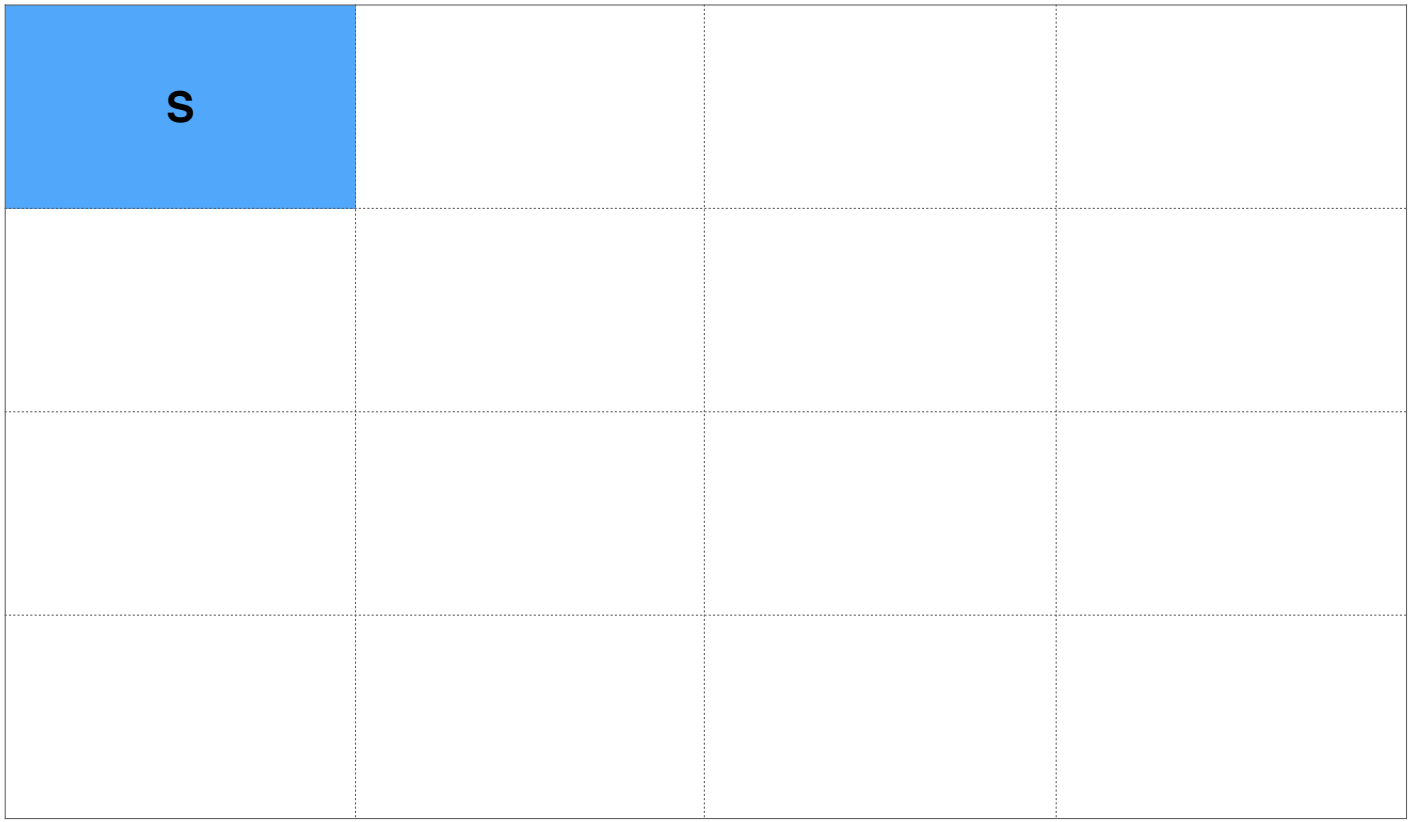
Lecture 3: Q-learning (table)

