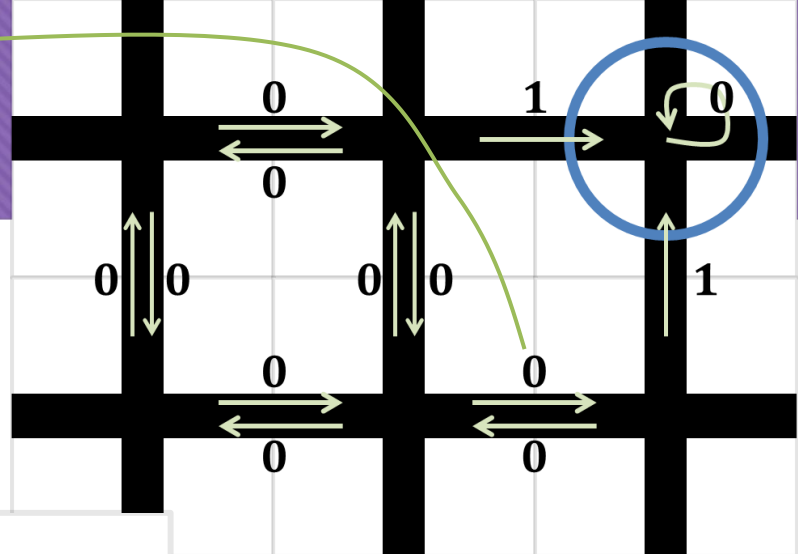
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

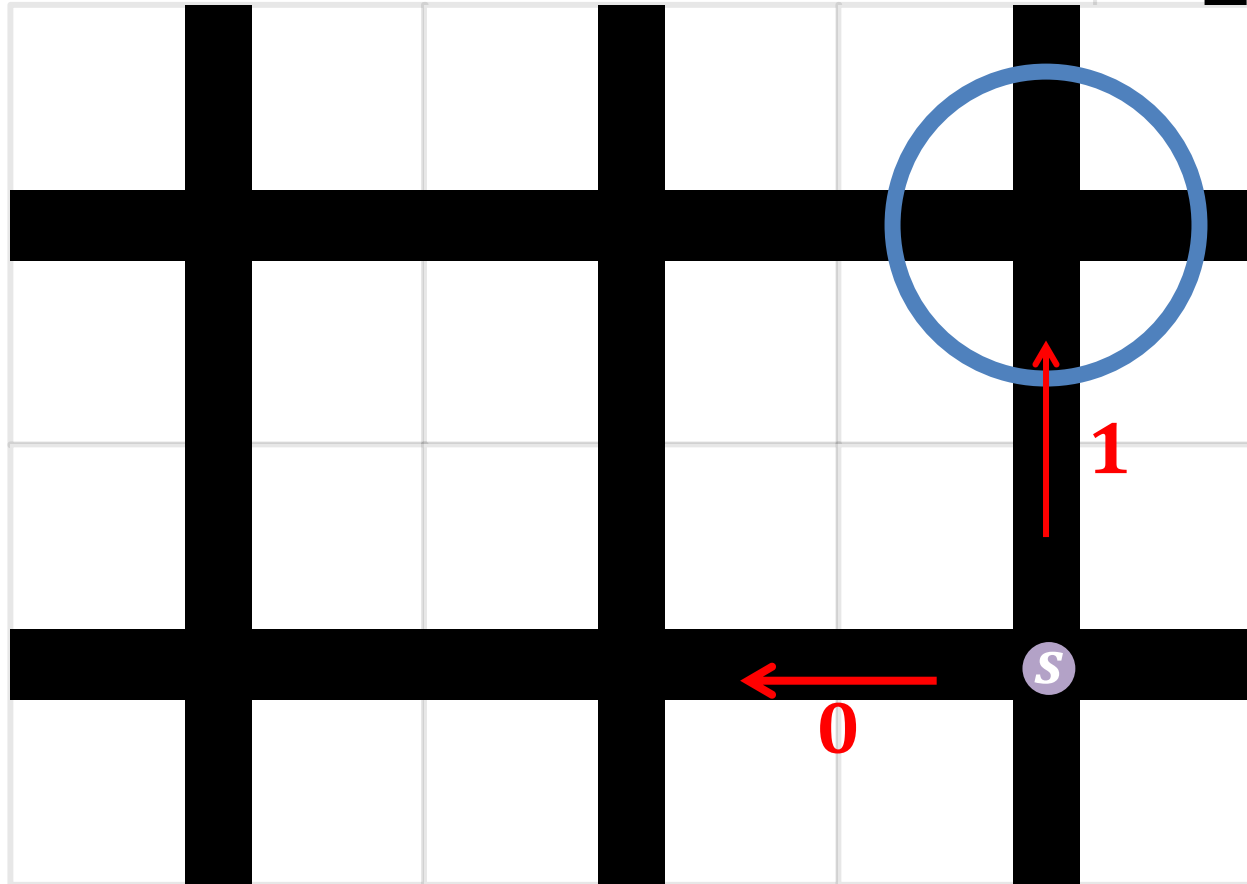
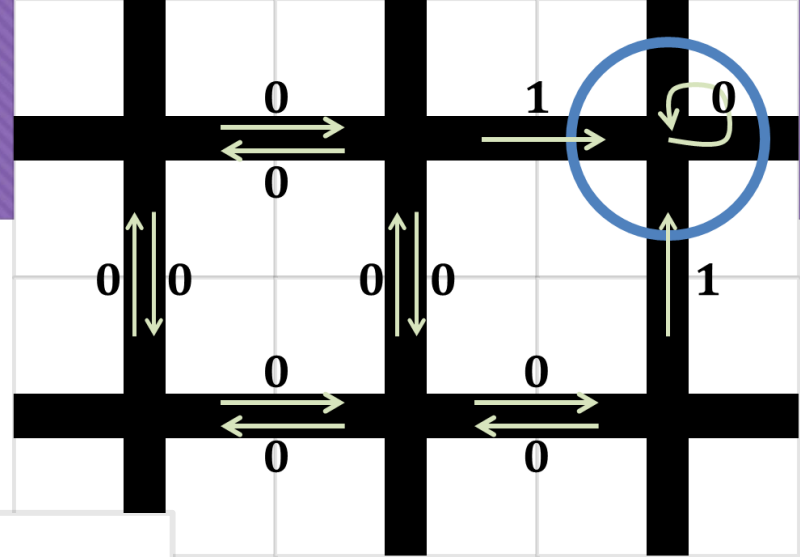
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 1



Q-러닝

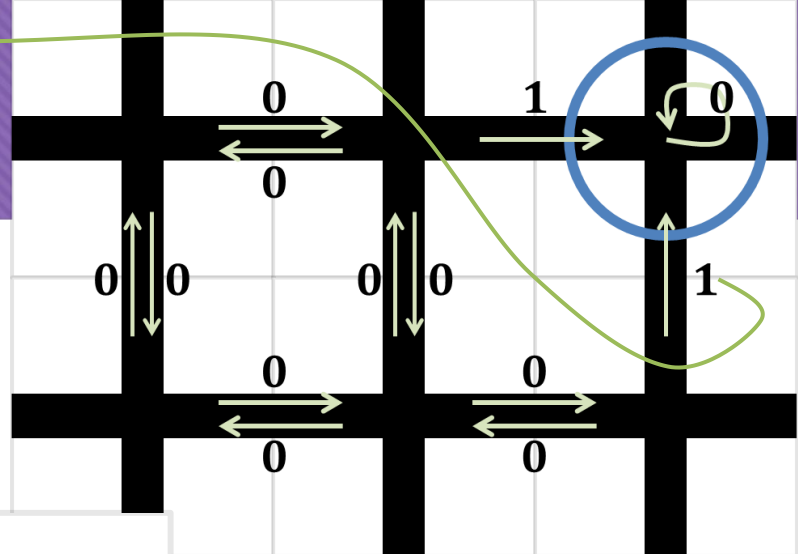
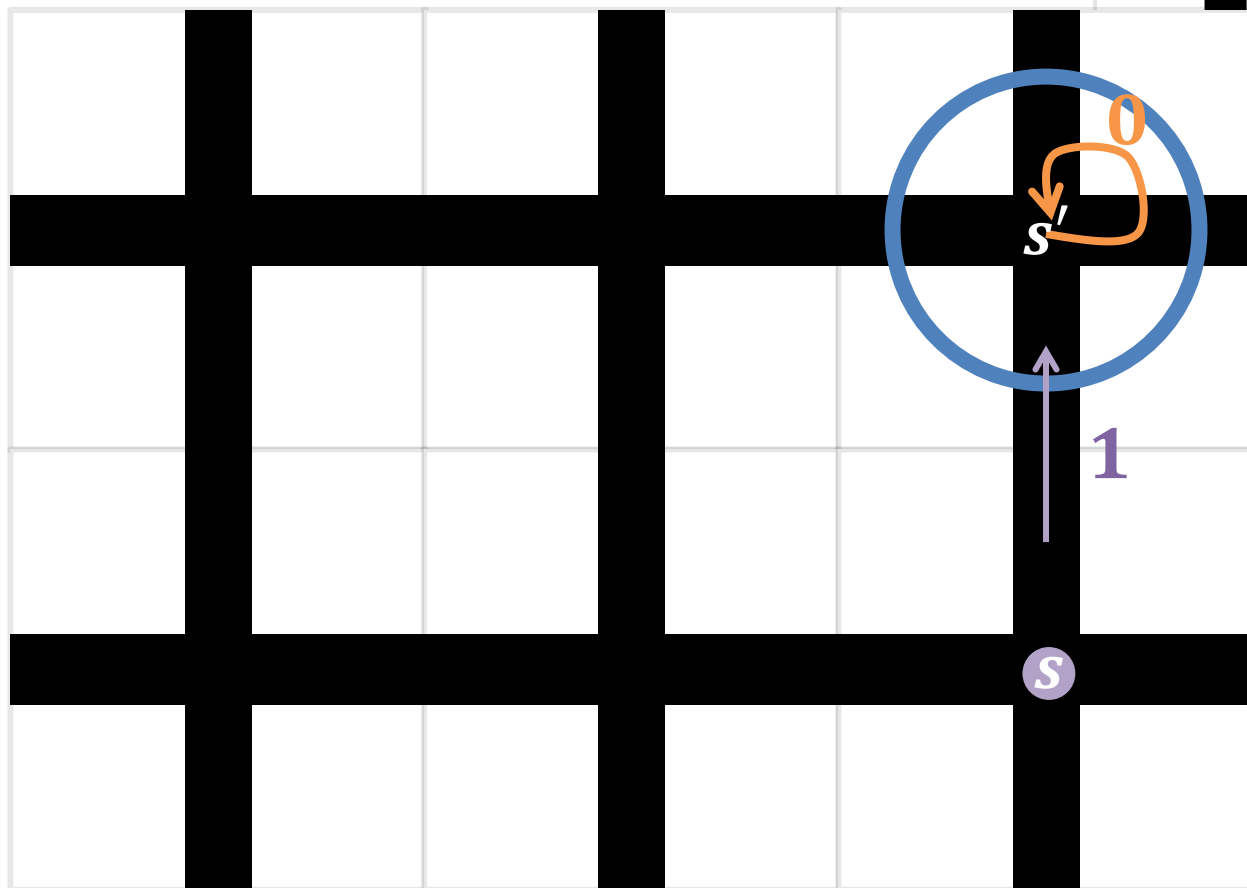
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

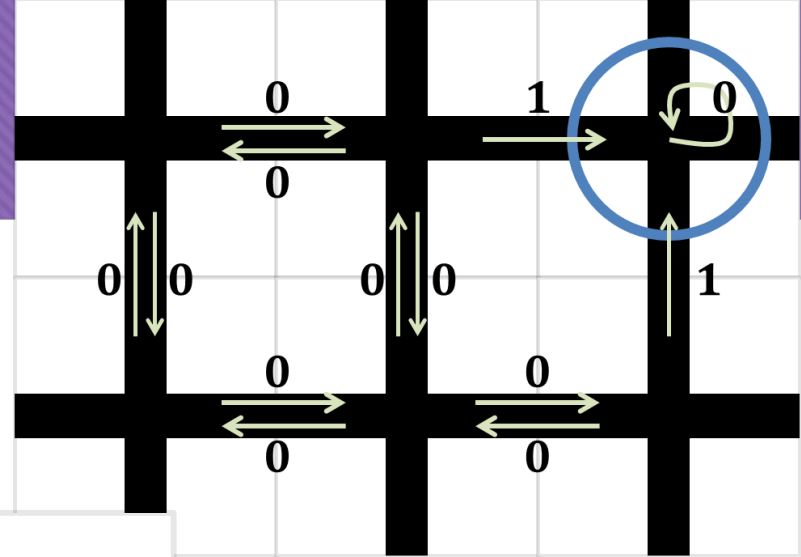
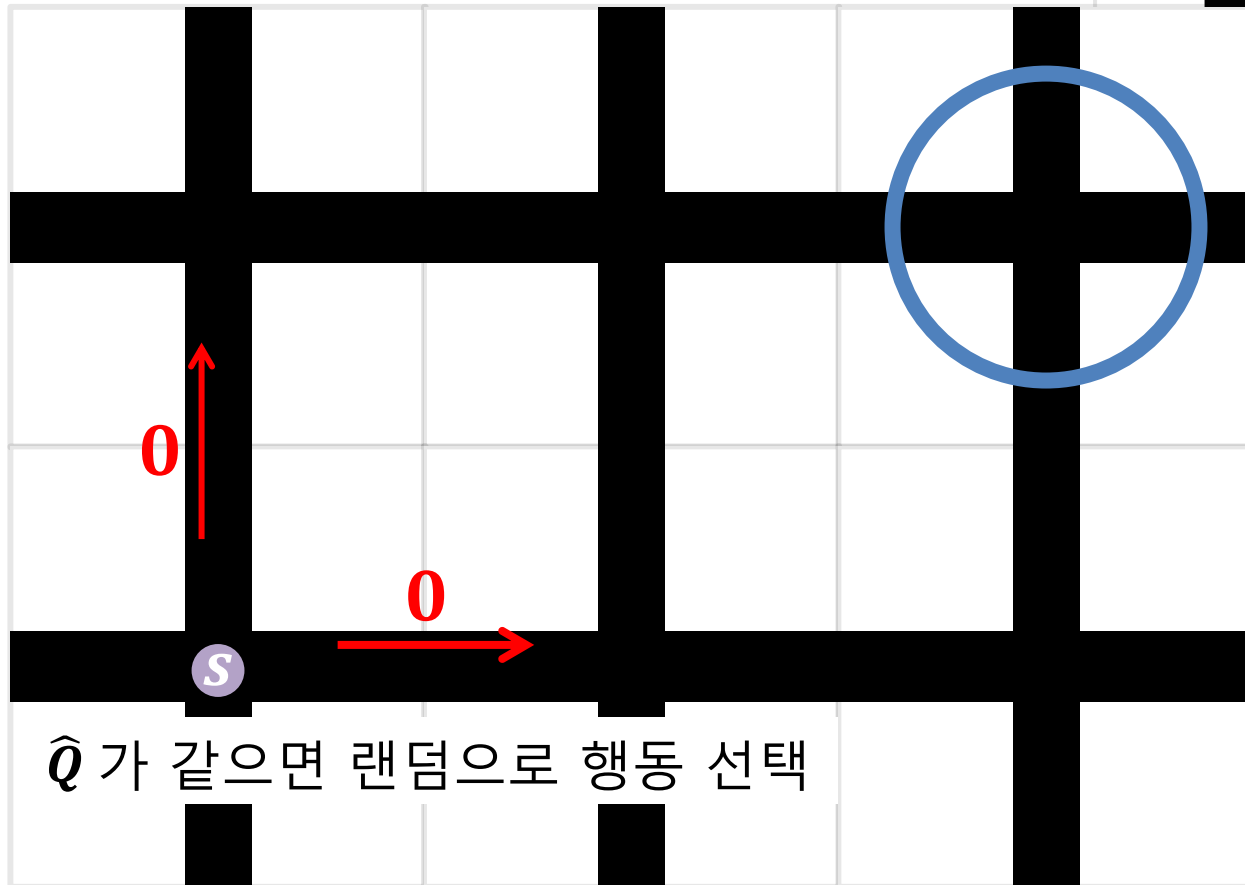
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

1 0



Q-러닝

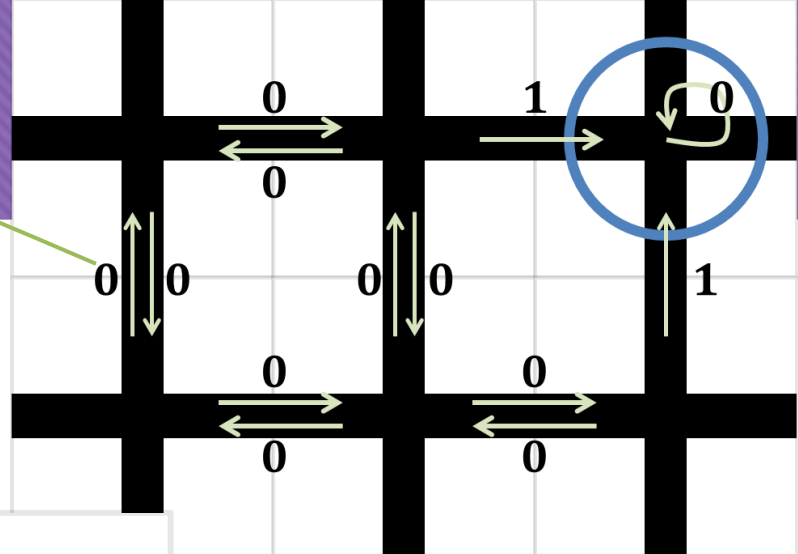
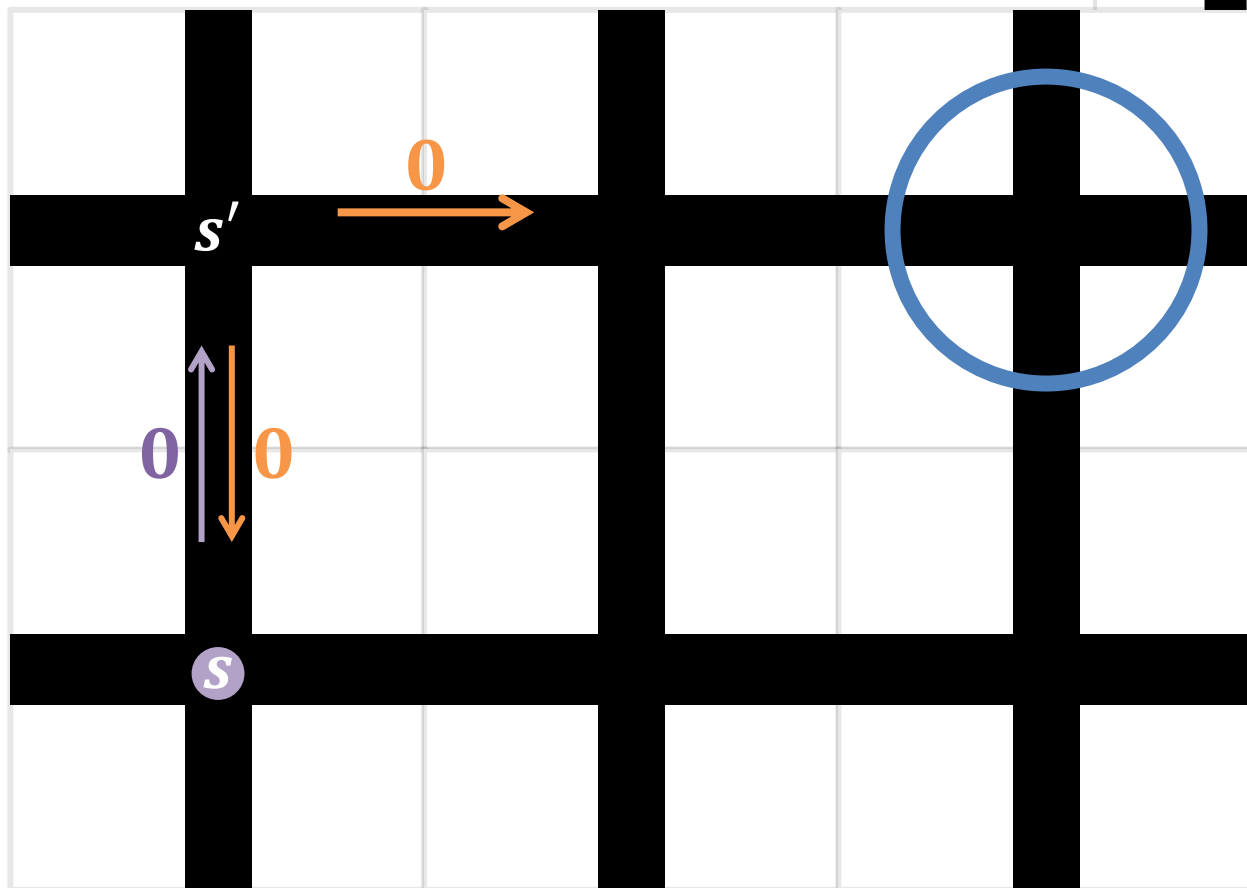
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