Reinforcement Learning with TensorFlow&OpenAI Gym

Sung Kim <hunkim+ml@gmail.com>

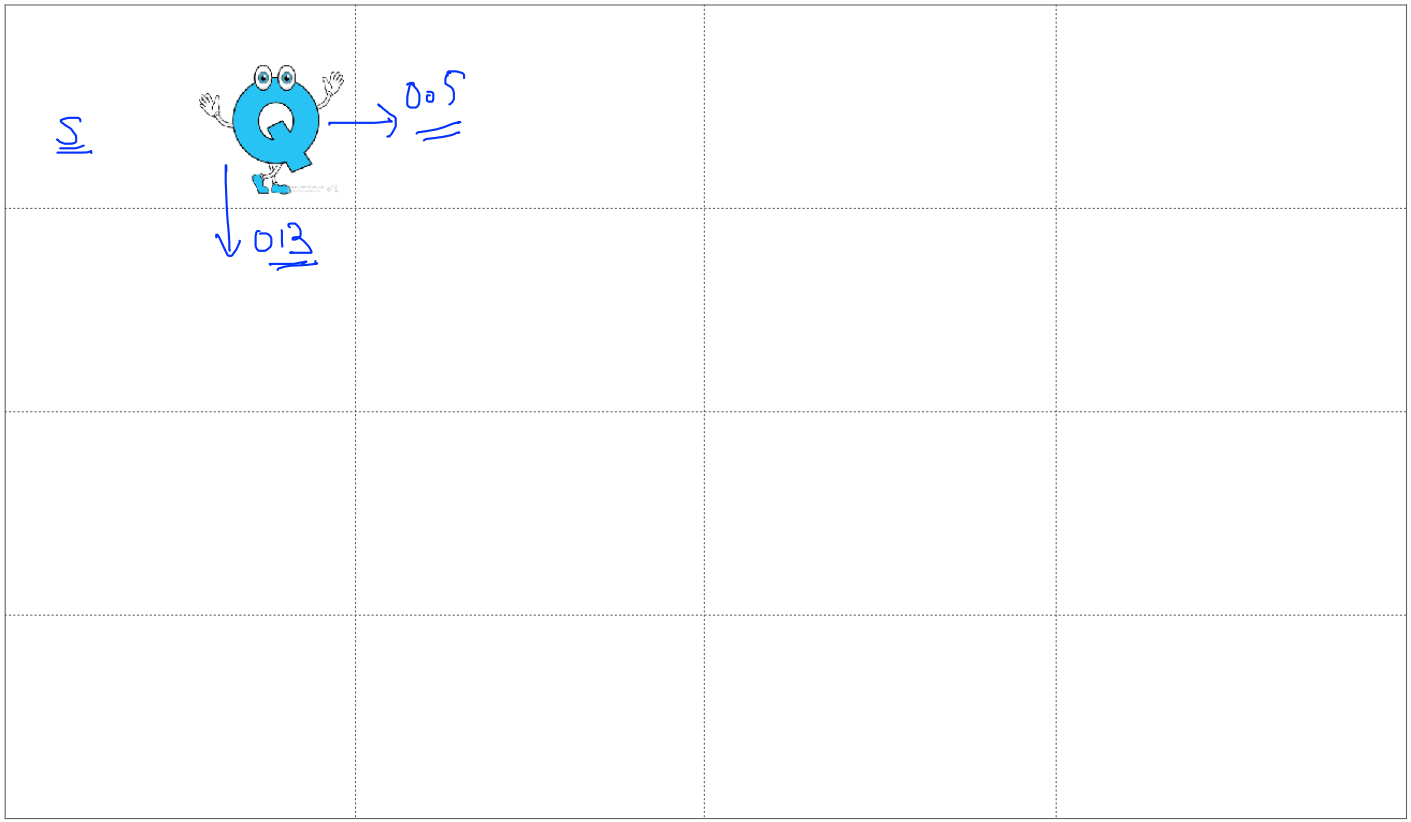
Try Frozen Lake, Real Game?

Frozen Lake: Random?

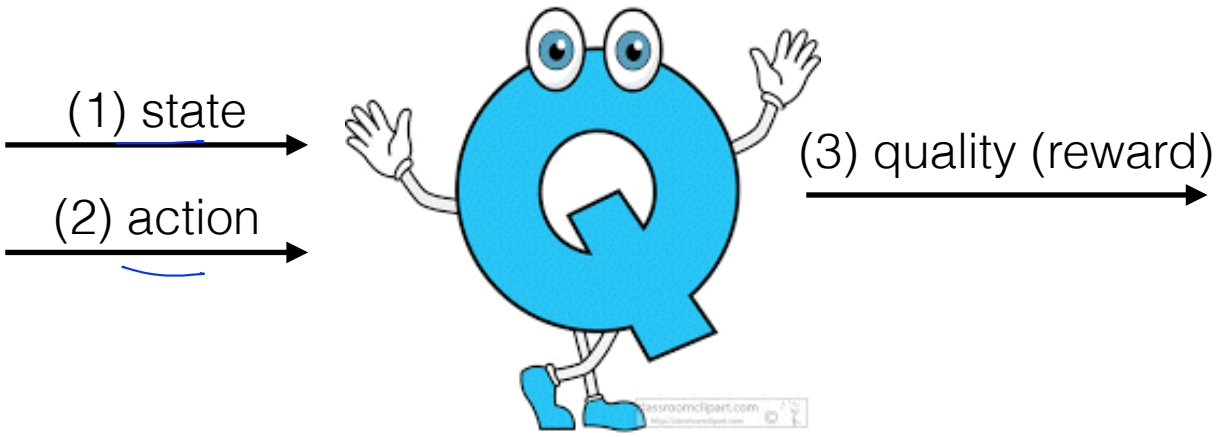


Frozen Lake: Even if you know the way, **ask**.

"아는 길도, **물어가라**"

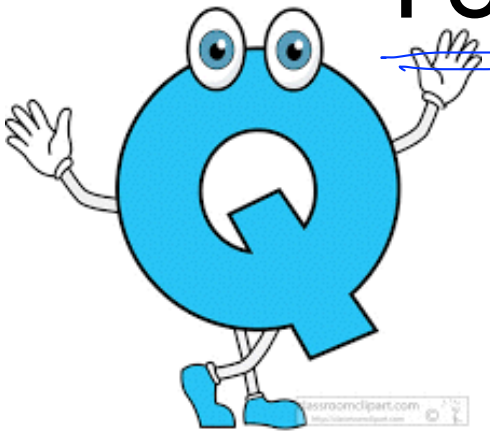


Q-function (state-action value function)



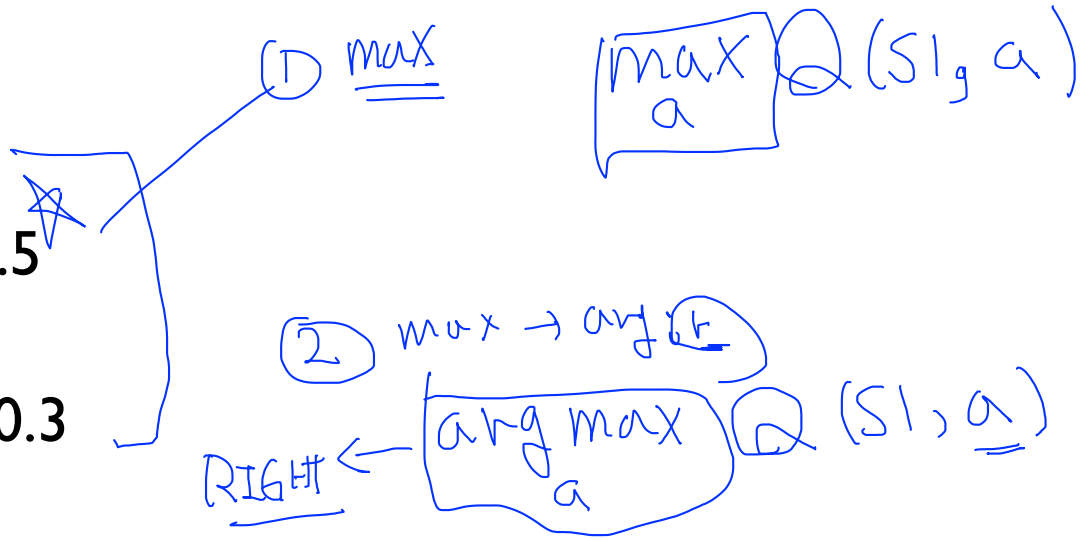
Q (state, action)

Policy using Q-function



Q (state, action)