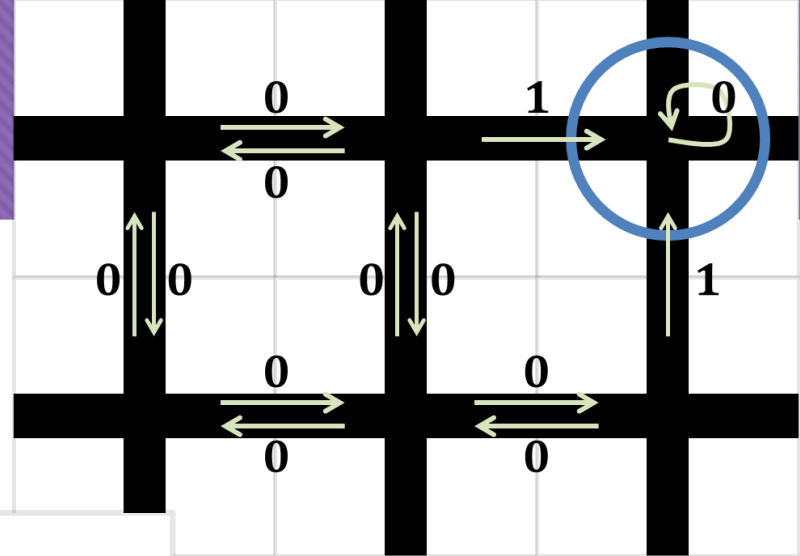
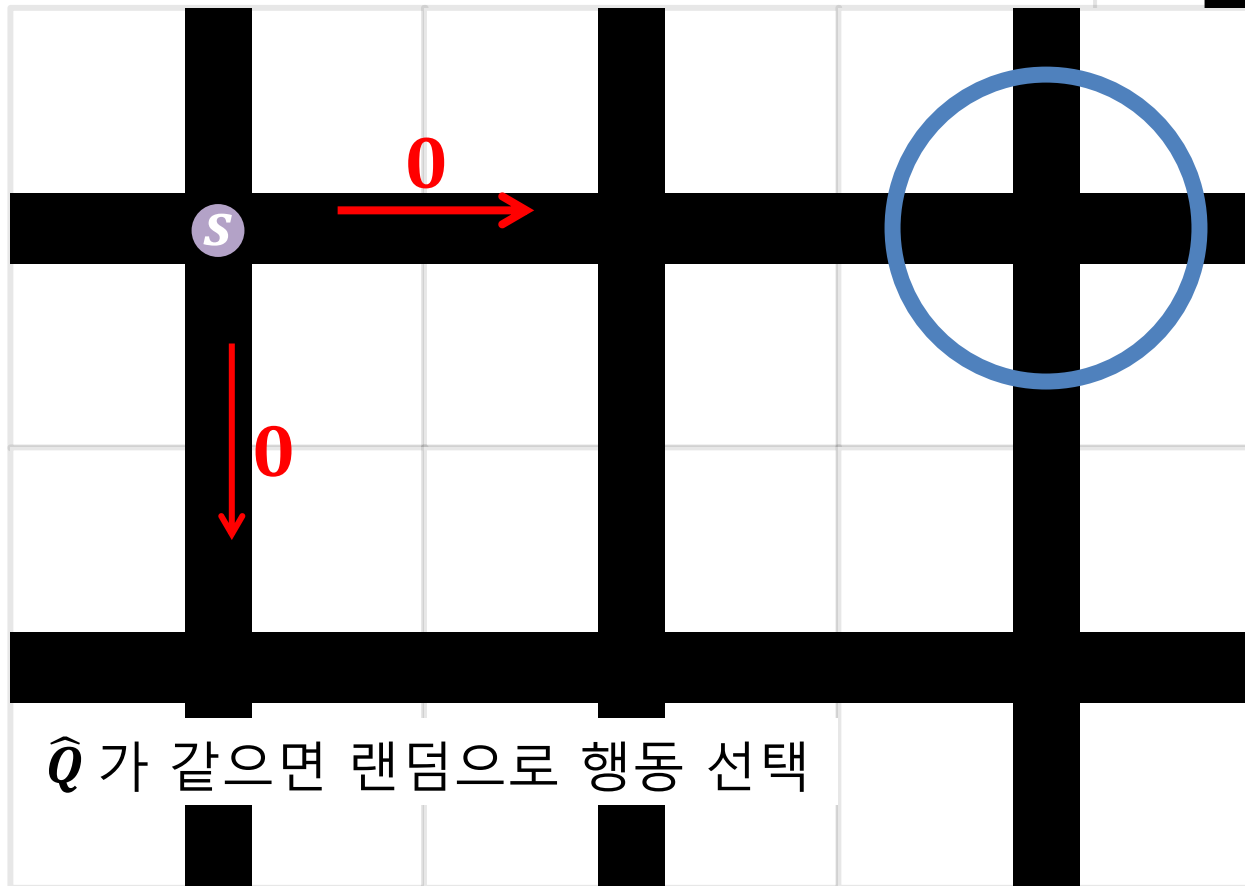
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0



Q-러닝

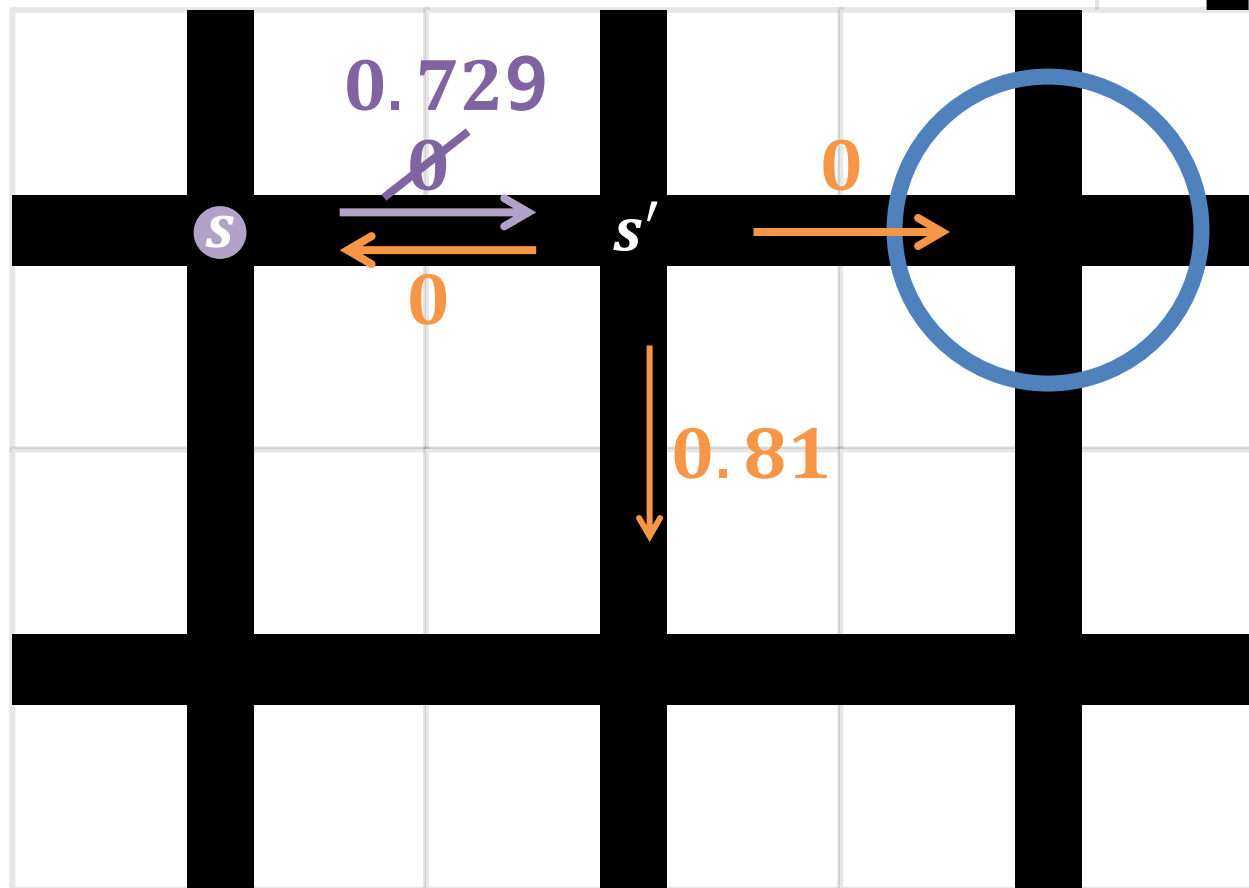
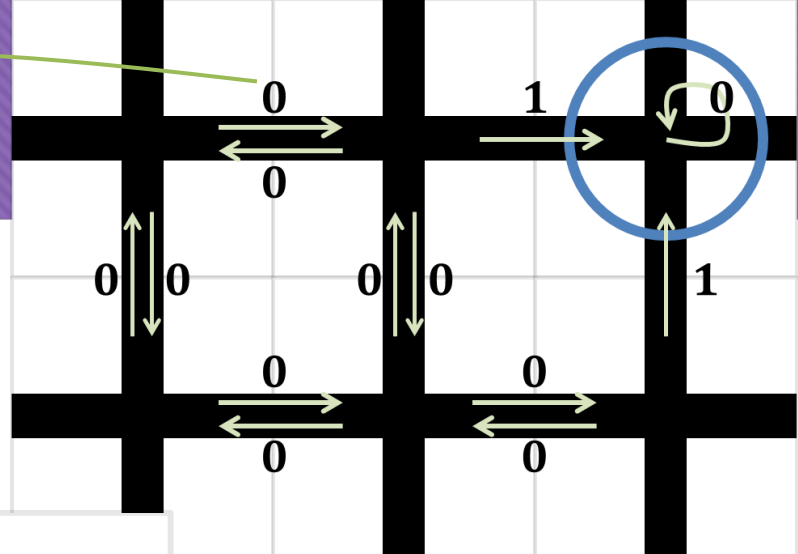
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

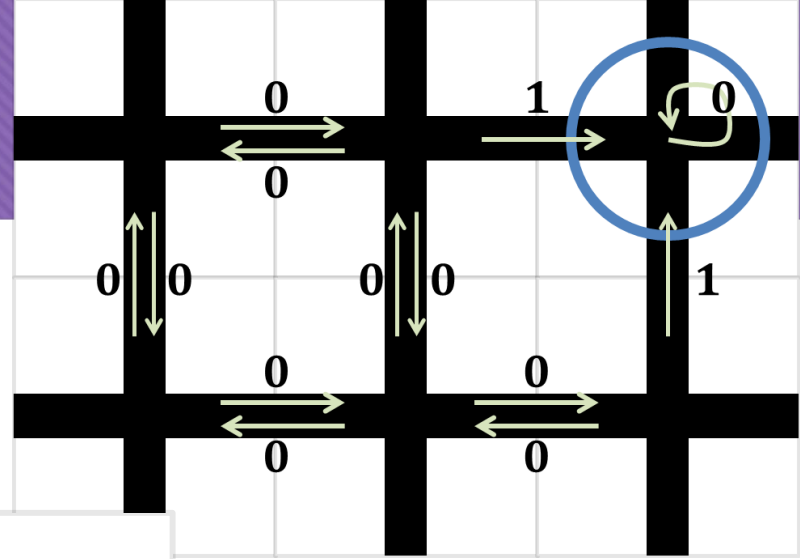
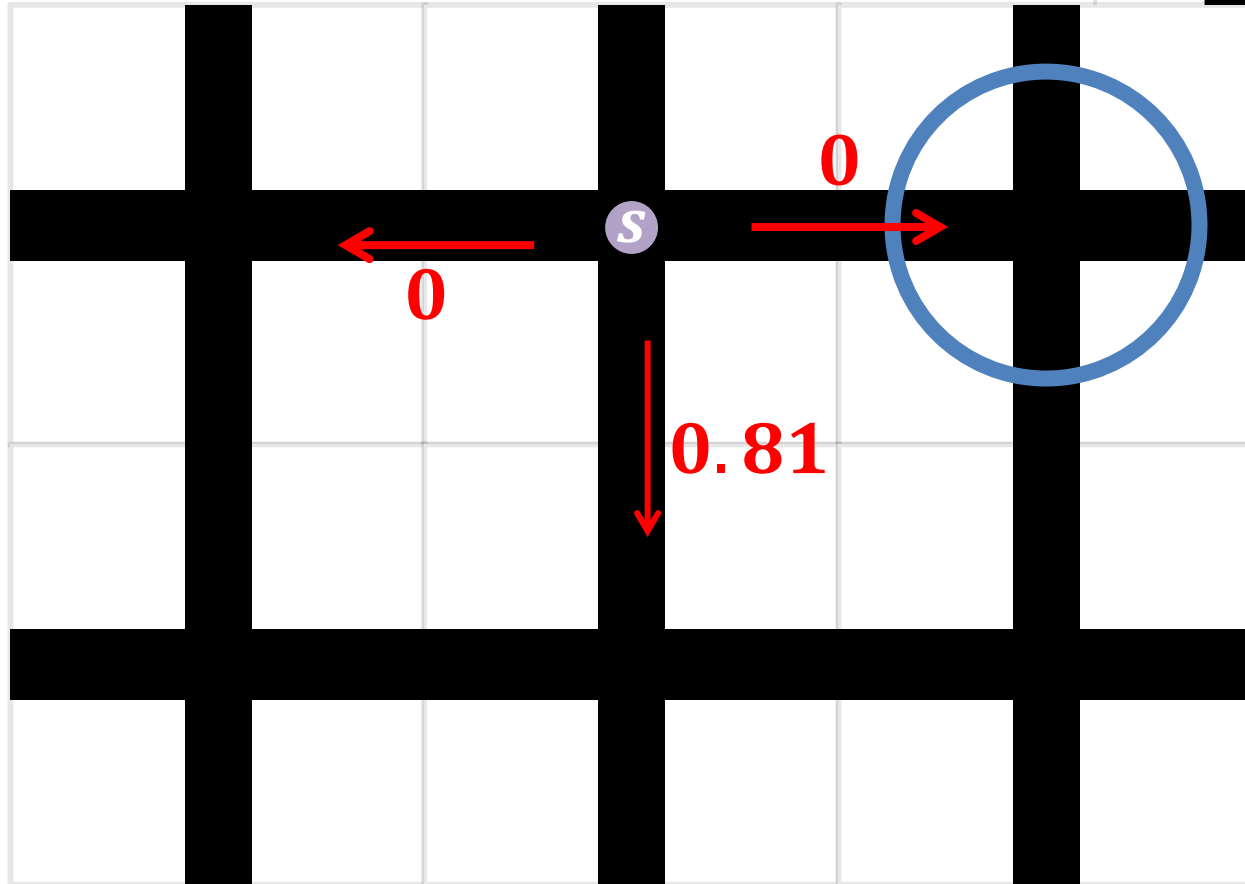
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.81



Q-러닝

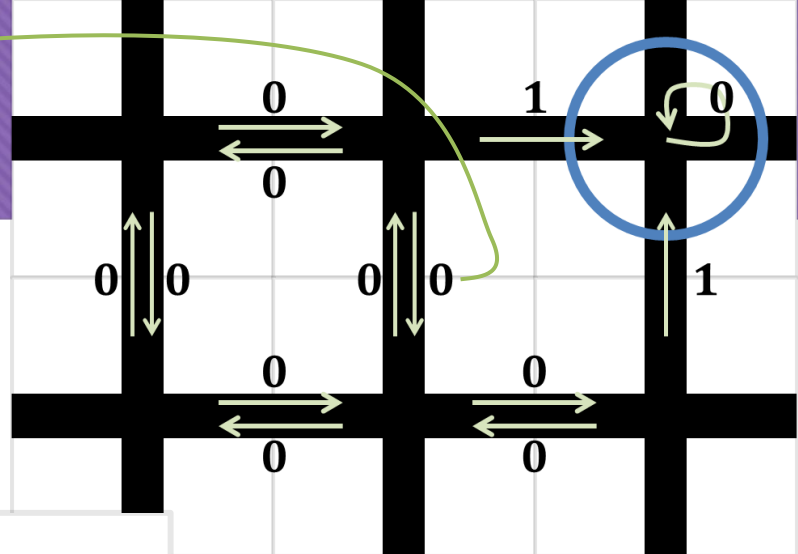
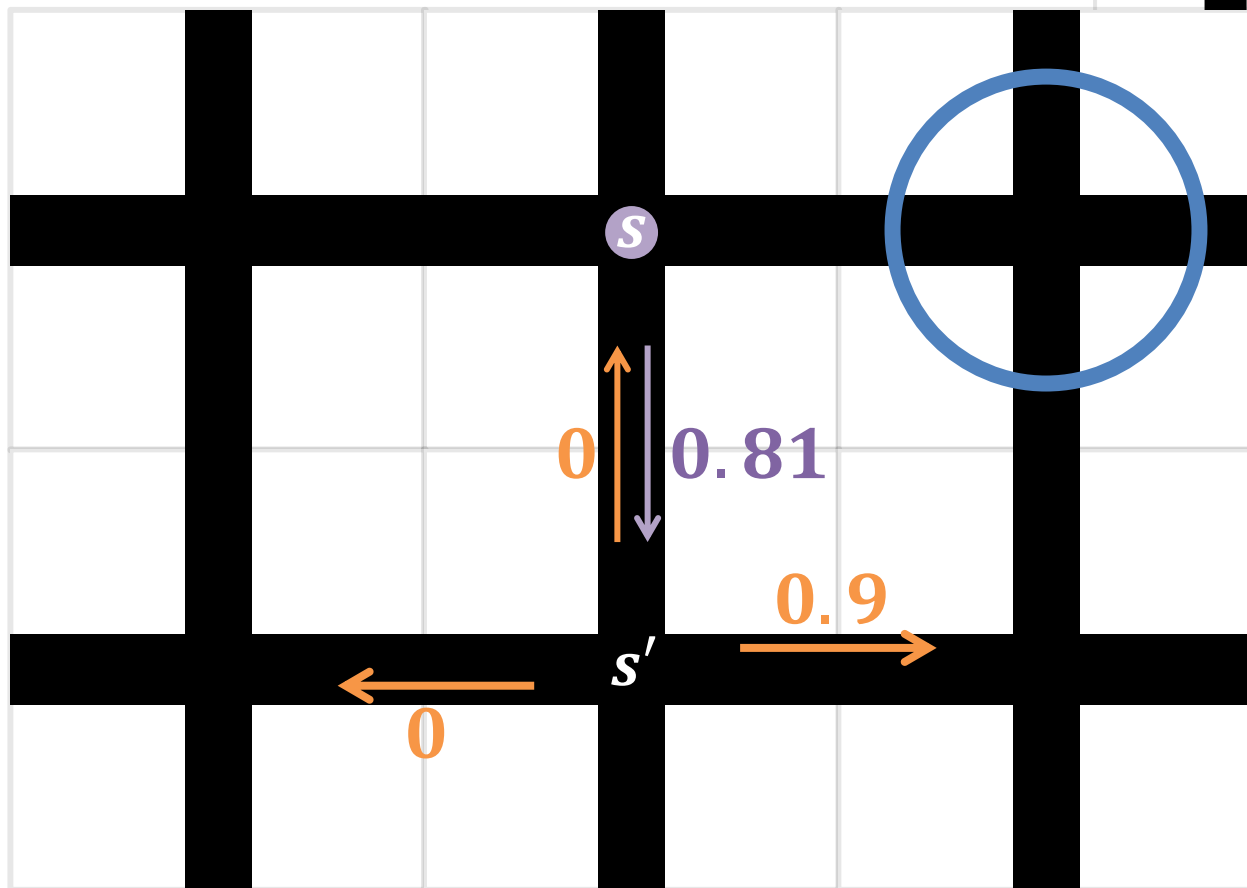
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