Q (s1, LEFT): 0
Q (s1, RIGHT): 0.5
Q (s1, UP): 0
Q (s1, DOWN): 0.3



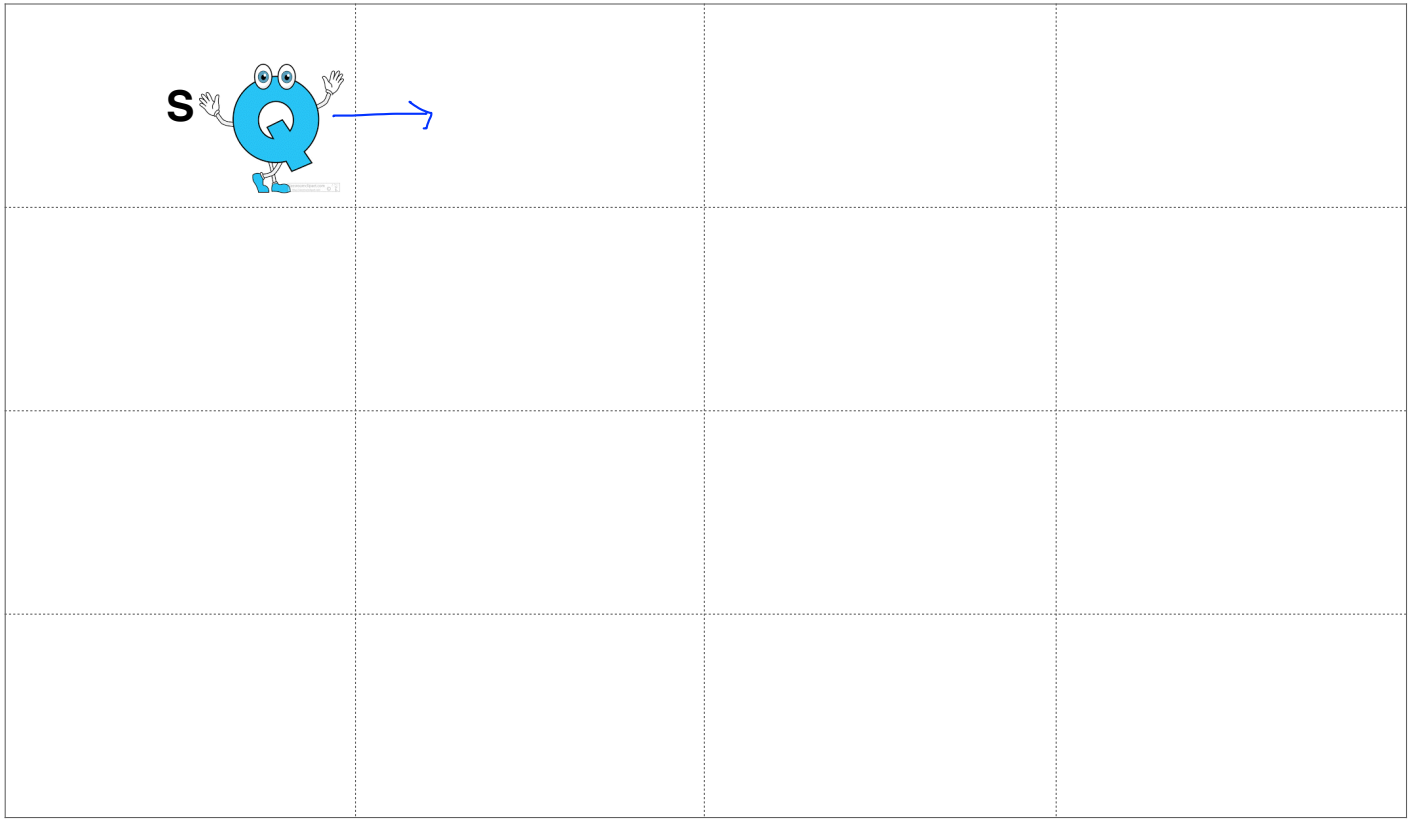
Optimal Policy, π and Max Q

Q (state, action)

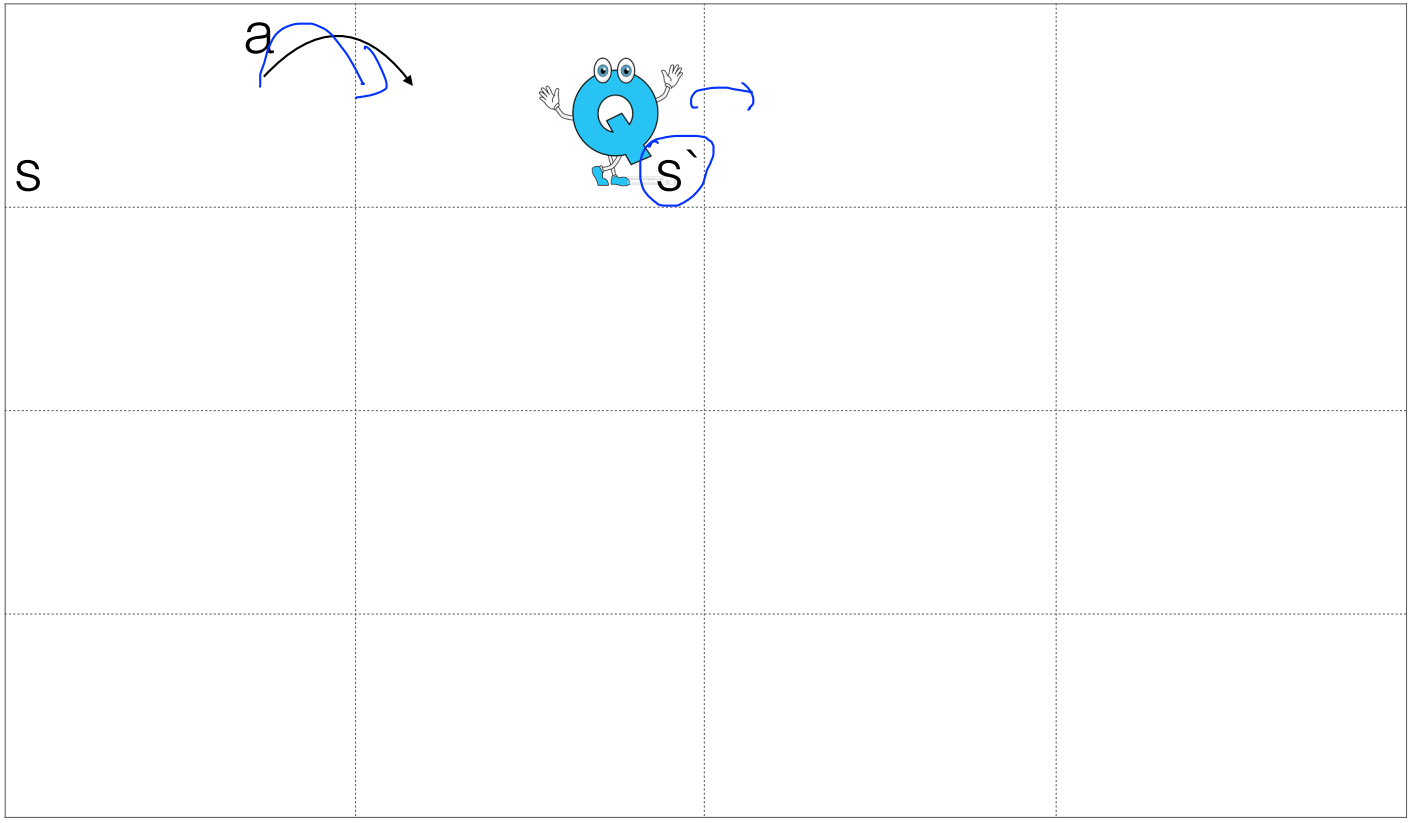
Max Q = $\max_{a'} Q(s, a')$

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

Frozen Lake: optimal policy with Q



Frozen Lake: optimal policy with Q

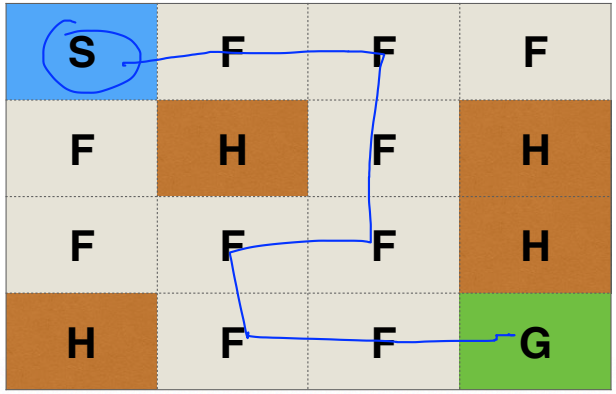


Finding, Learning Q

- Assume (believe) Q in s` exists!
- My condition
  - I am in **s**
  - when I do action **a**, I'll go to **s`**
  - when I do action **a**, I'll get reward **r**
  - Q in s`, Q(s`, a`) exist!
- How can we express Q(s, a) using Q(s`, a`)?

Learning Q (s, a)?

State, action, reward



$$s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{n-1}, a_{n-1}, r_n, s_n$$

state

action

reward

Terminal state

Future reward

$$s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{n-1}, a_{n-1}, r_n, s_n$$

state

action

reward

Terminal state

$$R = r_1 + r_2 + r_3 + \cdots + r_n$$

* optimal.