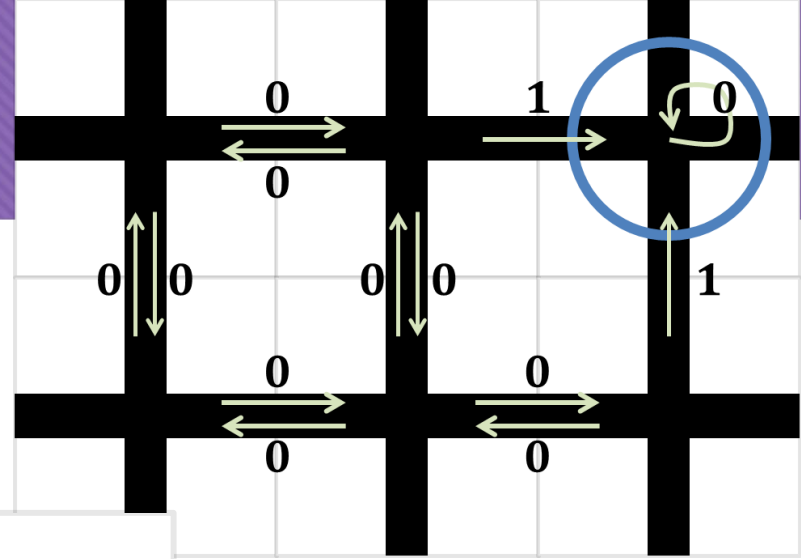
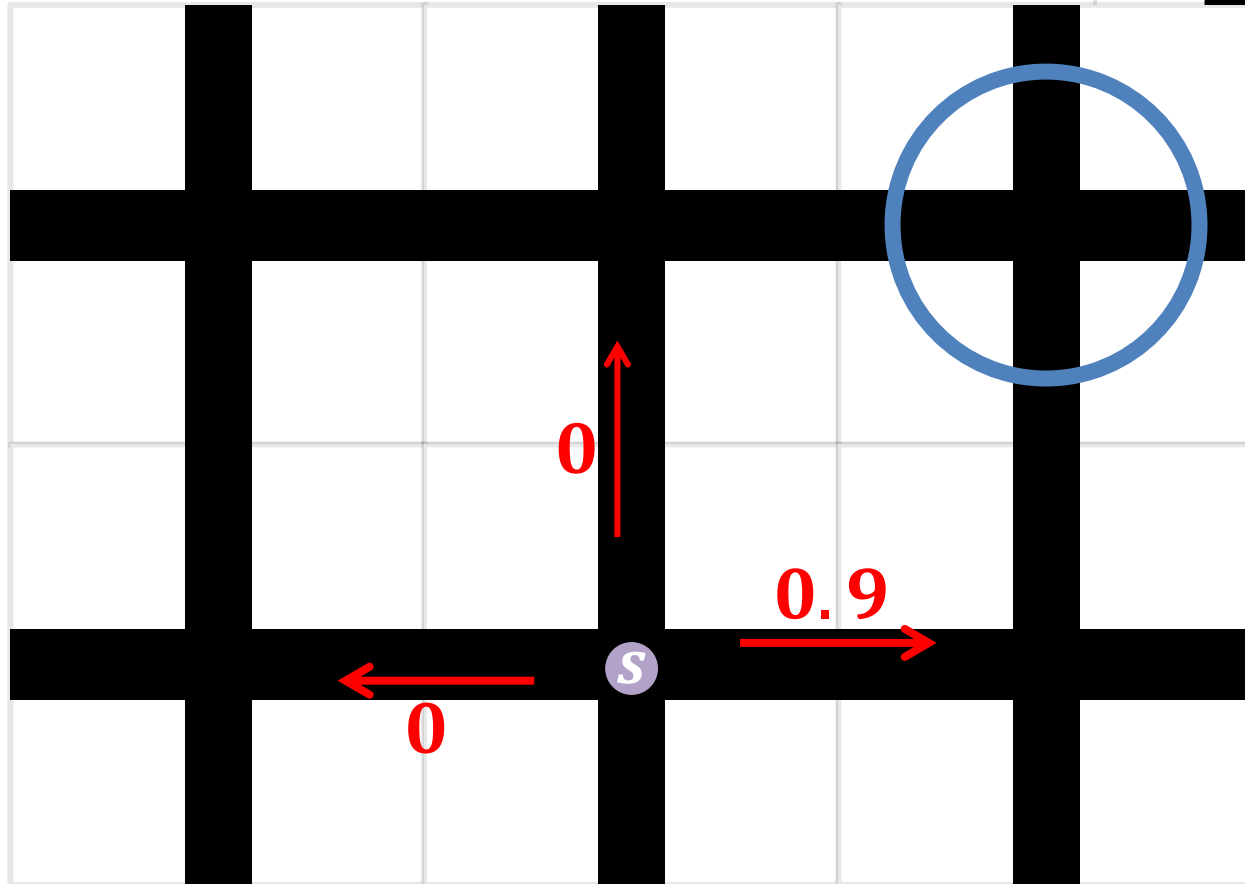
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



Q-러닝

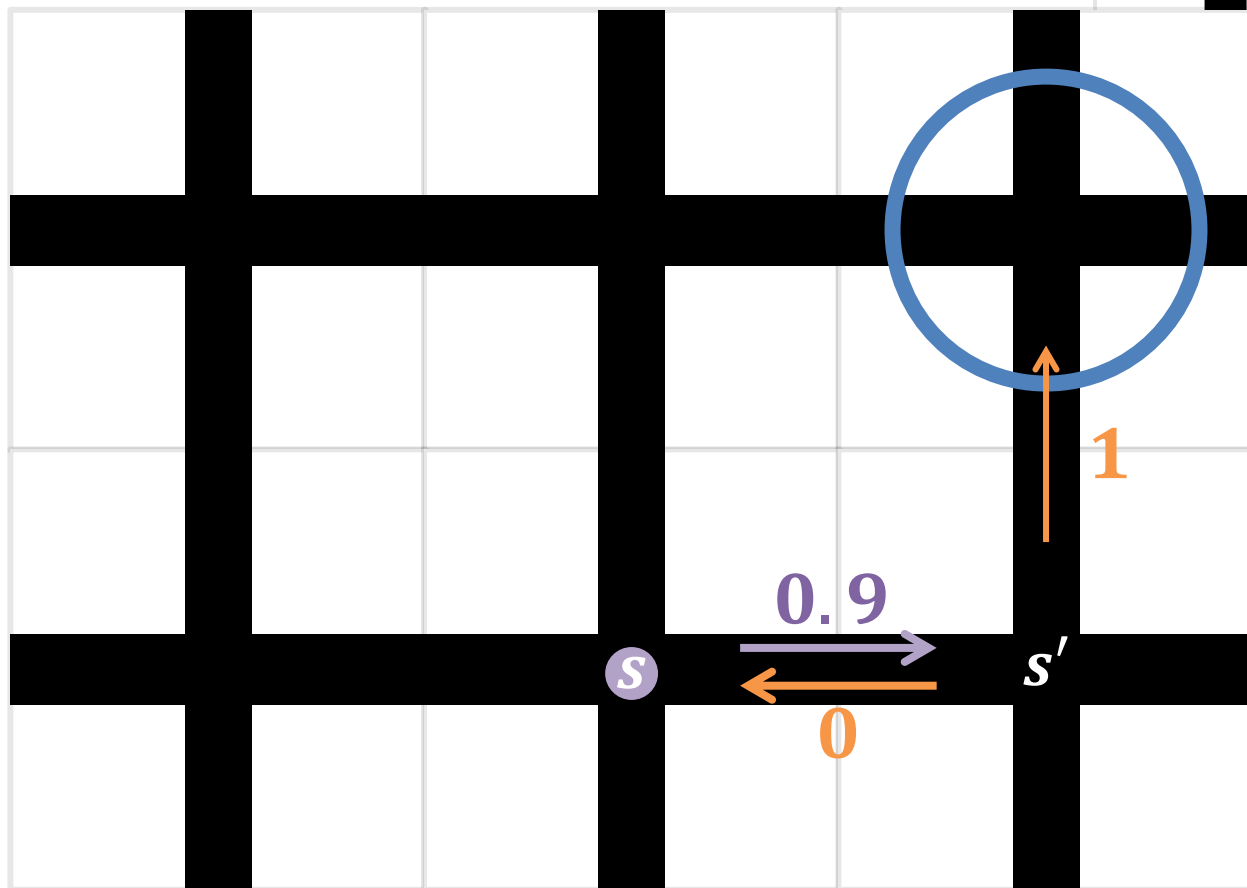
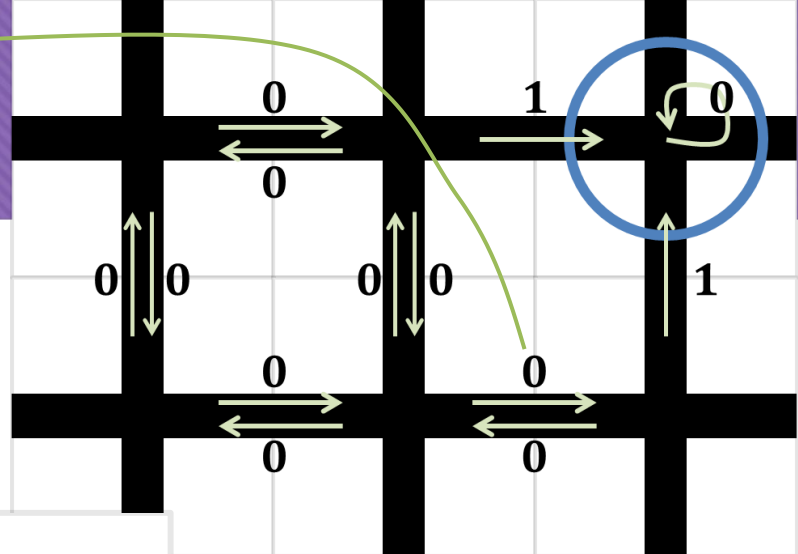
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

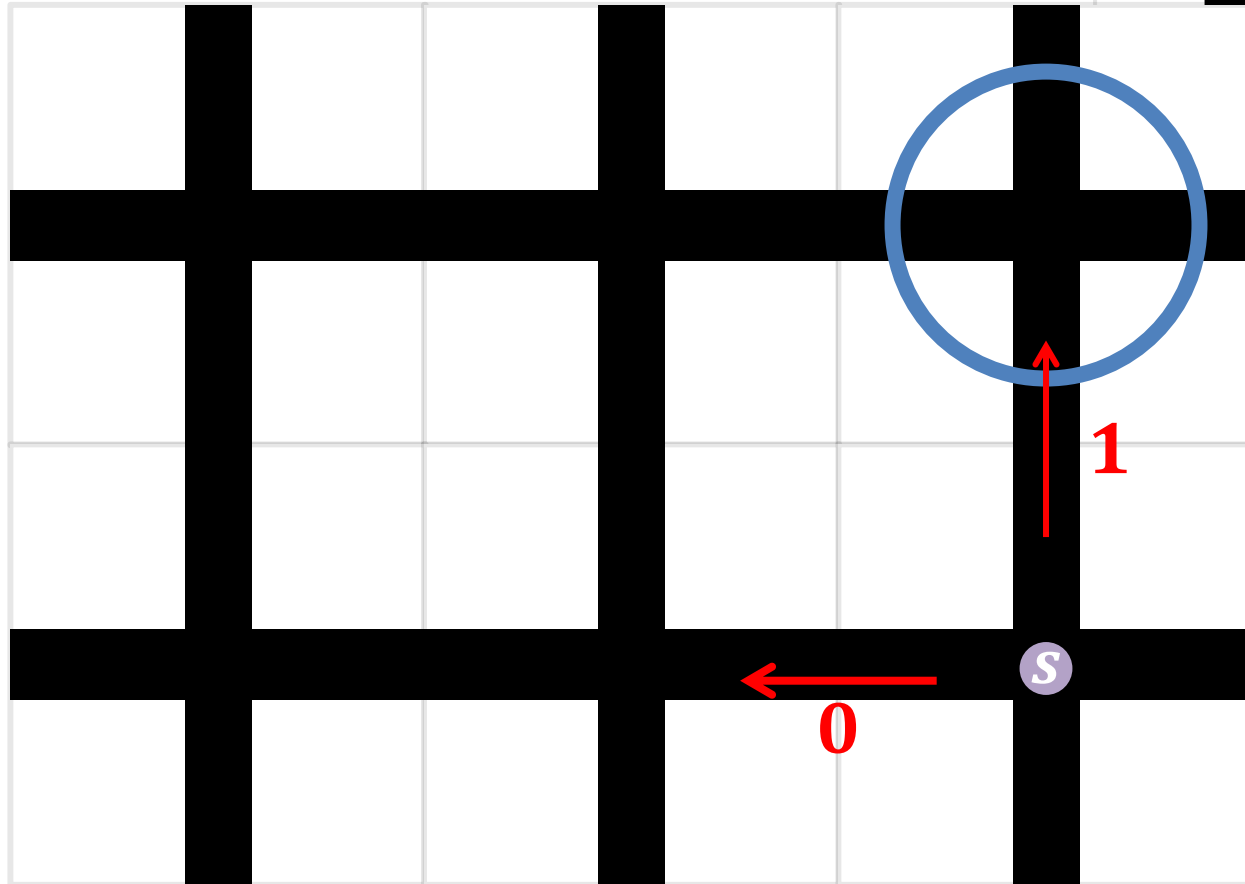
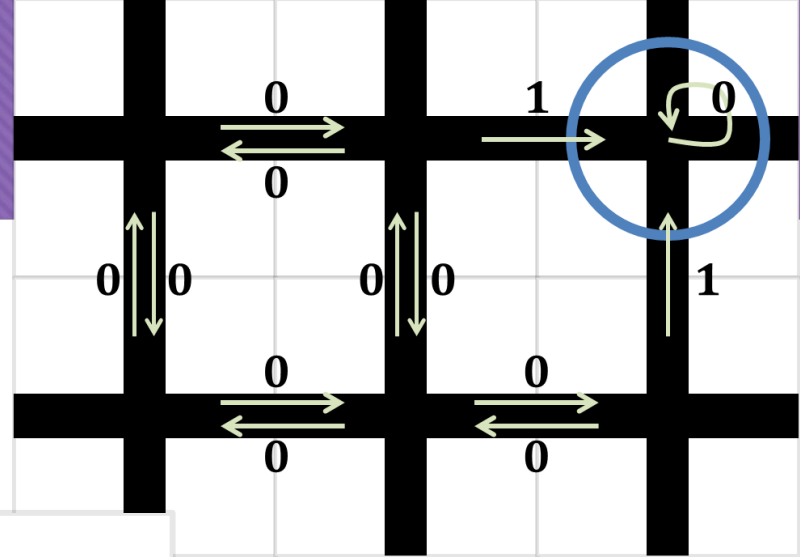
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 1



Q-러닝

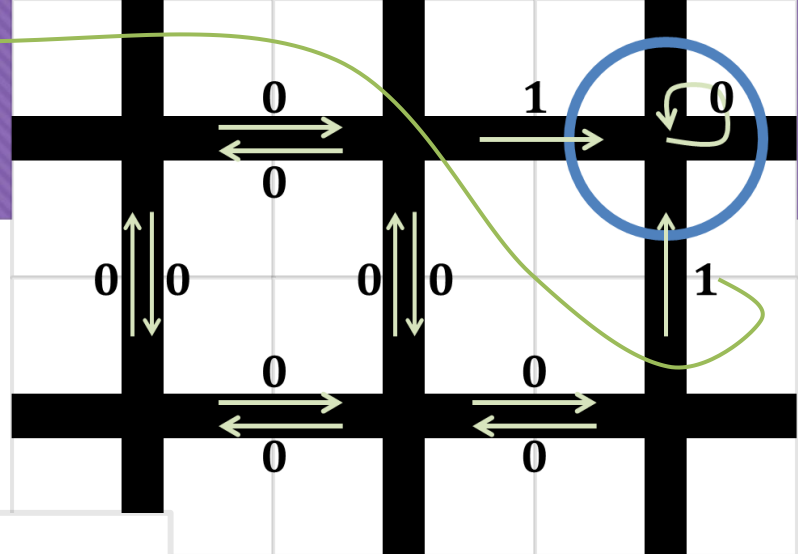
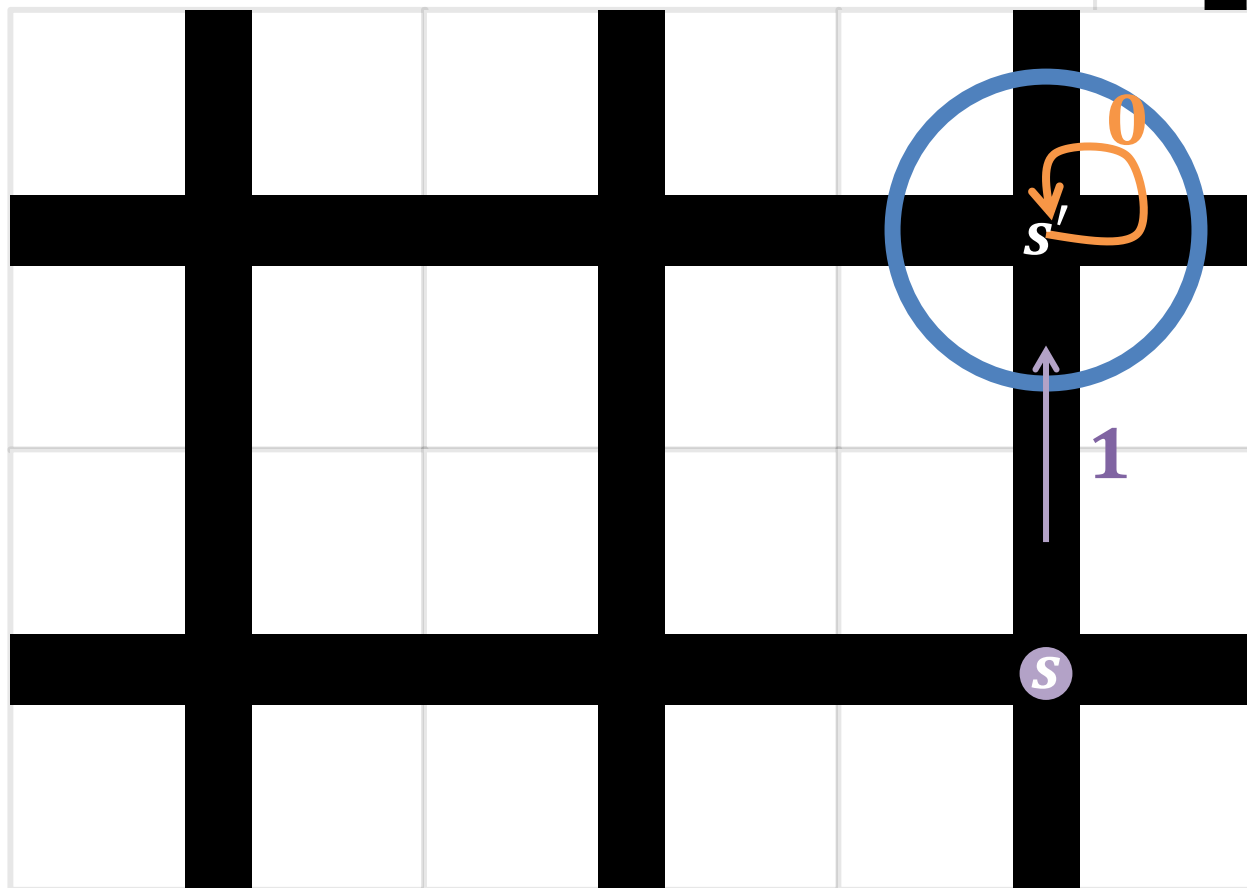
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

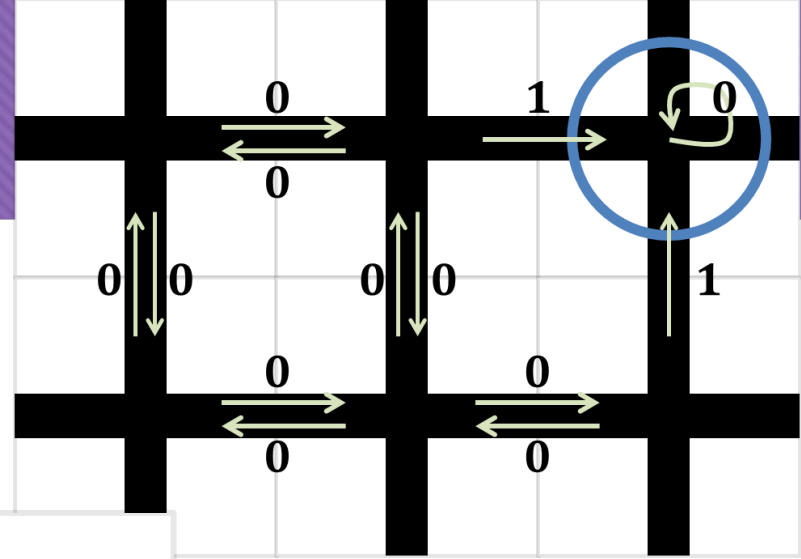
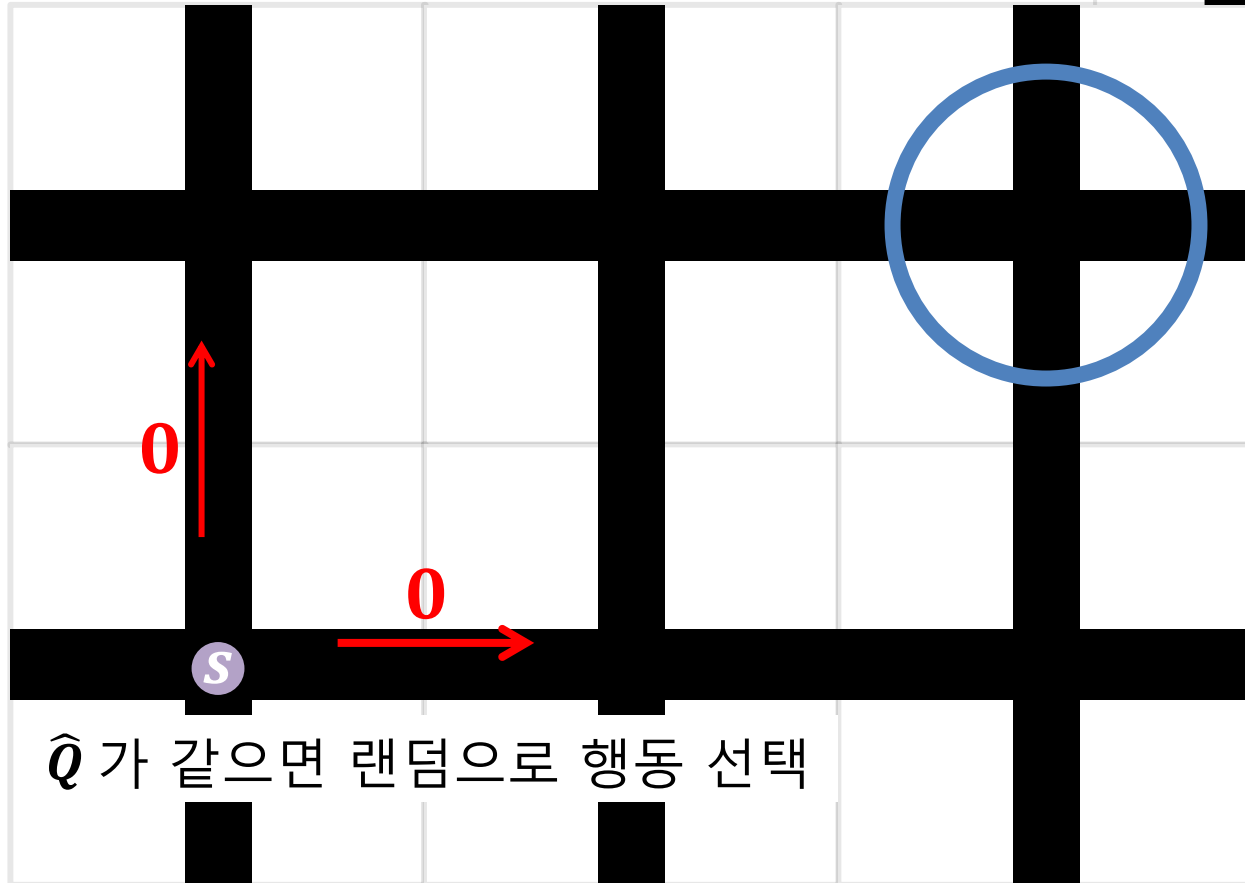
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

1 0



Q-러닝

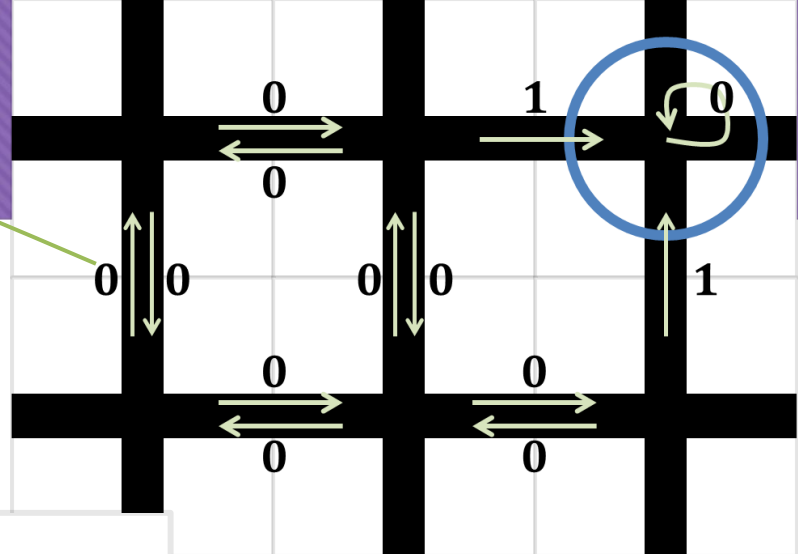
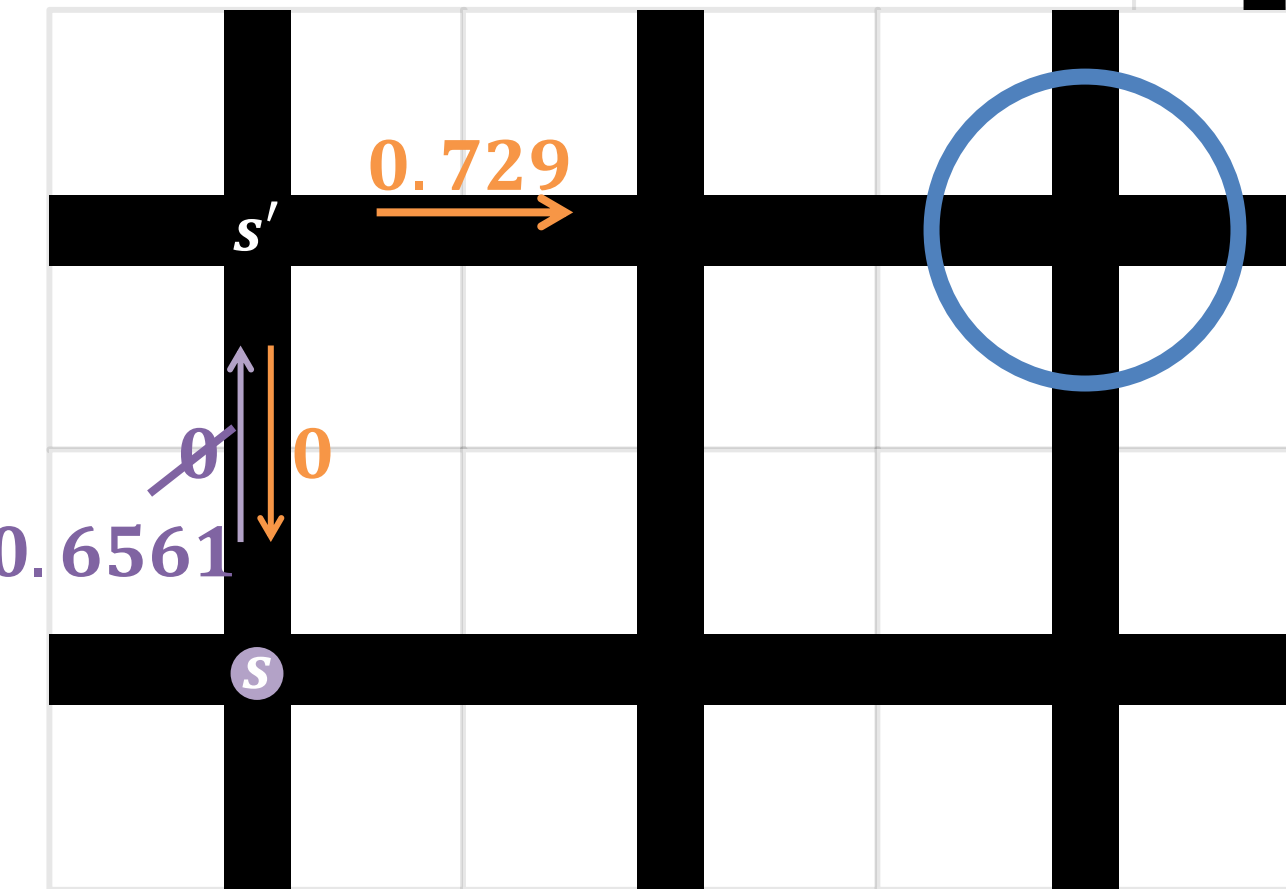
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

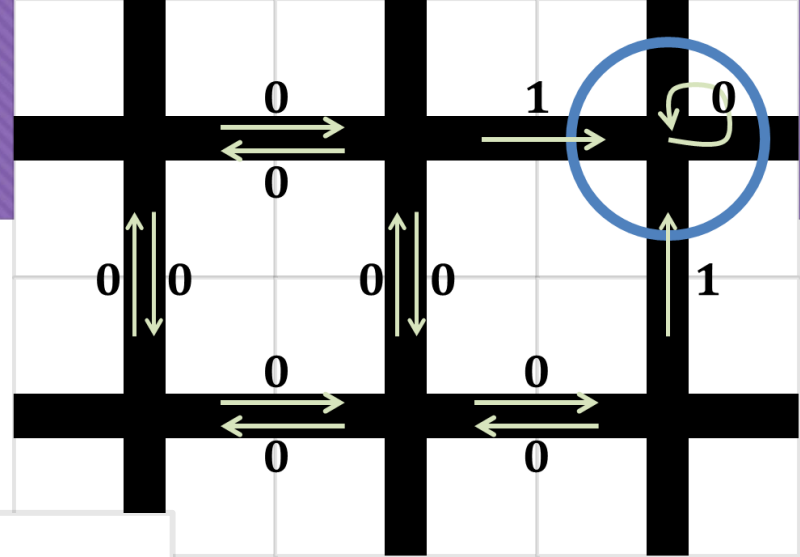
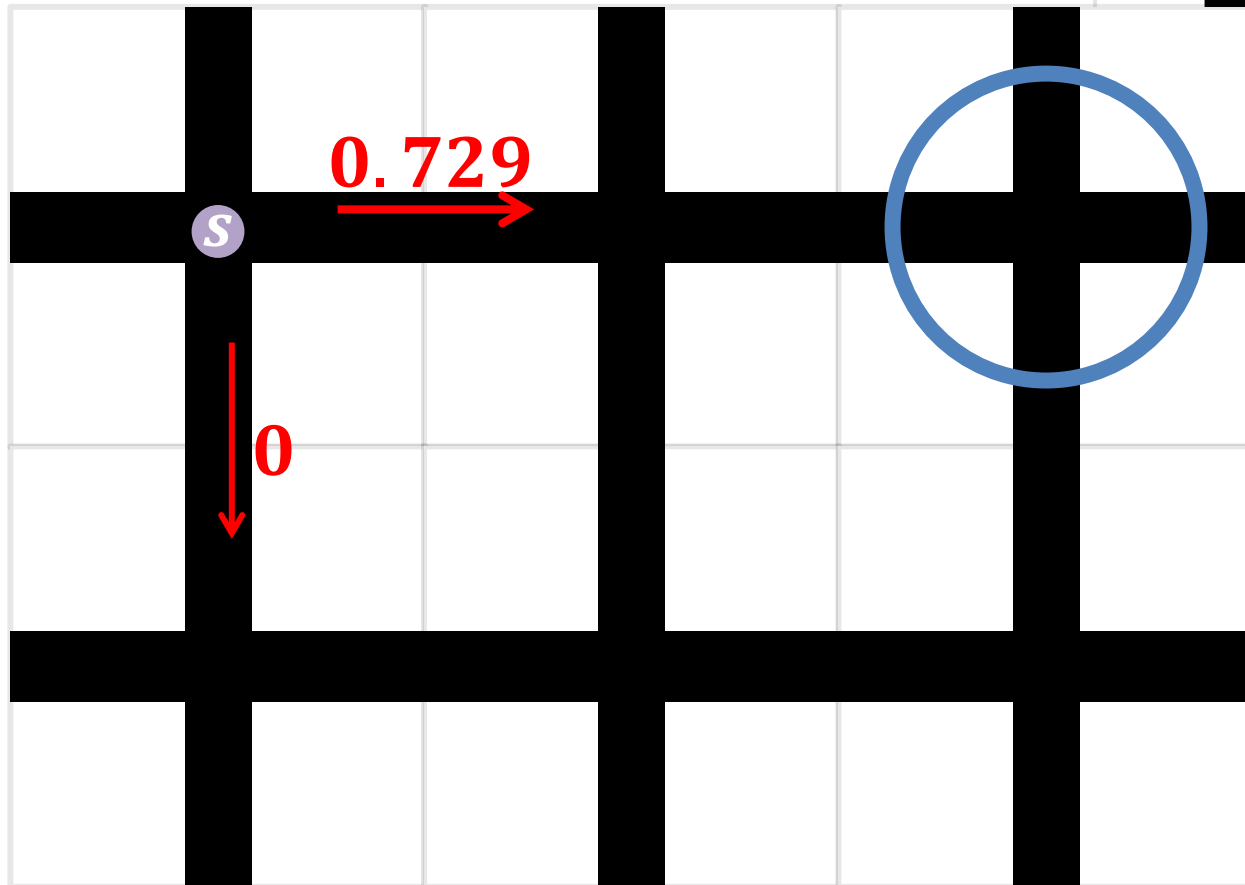
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.729



Q-러닝

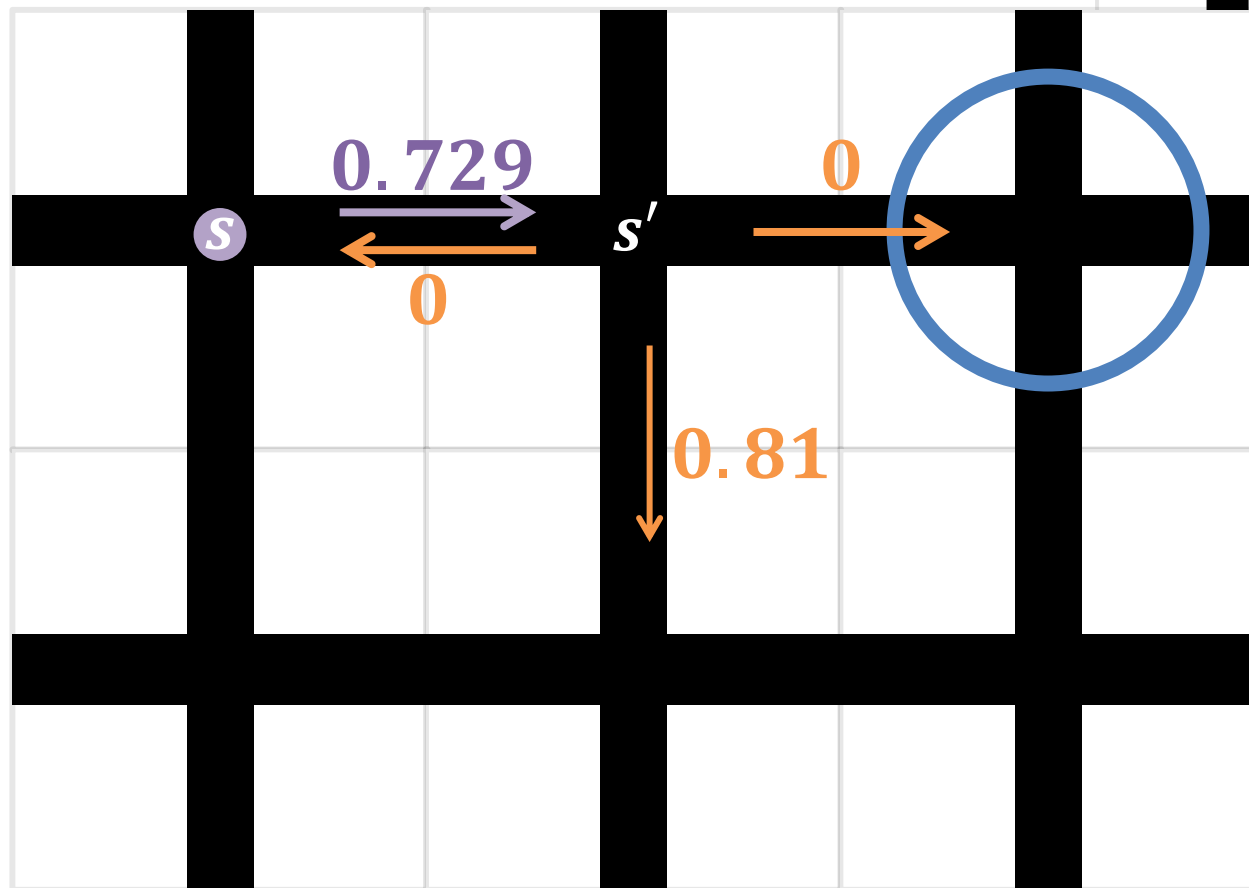
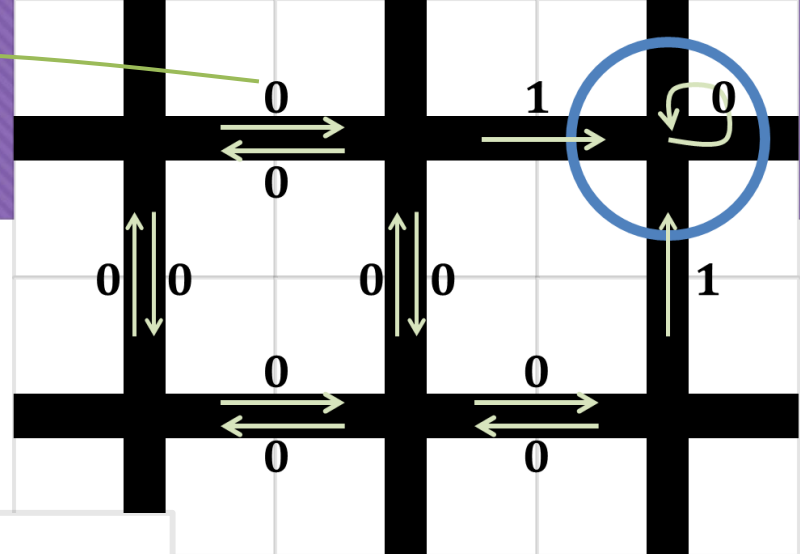
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