$$R_t = r_t + r_{t+1} + r_{t+2} + \cdots + r_n$$

$R(t) = r_t + R(t+1)$

$R^*(t) = r_t + \max R(t+1)$

$R(t+1) = r_{t+1} + r_{t+2} + \cdots + r_n$ //

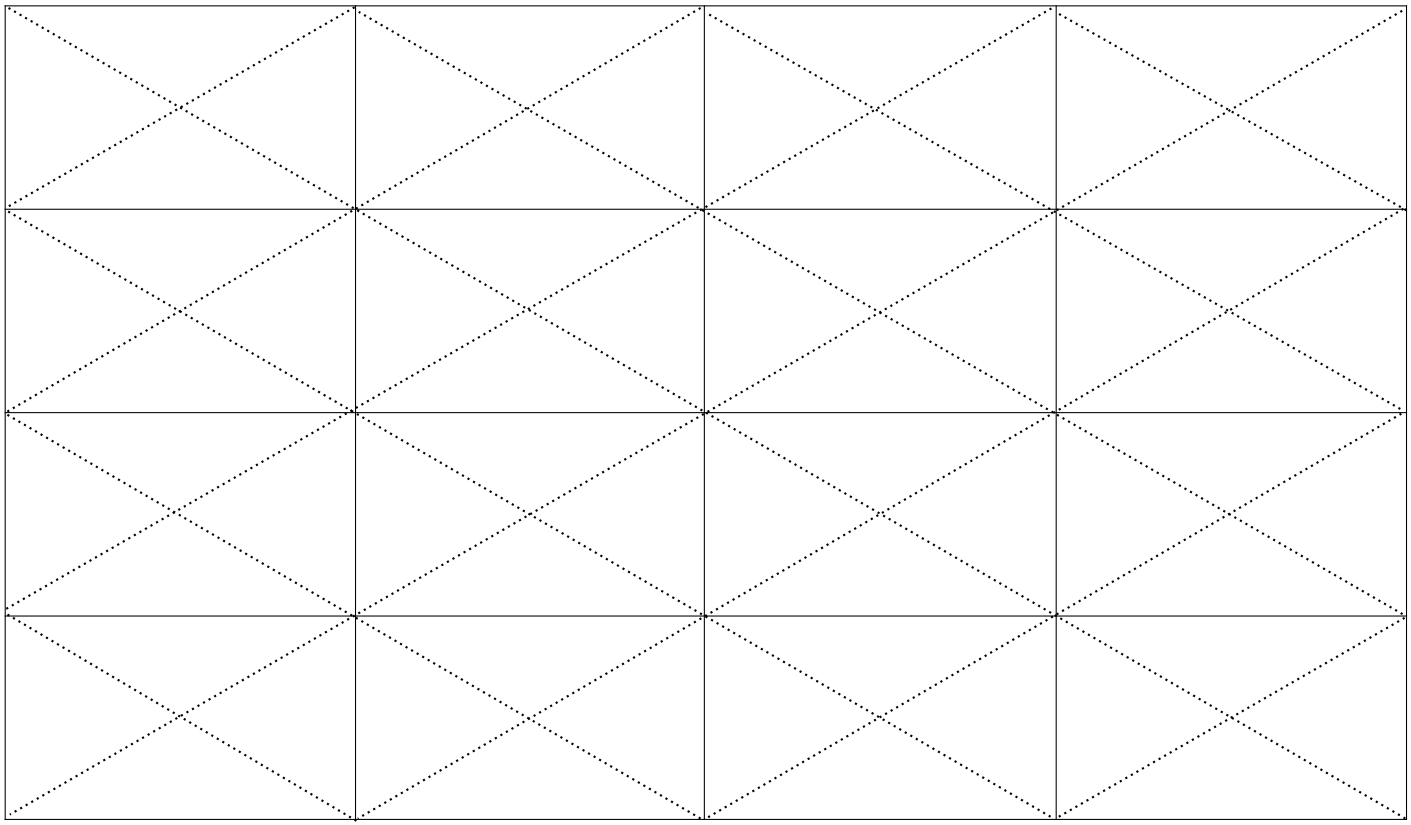