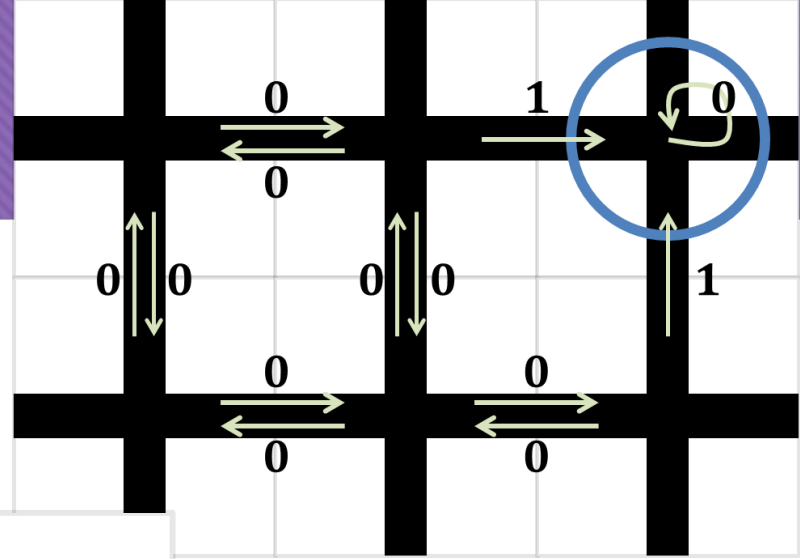
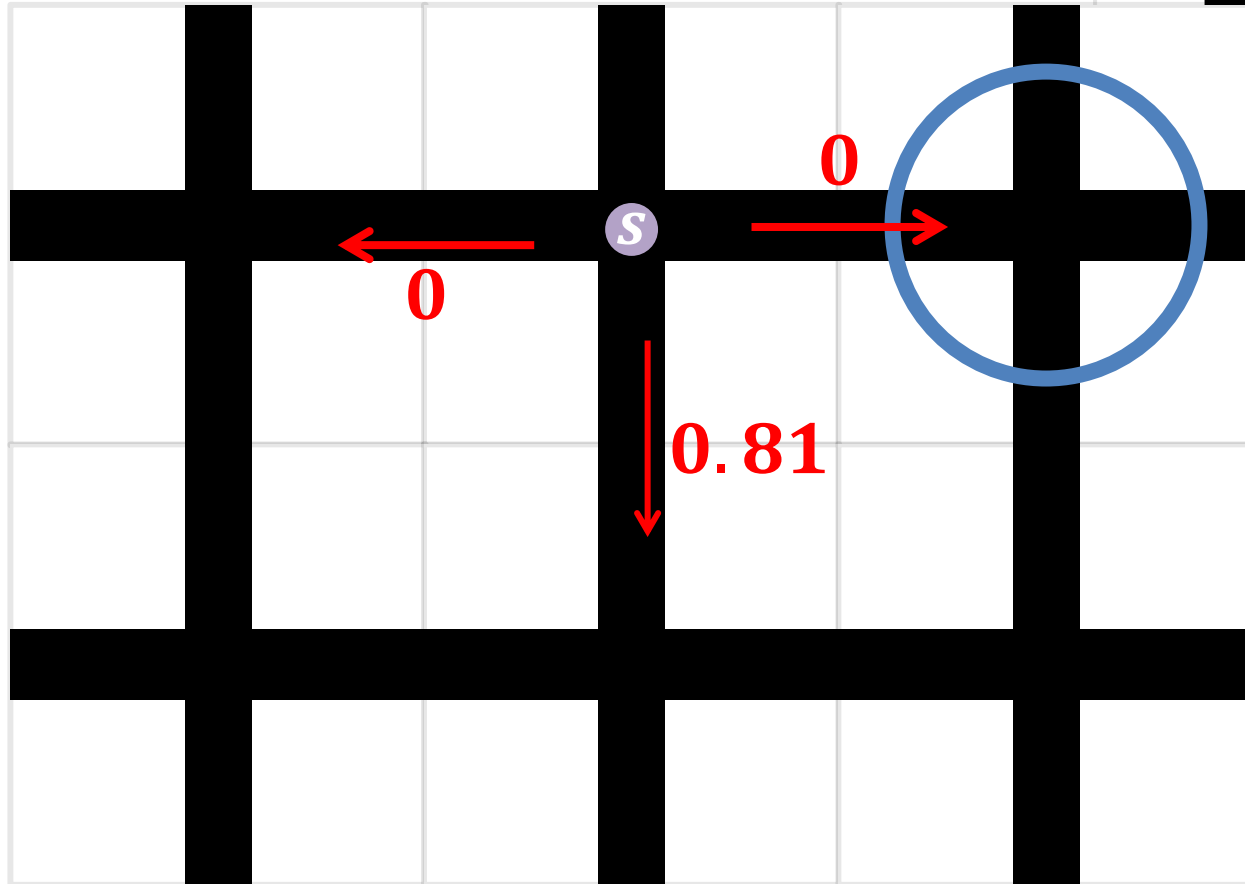
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.81



Q-러닝

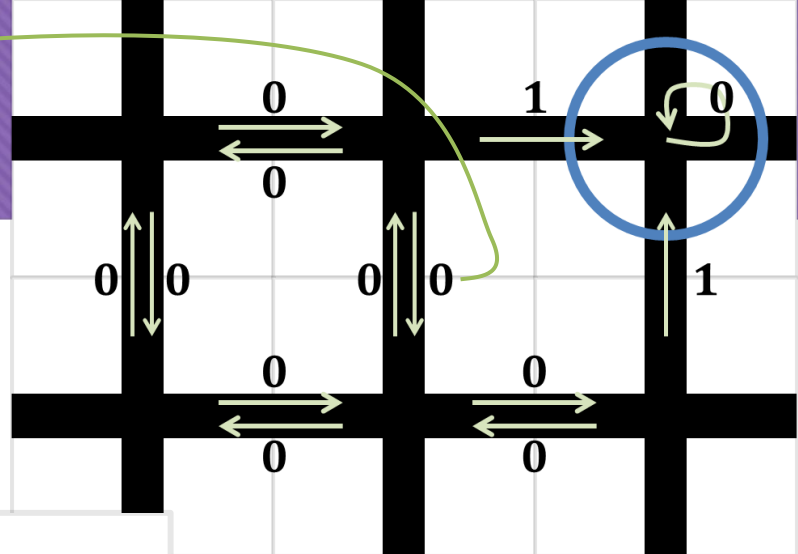
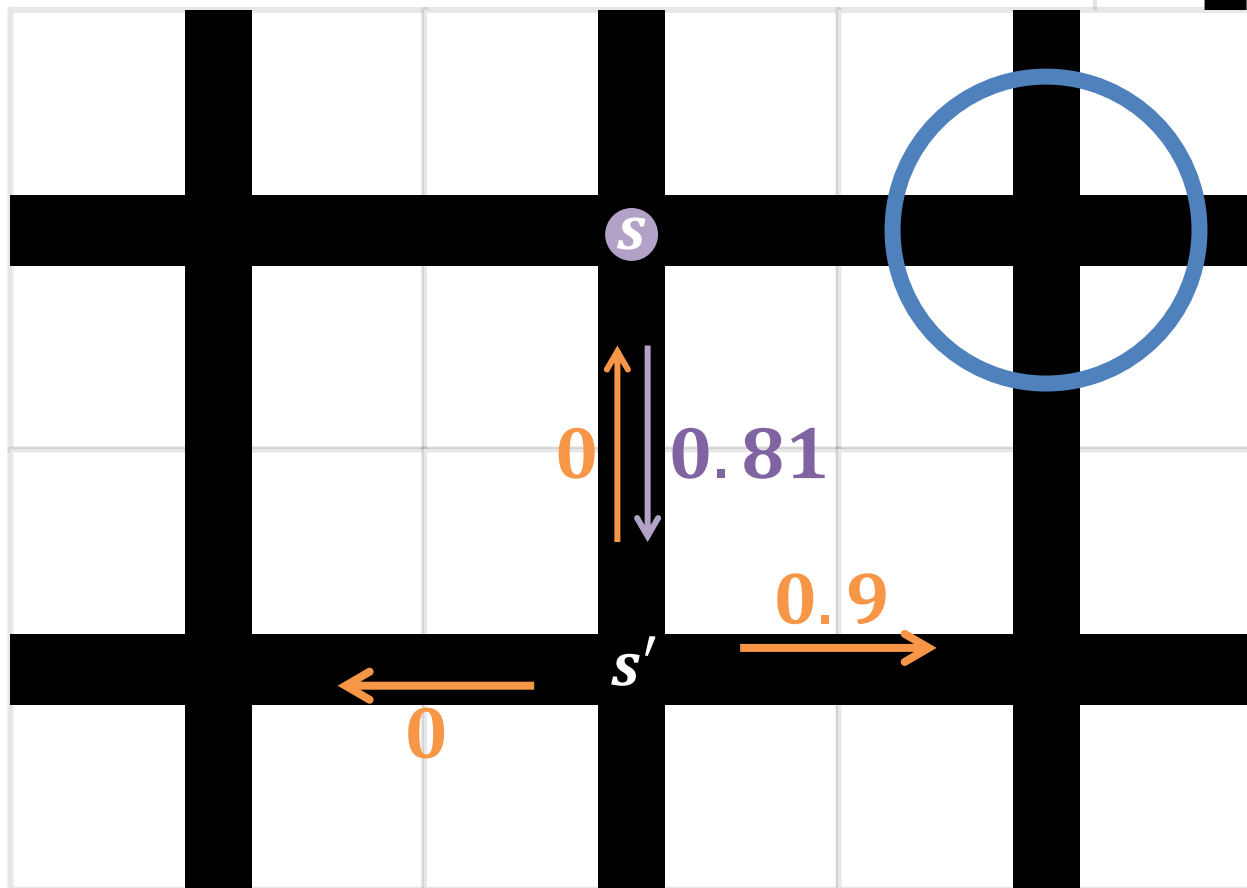
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

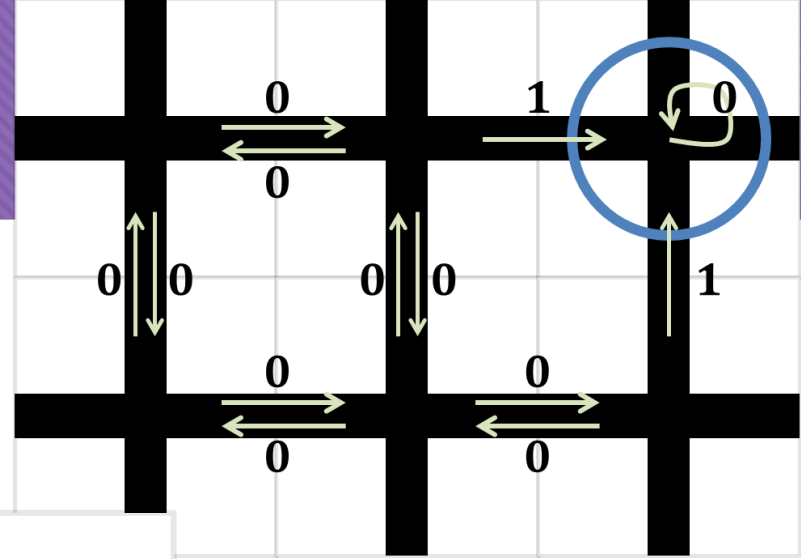
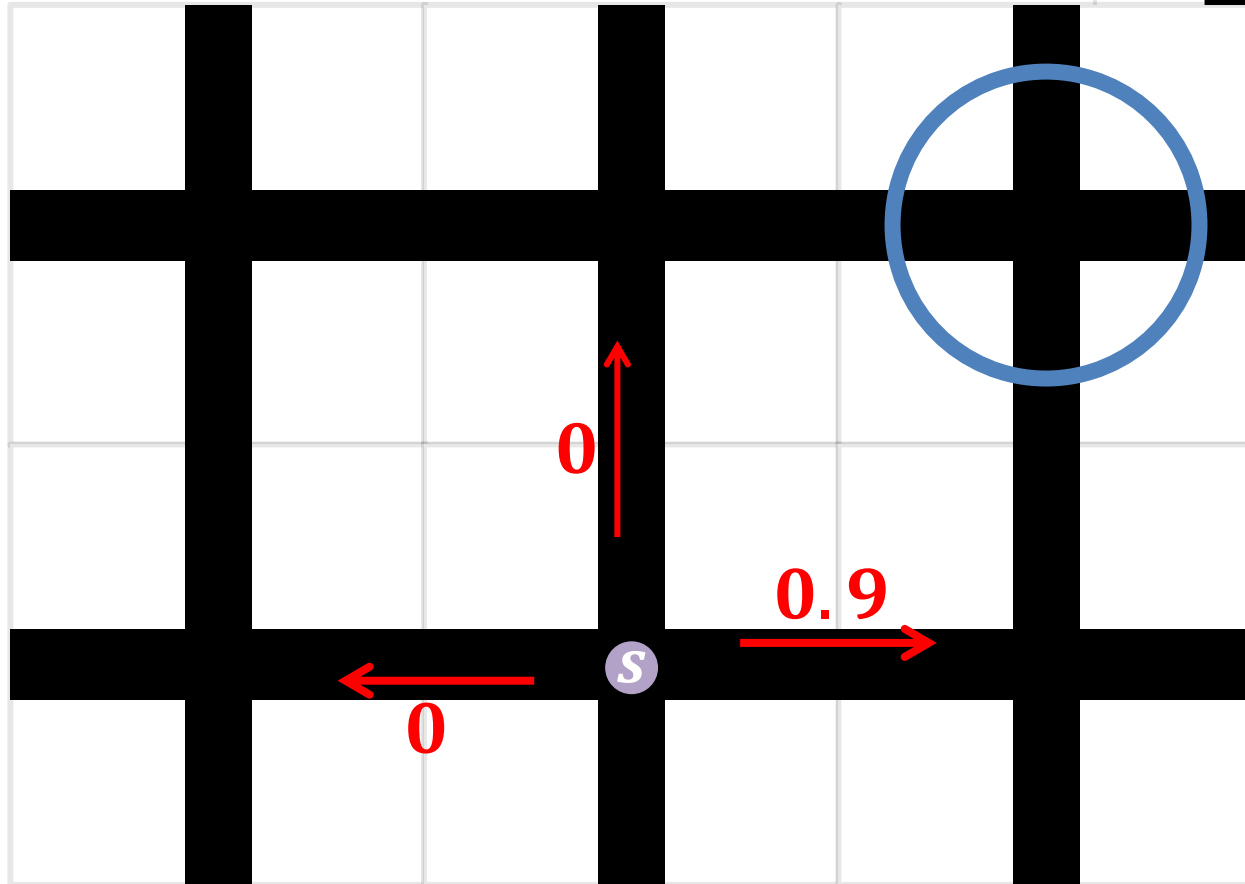
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

0 0.9



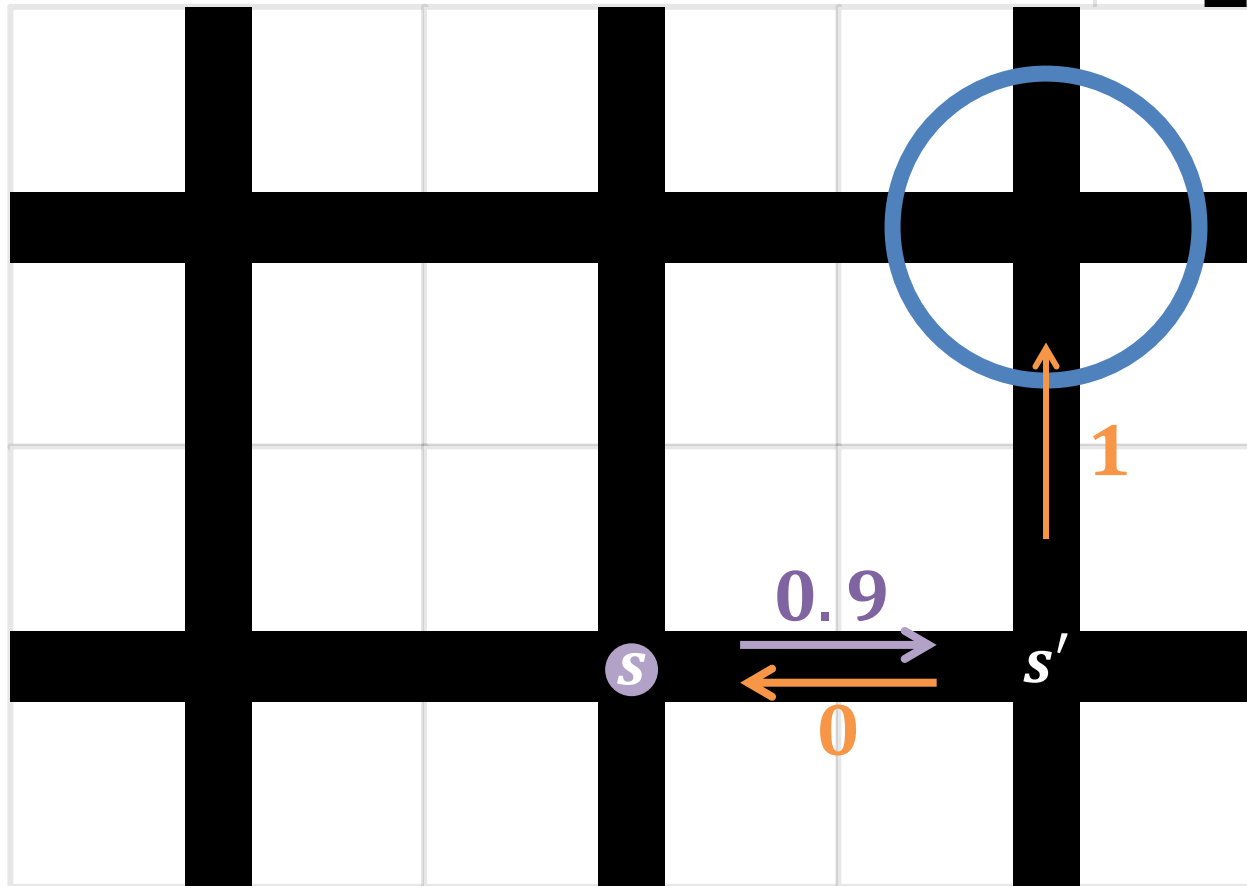
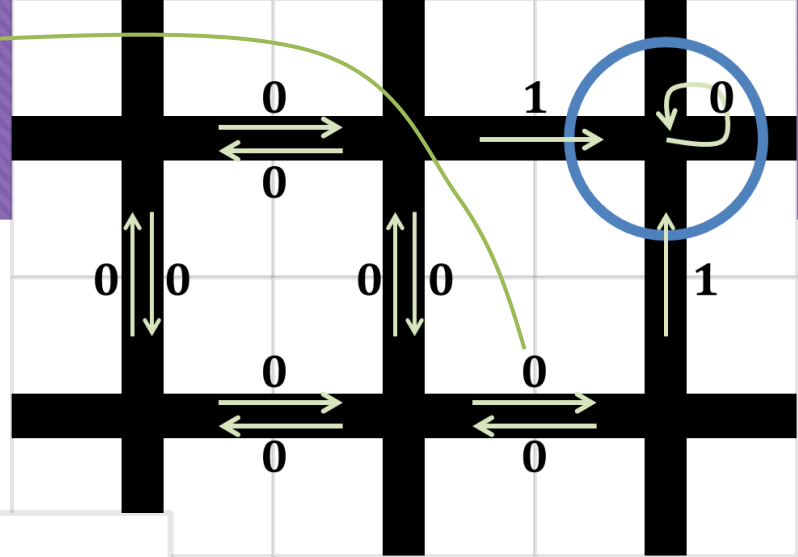
Q-러닝

$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



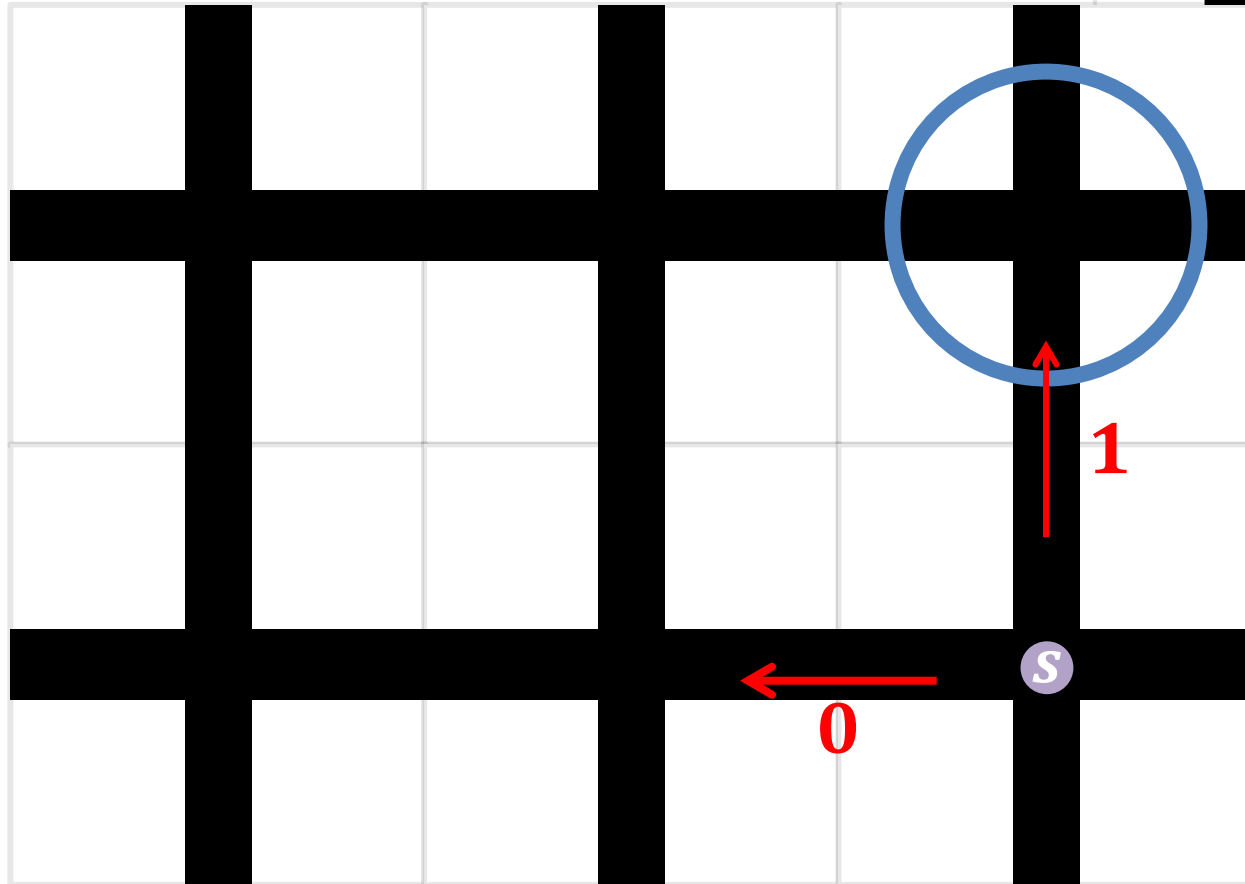
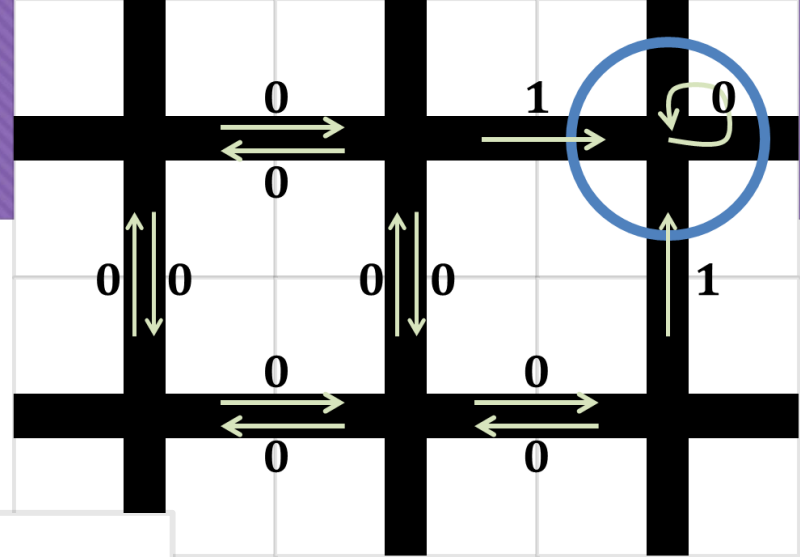
Q-러닝

$$\hat{Q}(s, a) \leftarrow \underset{0}{r} + \underset{1}{0.9} \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

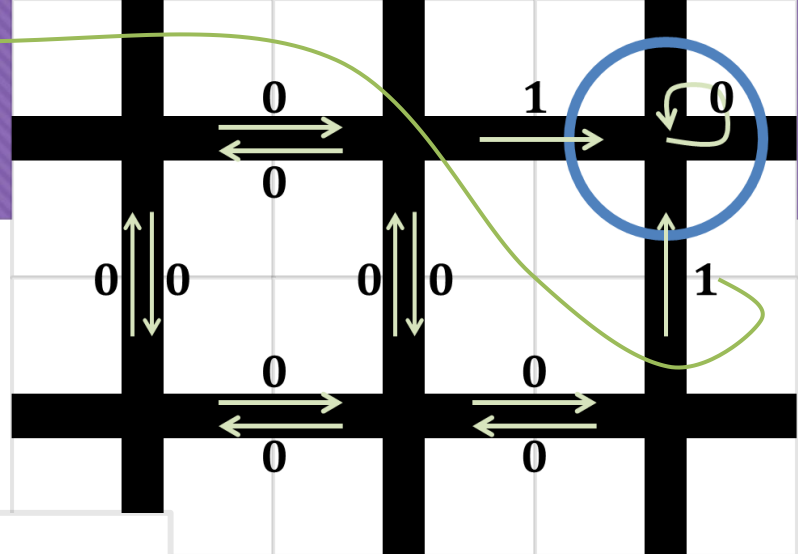
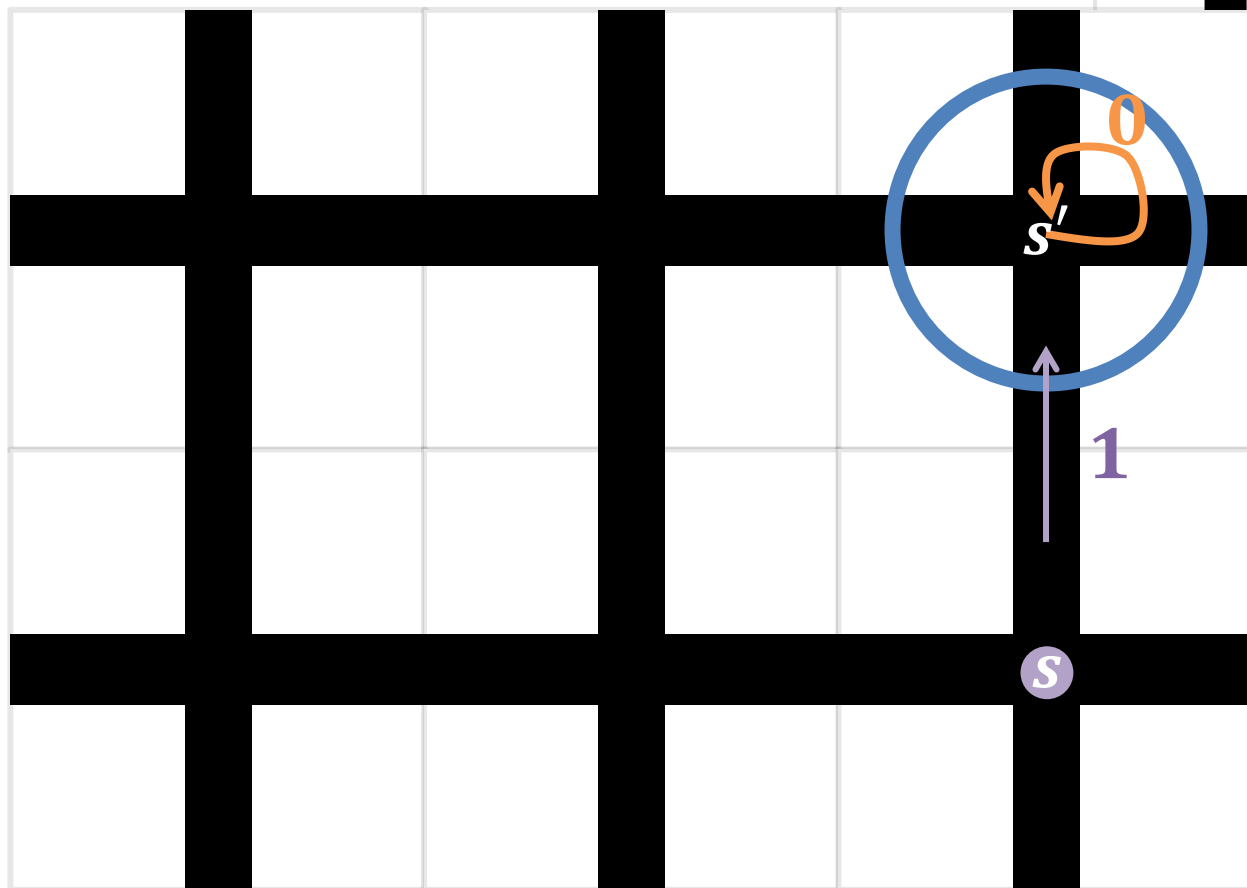
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$



Q-러닝

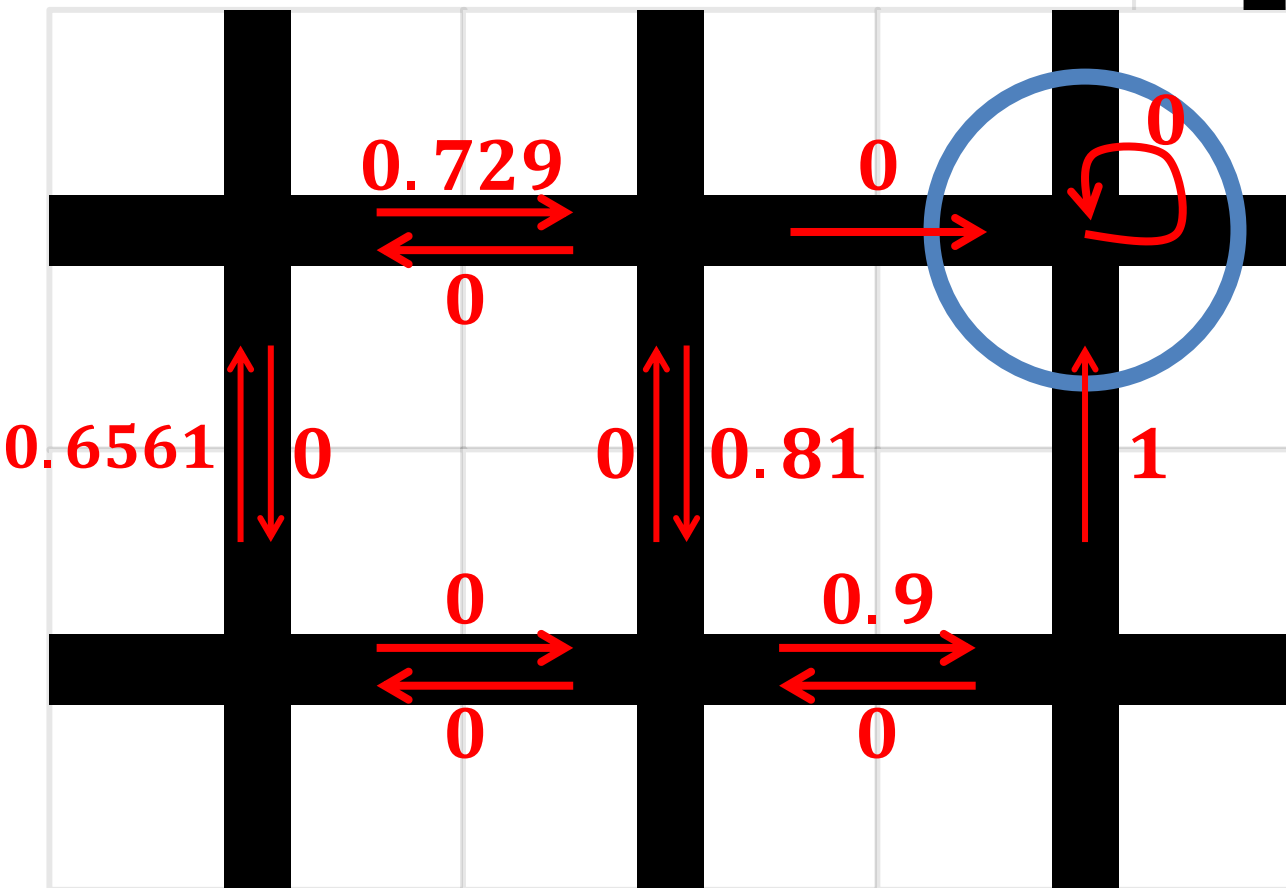
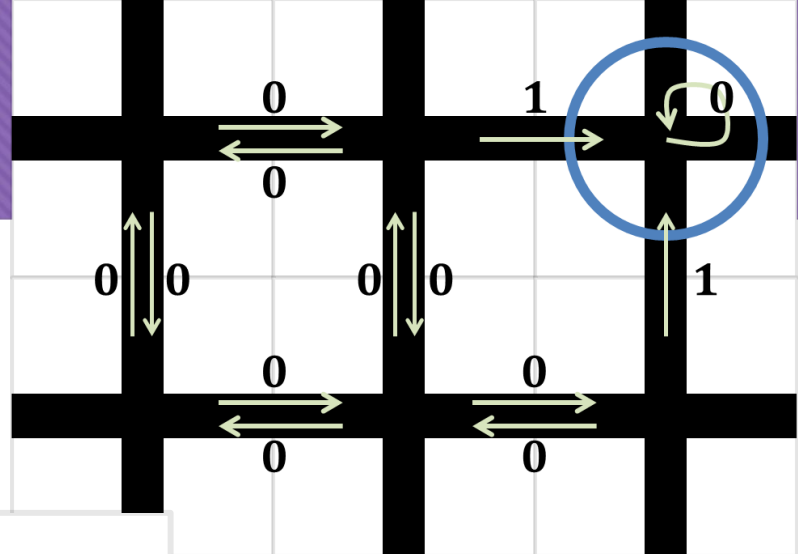
$$\hat{Q}(s, a) \leftarrow r + 0.9 \times \max_{a'} \hat{Q}(s', a')$$