$$Q(s, a) = r + \max_{a'} Q(s', a')$$

Learning Q (s, a)?

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

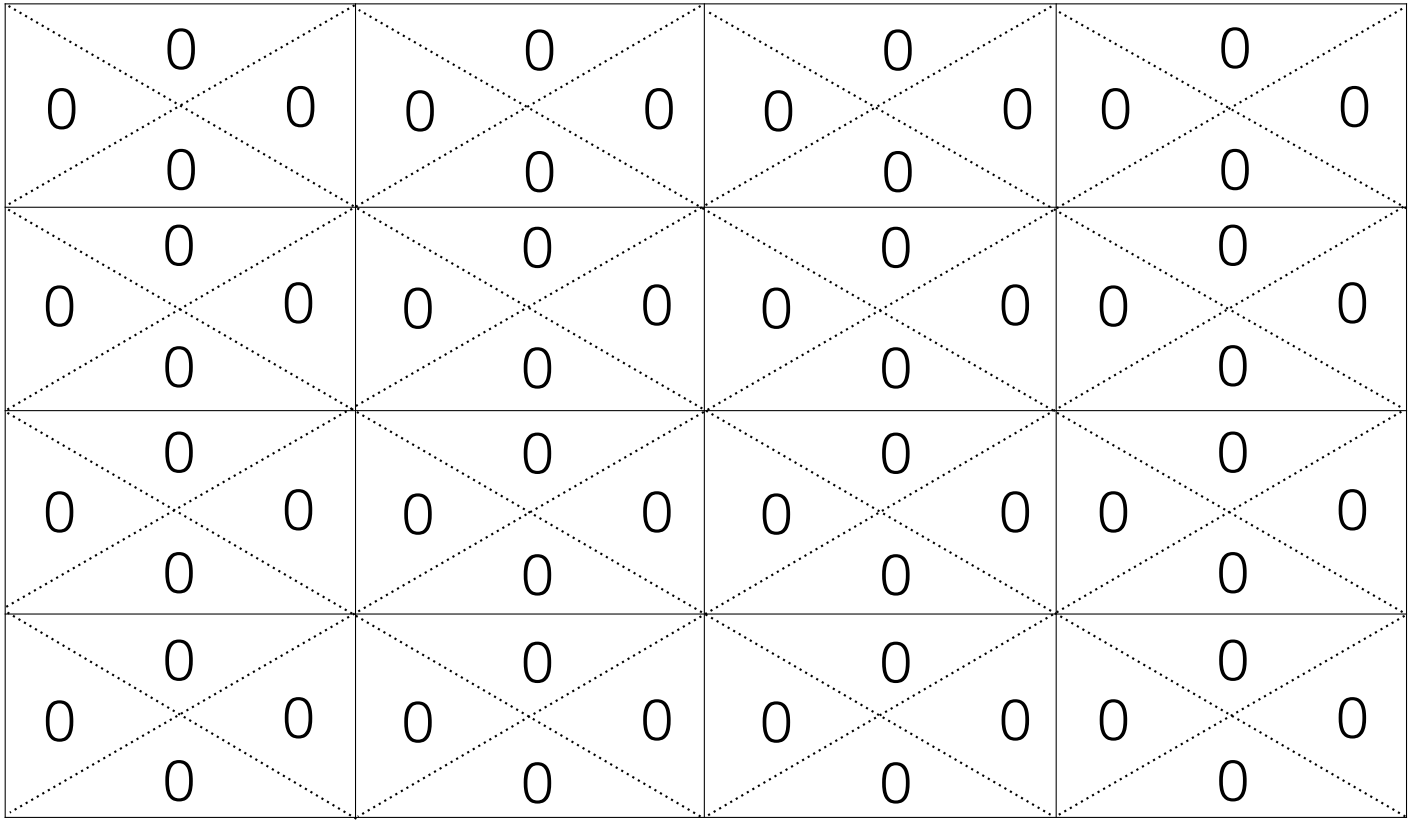
Machine Learning, T. Mitchell, McGraw Hill, 1997