1 0



Q-러닝

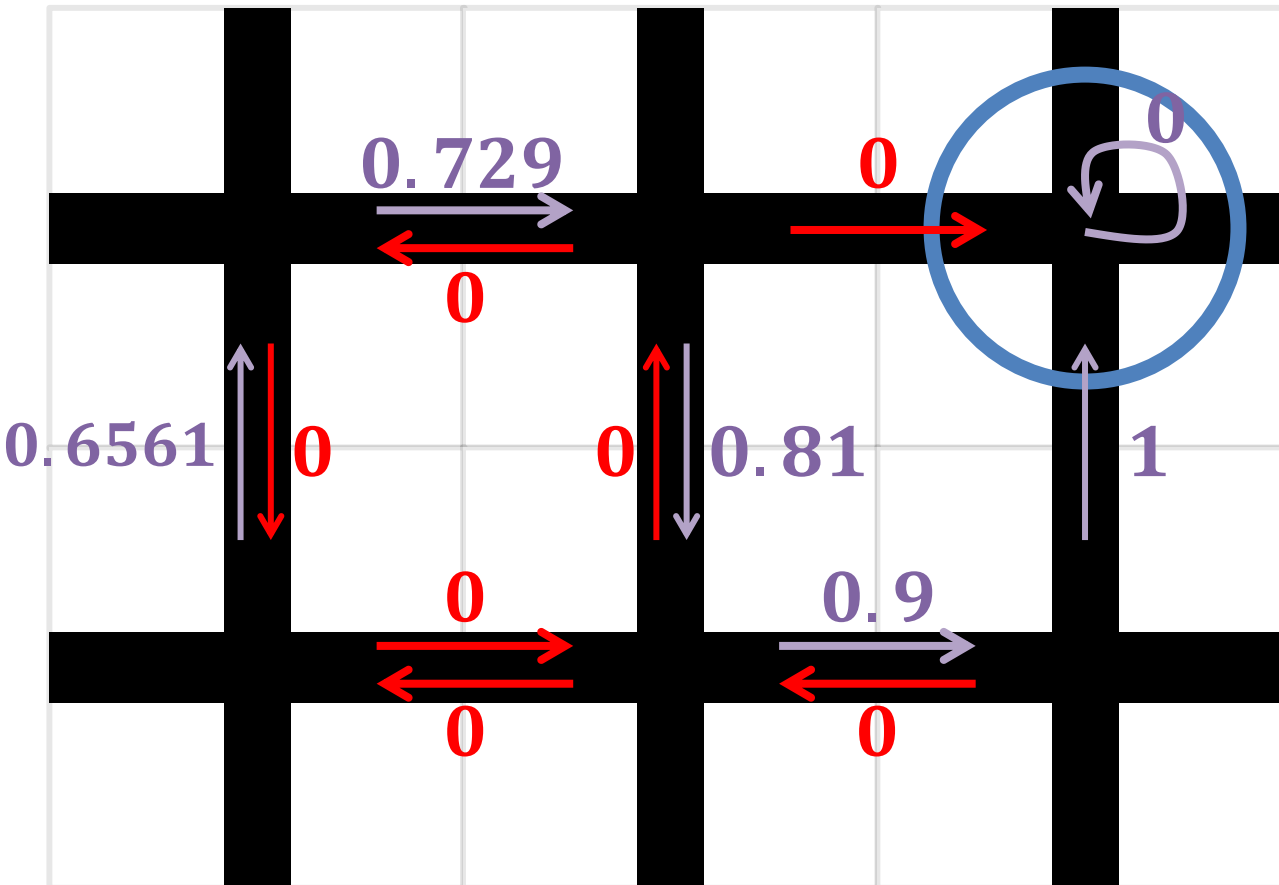
변화가 없을 때까지 계속 반복



최적의 정책(policy) $\pi^*(s)$

각 상태에서 Q값이 가장 큰 행동

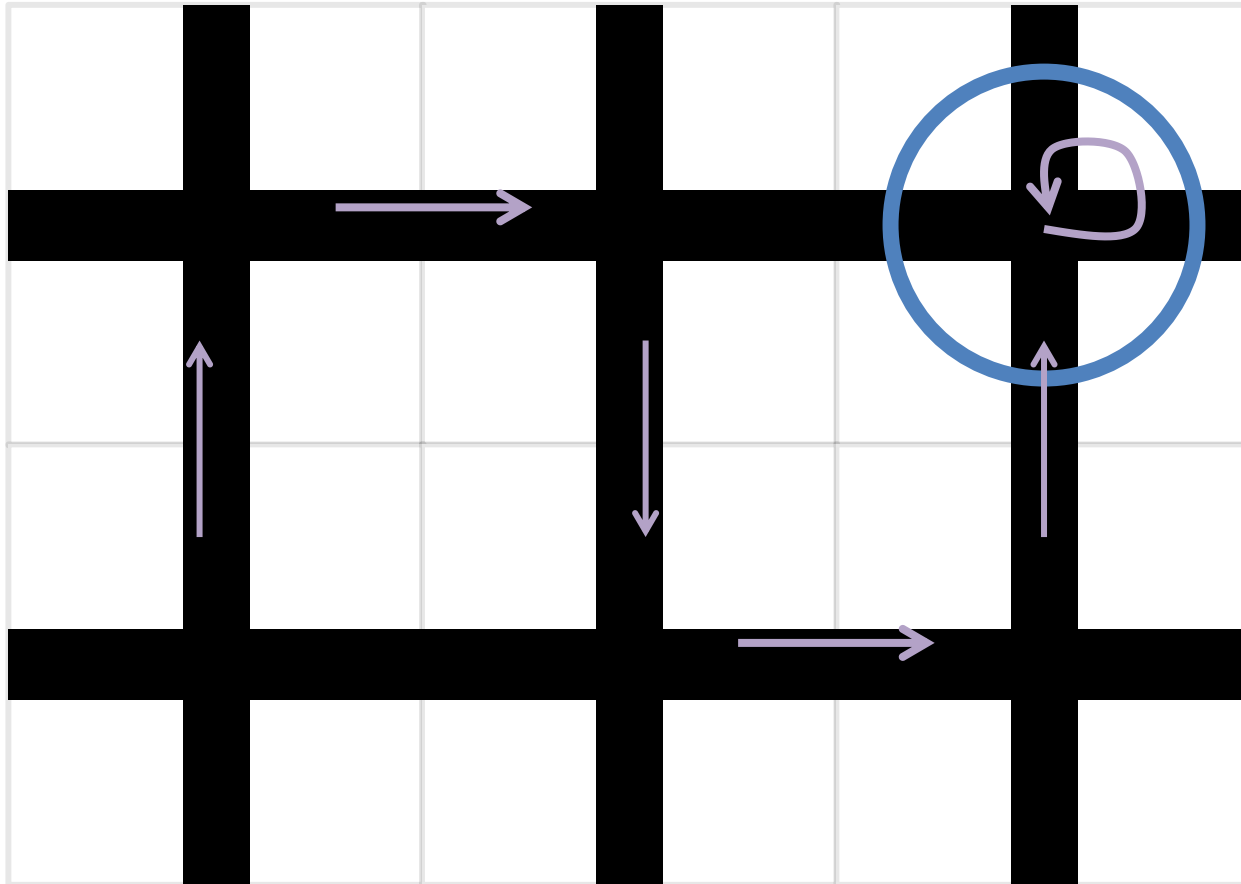
$$\pi^*(s) = \underset{a}{\operatorname{argmax}} Q(s, a)$$



최적의 정책(policy) $\pi^*(s)$

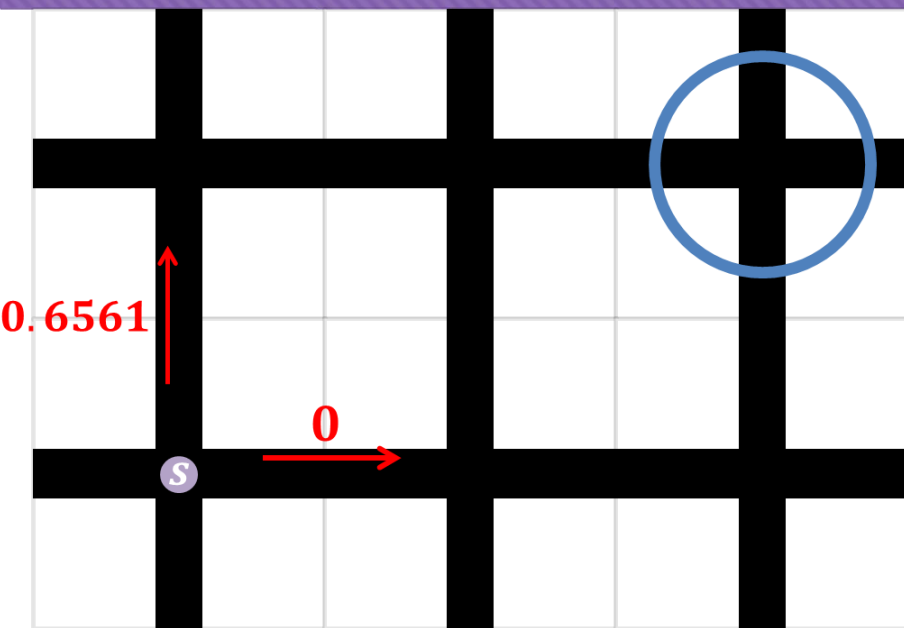
각 상태에서 Q값이 가장 큰 행동

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} Q(s, a)$$



탐험(exploration)과 착취(exploitation)

189



$$p(a_i|s) = \frac{\kappa^{\hat{Q}(s,a_i)}}{\sum_j \kappa^{\hat{Q}(s,a_j)}} \quad \kappa > 0$$

$$\kappa = 0.1 \quad \kappa^{\hat{Q}(s,a_{up})} = 0.1^{0.6561} = 0.22$$

$$\kappa^{\hat{Q}(s,a_{right})} = 0.1^0 = 1$$

$$\kappa = 10 \quad \kappa^{\hat{Q}(s,a_{up})} = 10^{0.6561} = 4.53$$

$$\kappa^{\hat{Q}(s,a_{right})} = 10^0 = 1$$

$$p(a_{up}|s) = \frac{0.22}{0.22 + 1} = 0.18$$

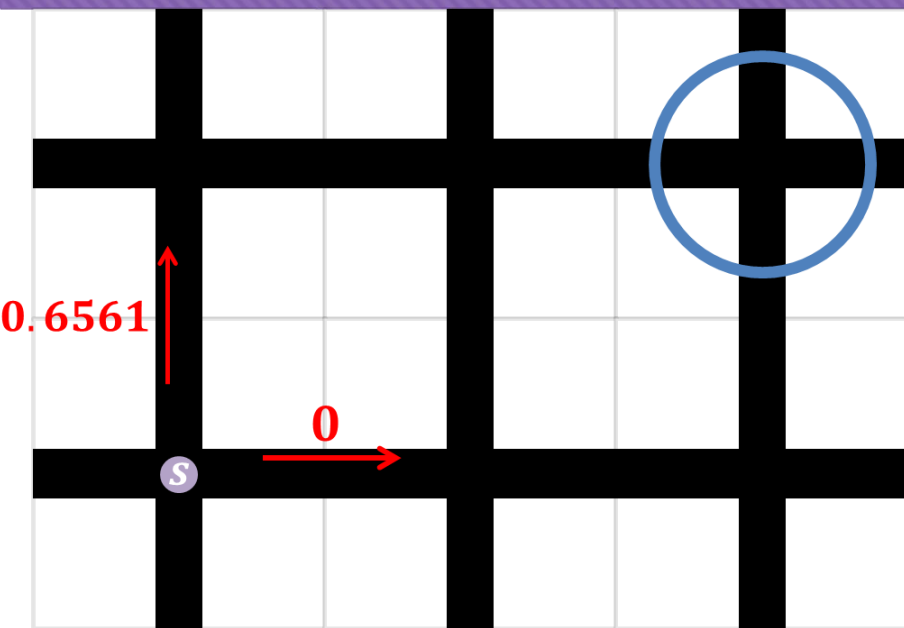
$$p(a_{right}|s) = \frac{1}{0.22 + 1} = 0.82$$

$$p(a_{up}|s) = \frac{4.53}{4.53 + 1} = 0.82$$

$$p(a_{right}|s) = \frac{1}{4.53 + 1} = 0.18$$

탐험(exploration)과 착취(exploitation)

190



랜덤 노이즈:

$$\hat{Q}(s, a) + \text{노이즈}$$

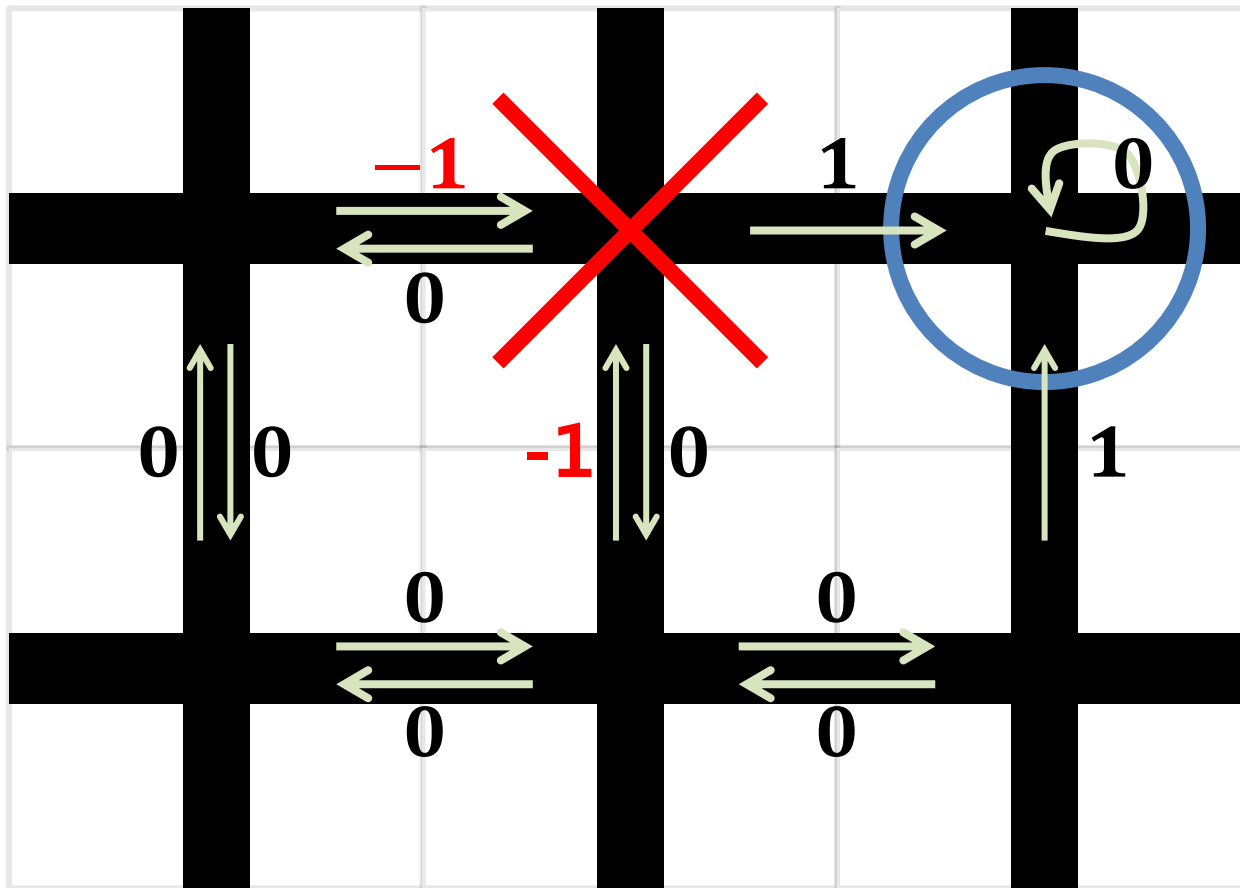
ϵ -그리디(greedy):

ϵ 확률로 탐험

$(1 - \epsilon)$ 확률로 착취(Q값에 따라 행동)

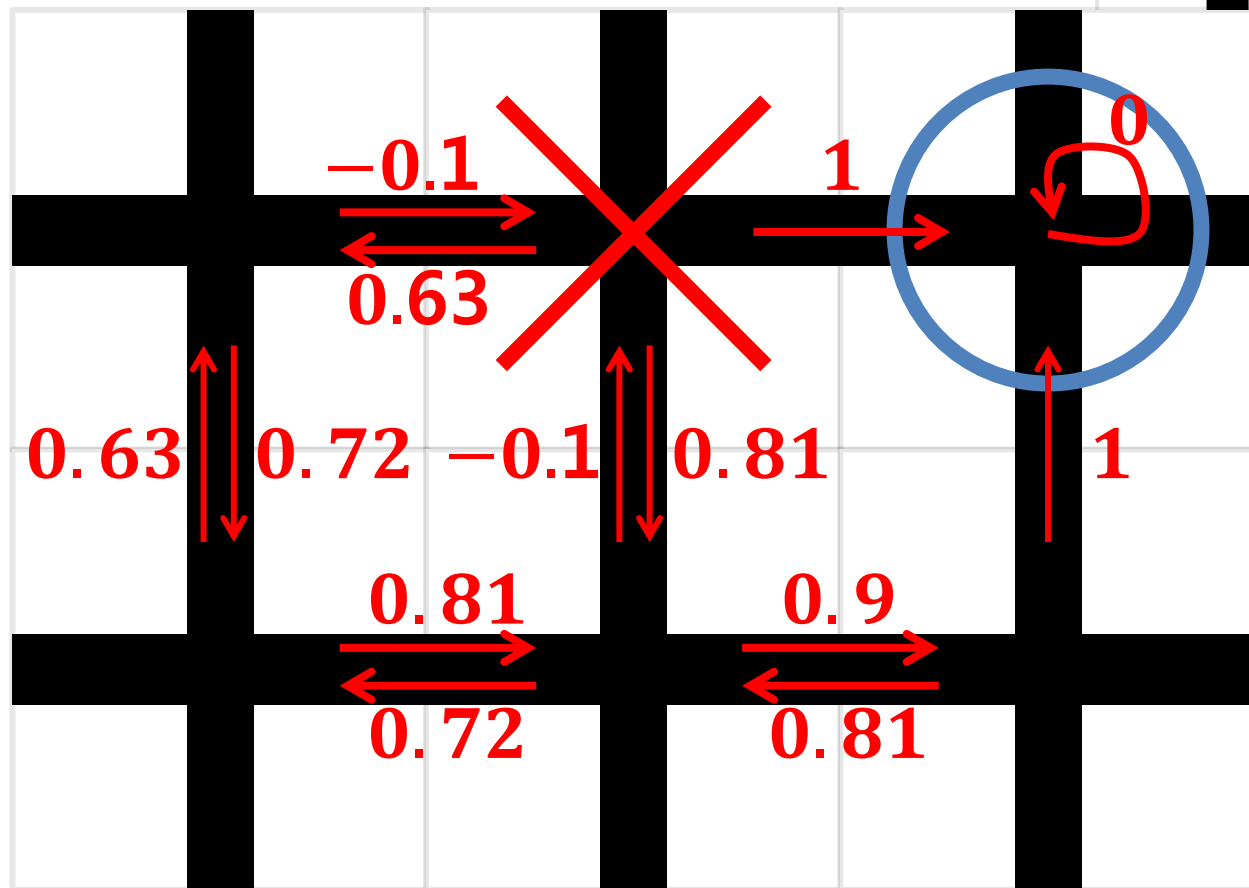
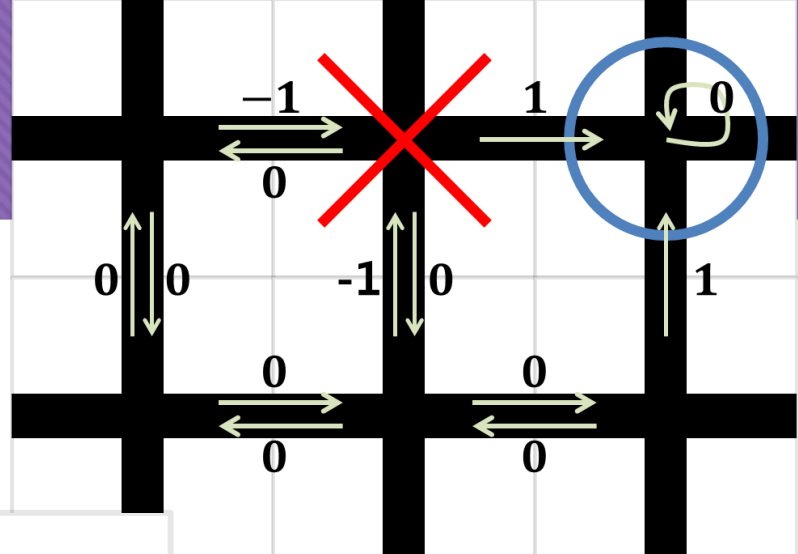
감소하는 ϵ -그리디(decaying ϵ -greedy):

ϵ 값을 점점 줄여 가며 ϵ -그리디



합정 추가

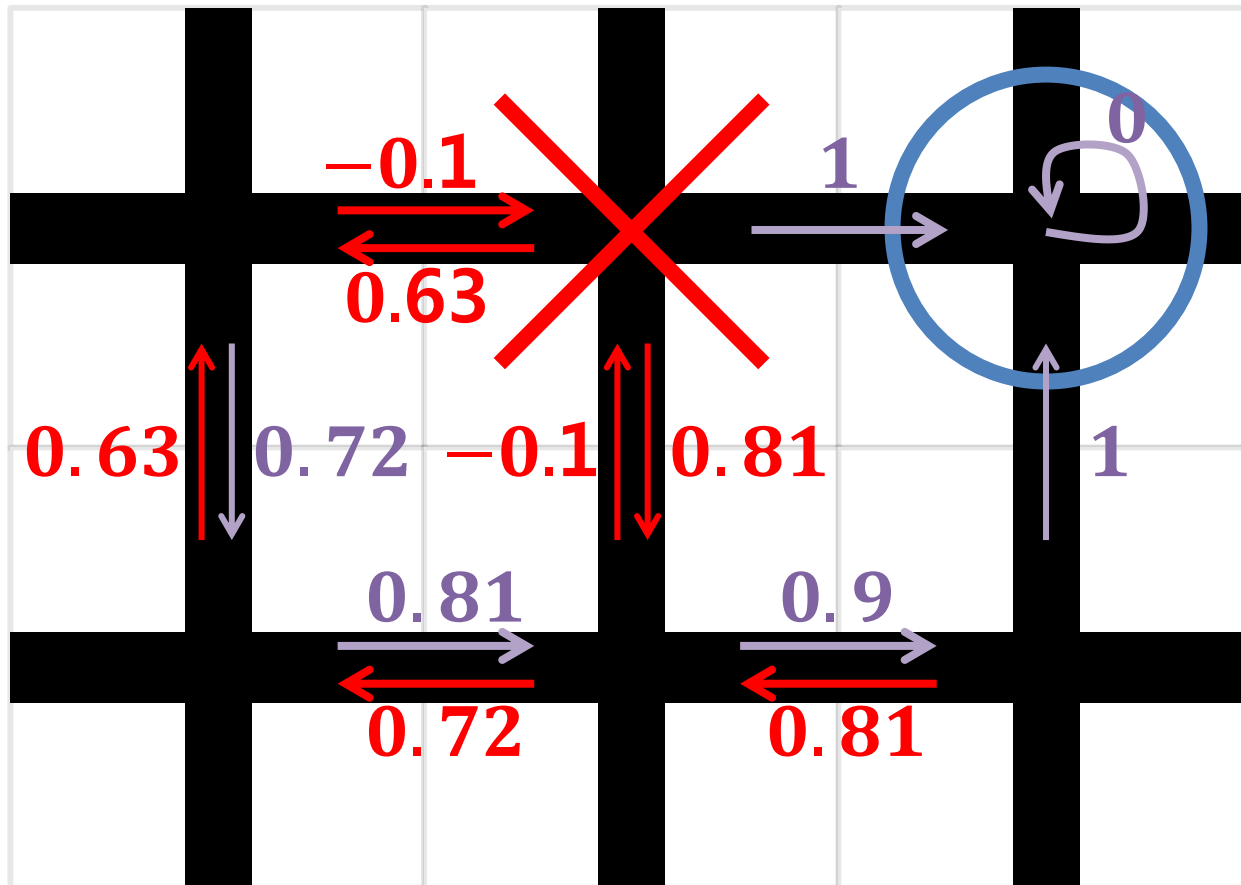
$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$



최적의 정책(policy) $\pi^*(s)$

각 상태에서 Q값이 가장 큰 행동

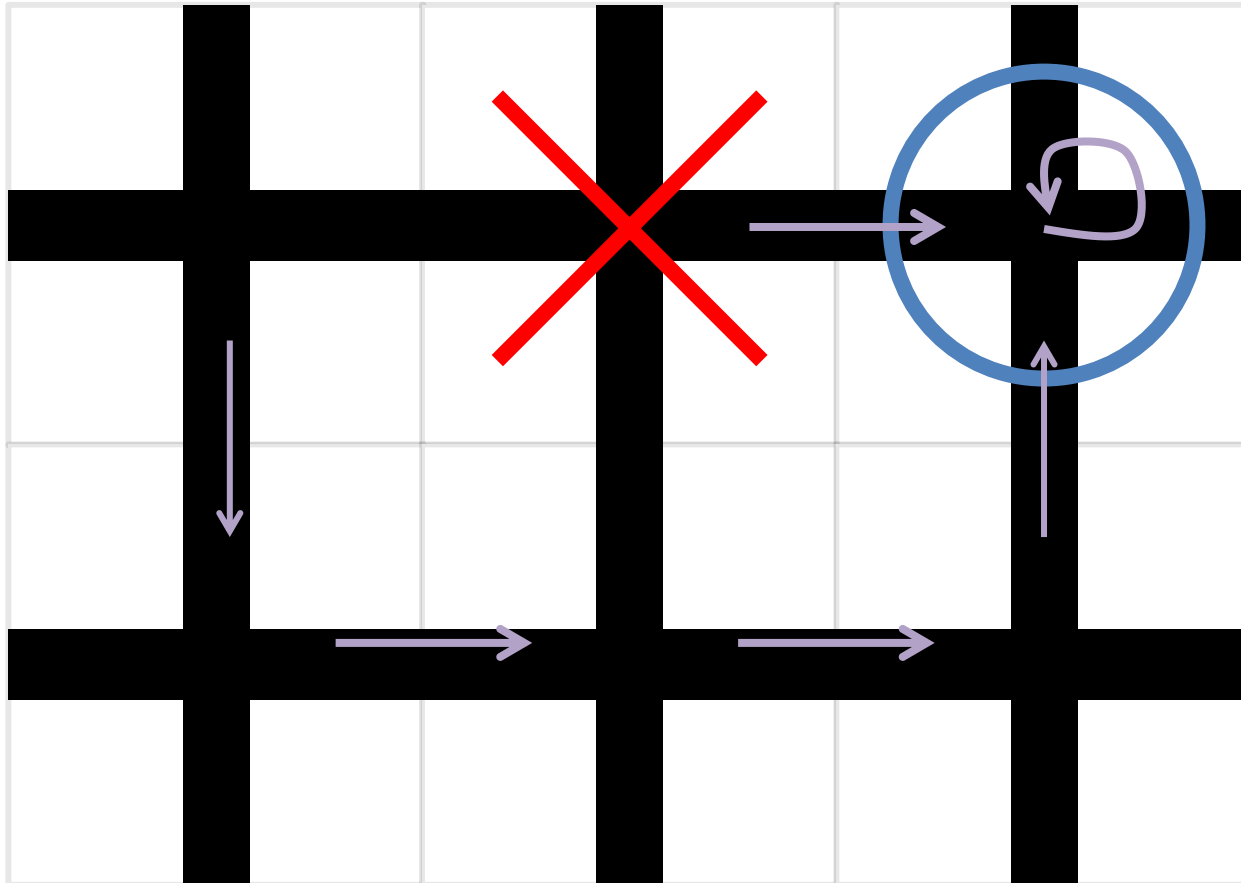
$$\pi^*(s) = \underset{a}{\operatorname{argmax}} Q(s, a)$$



최적의 정책(policy) $\pi^*(s)$

각 상태에서 Q값이 가장 큰 행동

$$\pi^*(s) = \operatorname{argmax}_a Q(s, a)$$



엔트리로 구현하기

<http://naver.me/FoAixwi6>

The screenshot displays the Playentry web editor interface. At the top, a browser tab is open with the URL `playentry.org/ws?lang=ko#!/`. A red arrow points from the URL `http://naver.me/FoAixwi6` to this tab. The editor's header bar is blue and contains the text "200709_작품" and a "장면 1" (Scene 1) button. The main workspace shows a character named "엔트리봇" (EntriBot) with a blue body and a yellow head, positioned at coordinates X: -234.5, Y: -127.7. The left sidebar contains a list of tools: 시각 (Visual), 효율 (Efficiency), 움직임 (Movement), 생김새 (Appearance), 붓 (Brush), 소리 (Sound), 판단 (Judgment), 개산 (Calculation), 자료 (Data), 함수 (Function), 데이터분석 (Data Analysis), 인공지능 (Artificial Intelligence), 확장 (Extension), and 하드웨어 (Hardware). The right sidebar shows a list of events and actions for the character, including "시작하기 버튼을 클릭했을 때" (When the start button is clicked), "카를 눌렀을 때" (When the 'K' key is pressed), "마우스를 클릭했을 때" (When the mouse is clicked), "마우스 클릭을 해제했을 때" (When the mouse click is released), "오브젝트를 클릭했을 때" (When the object is clicked), "오브젝트 클릭을 해제했을 때" (When the object click is released), "대상 없음" (No target), "신호를 받았을 때" (When the signal is received), "신호 보내기" (Send signal), "신호 보내고 기다리기" (Send signal and wait), "장면이 시작되었을 때" (When the scene starts), "장면 1 시작하기" (Start scene 1), and "다들" (Everyone) "장면 시작하기" (Start scene).

말판: 가로 4칸, 세로 4칸

각 칸의 행동: 4가지

- 왼쪽으로 이동하기
- 오른쪽으로 이동하기
- 위로 이동하기
- 아래로 이동하기

➔ 4 x 4 x 4 만큼의 저장 공간 필요







Q값을 저장할 공간의 인덱스




198

가로 방향 칸 번호 (로봇X)	세로 방향 칸 번호 (로봇Y)	행동 번호 (0: 위, 1: 오른쪽, 2: 아래, 3: 왼쪽)	리스트의 인덱스 (큐 번호)
1	1	0	21
1	1	1	22
1	1	2	23
1	1	3	24
1	2	0	25
1	2	1	26
1	2	2	27
1	2	3	28
1	3	0	29
1	3	1	30
⋮	⋮	⋮	⋮

$$[\text{큐 번호}] = 1 + [\text{로봇X}] \times 16 + [(\text{로봇Y}) \times 4 + [\text{행동}]]$$

변수 이름	설명
로봇X	말판 위에서 로봇의 현재 위치의 x좌표 (1, 2, 3, 4)
로봇Y	말판 위에서 로봇의 현재 위치의 y좌표 (1, 2, 3, 4)
로봇 방향	말판 위에서 로봇의 현재 방향 (왼쪽, 오른쪽, 위쪽, 아래쪽)
최적의 행동	교차로의 네 방향에 서 있는 Q들에게 물어 보고 Q값이 가장 큰 행동을 선택한다고 하였는데, 현재 교차로(이동하기 전의 교차로)에서 유효한 행동 중 Q값이 가장 큰 행동
o키를 눌렀는가	키보드의 알파벳 o키를 눌렀으면 "참", 아니면 "거짓"
x키를 눌렀는가	키보드의 알파벳 x키를 눌렀으면 "참", 아니면 "거짓"
보상	현재 교차로에서 선택된 행동을 수행하였을 때의 보상 값
큐 최댓값	유효한 행동들에 대한 Q값의 최댓값
큐 번호	반복문 내에서 사용하는 임시 변수
수정할 큐 번호	현재 교차로(이동하기 전의 교차로)에서 선택된 행동에 대한 큐 리스트의 인덱스

함수	설명
초기화하기 	로봇의 위치를 출발 위치 (1, 1)로 설정하고, 로봇의 방향을 오른쪽으로 설정합니다.
왼쪽으로 이동하기 	말판 위에서 왼쪽으로 한 칸 이동합니다.
오른쪽으로 이동하기 	말판 위에서 오른쪽으로 한 칸 이동합니다.
위로 이동하기 	말판 위에서 위로 한 칸 이동합니다.
아래로 이동하기 	말판 위에서 아래로 한 칸 이동합니다.
최적의 행동 계산하기 	교차로의 네 방향에 서 있는 Q들에게 물어 보고 Q값이 가장 큰 행동을 선택한다고 하였는데, 현재 교차로(이동하기 전의 교차로)에서 유효한 행동 중 Q값이 가장 큰 행동을 찾아서 변수 "최적의 행동"에 대입합니다. 즉, 이 함수를 호출하면 변수 "최적의 행동"에 "왼쪽", "오른쪽", "위쪽", "아래쪽" 중 하나가 대입됩니다.

함수	설명
결과 표현하기 	키보드의 알파벳 o키를 눌렀을 때 또는 x키를 눌렀을 때 각각에 대한 행동을 소리와 LED로 표현하고 출발 위치로 이동합니다.
키보드 입력 확인하기 	키보드의 스페이스 키, 알파벳 o키, 알파벳 x키를 누를 때 까지 기다립니다.
큐 번호 계산하기 	로봇의 현재 위치(로봇X, 로봇Y)에 대한 큐 리스트의 인덱스를 계산하여 변수 "큐 번호"에 대입합니다. 즉 (큐 번호) = $1 + (\text{로봇X}) \times 16 + (\text{로봇Y}) \times 4$ 를 계산합니다.

시작하기 버튼을 클릭했을 때

초기화하기

테이블 명령 수 의 차트 창 열기

계속 반복하기

최적의 행동 계산하기

만일 최적의 행동 값 = 왼쪽 (이)라면

왼쪽으로 이동하기

아니면

만일 최적의 행동 값 = 오른쪽 (이)라면

오른쪽으로 이동하기

아니면

만일 최적의 행동 값 = 위쪽 (이)라면

위로 이동하기

아니면

아래로 이동하기

키보드 입력 확인하기

만일 o키를 눌렀는가 참 (이)라면

시작하기 버튼을 클릭했을 때

초기화하기

테이블 명령 수 의 차트 창 열기

계속 반복하기

최적의 행동 계산하기

만일 <최적의 행동 > 값 = 왼쪽 (이)라면

왼쪽으로 이동하기

아니면

만일 <최적의 행동 > 값 = 오른쪽 (이)라면

오른쪽으로 이동하기

아니면

만일 <최적의 행동 > 값 = 위쪽 (이)라면

위로 이동하기

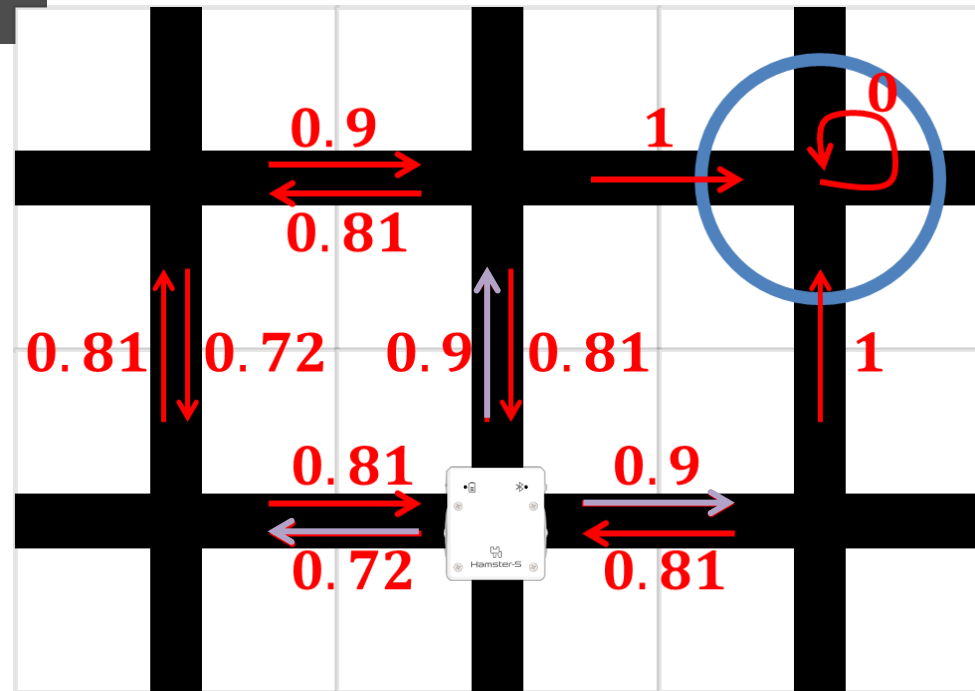
아니면

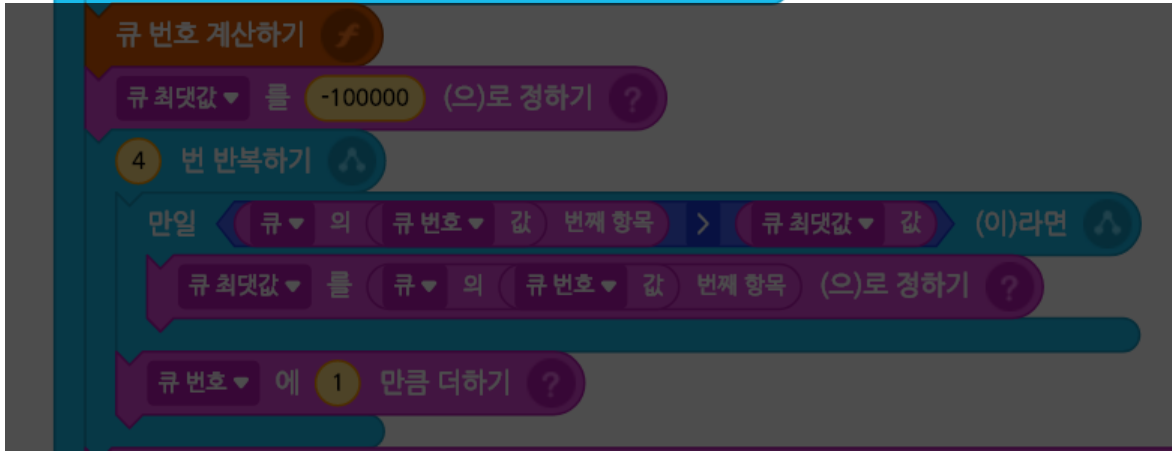
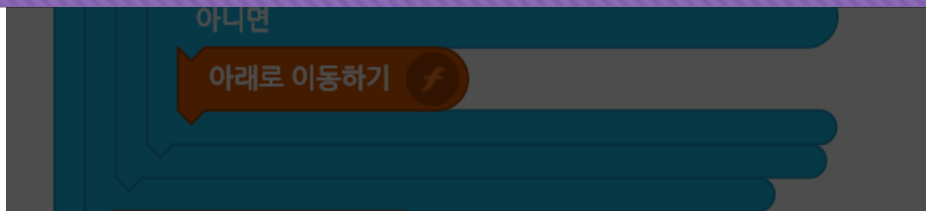
아래로 이동하기

키보드 입력 확인하기

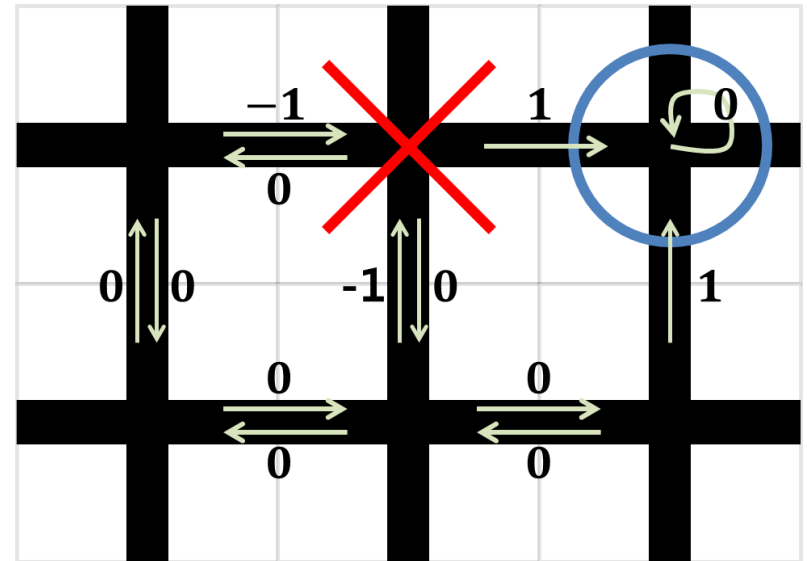
만일 <키를 눌렀는가 > 값 = 참 (이)라면

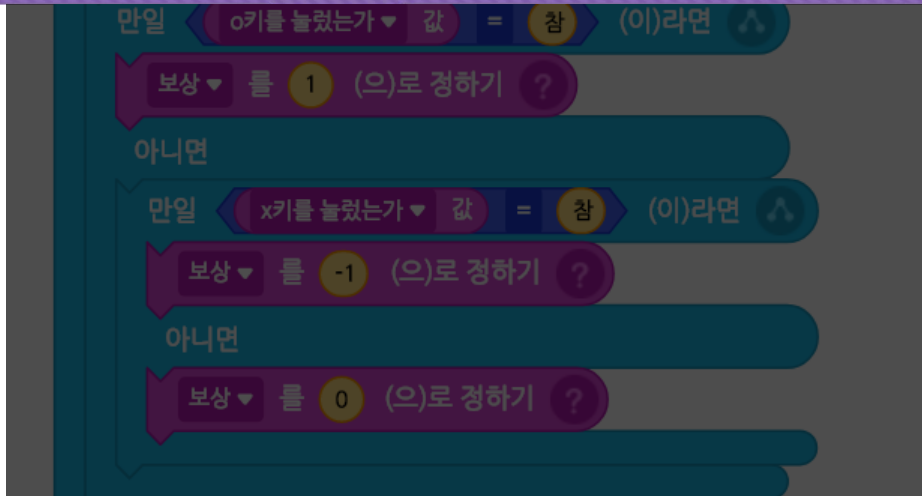
$$\hat{Q}(s, a)$$





보상 r





→ 다음 교차로(이동한 후)의 큐 번호
 $[큐\ 번호] = 1 + [로봇X] \times 16 + [(로봇Y) \times 4 + [행동]$

[현재 교차로에서 Q_a 값] \leftarrow [점수 r] + 0.9 x [다음 교차로에서 Q_a 값들 중 최댓값]

심화

ϵ -그리디(greedy)

207



보상을 다르게 주면 어떻게 될까?

208

변수 추가



어느 날 로봇이 도착해야 하는 목표 지점이 제일 오른쪽에 있다는 정보를 입수하였습니다.

제일 오른쪽에서 정확히 어느 곳에 있는지는 알 수 없지만 오른쪽으로 이동하는 것이 유리할 것 같습니다.

로봇이 오른쪽으로 이동하는 것을 선호하도록 보상을 주는 방법을 변경하고 다시 게임을 수행하여 로봇의 행동에 어떤 변화가 있는지 서술해 봅시다.

경우에 따라서는 기대했던 대로 로봇이 행동하지 않는 경우도 있는데 게임을 처음부터 다시 여러 번 해보면서 전체적으로 어떻게 행동하는지 관찰해 봅시다.

감사합니다