Learning Q(s, a): 16x4 Table
16 states and 4 actions (up, down, left, right)



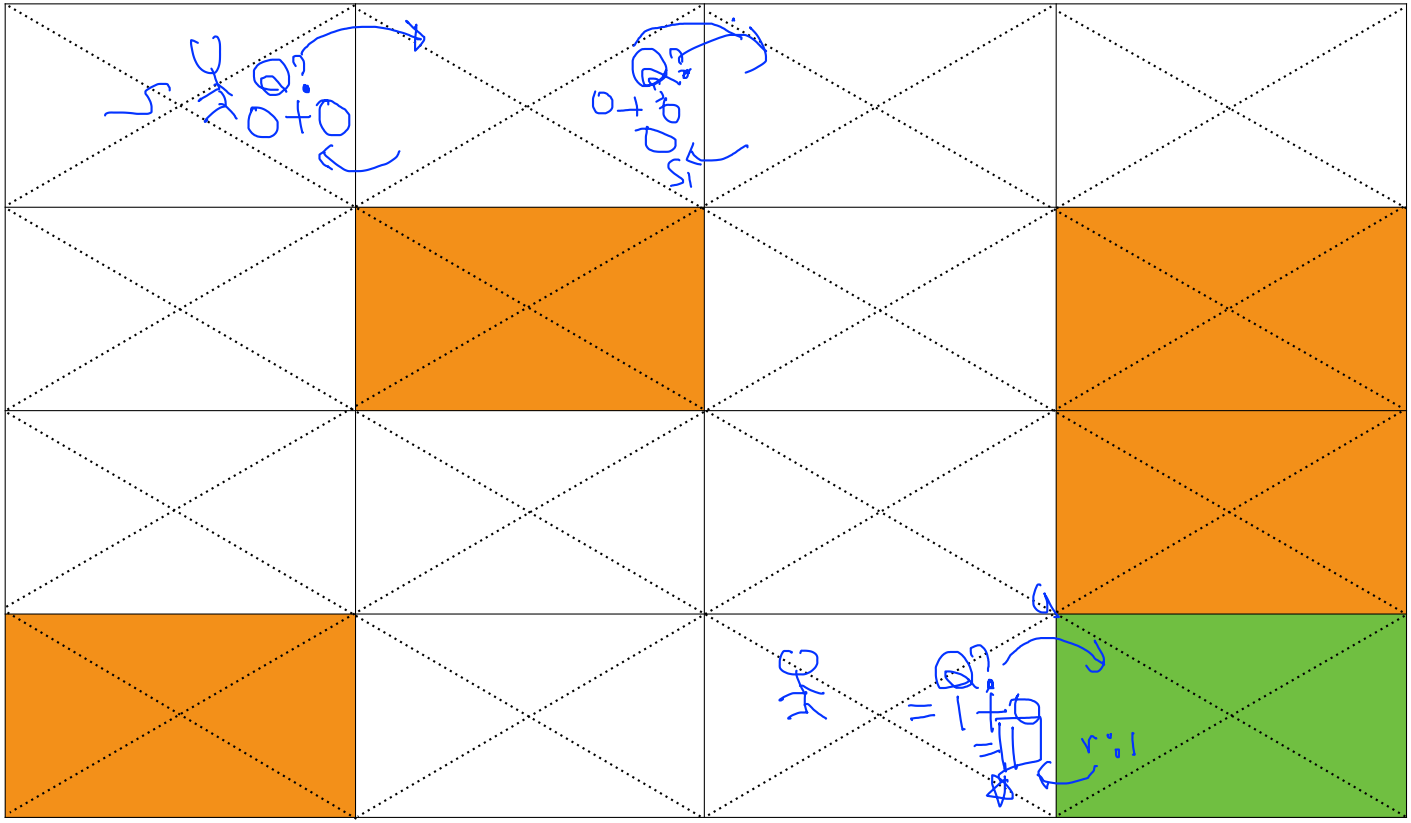
Learning Q(s, a): Table

initial Q values are 0



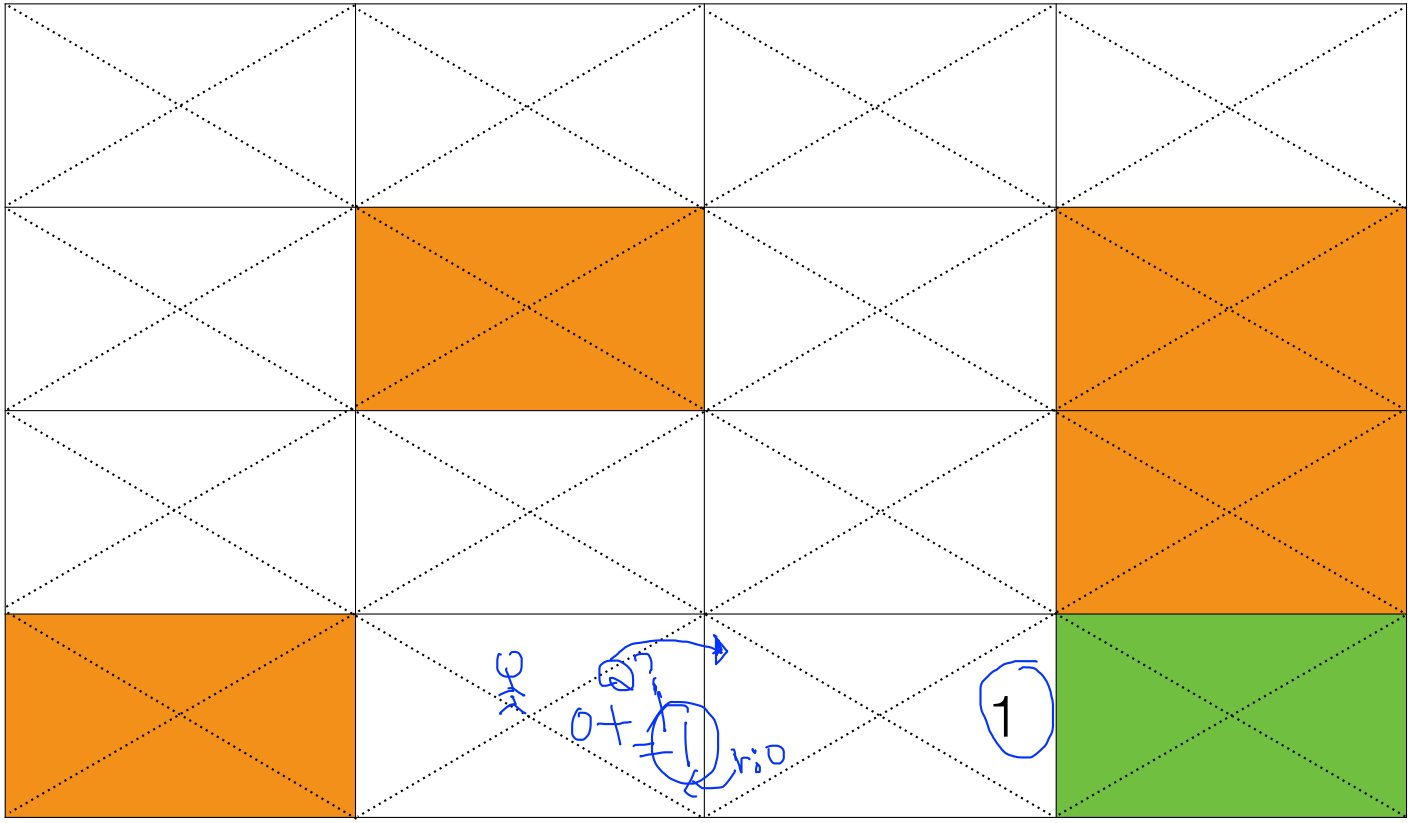
Learning Q(s, a) Table (with many trials)

initial Q values are 0



Learning Q(s, a) Table (with many trials)

initial Q values are 0

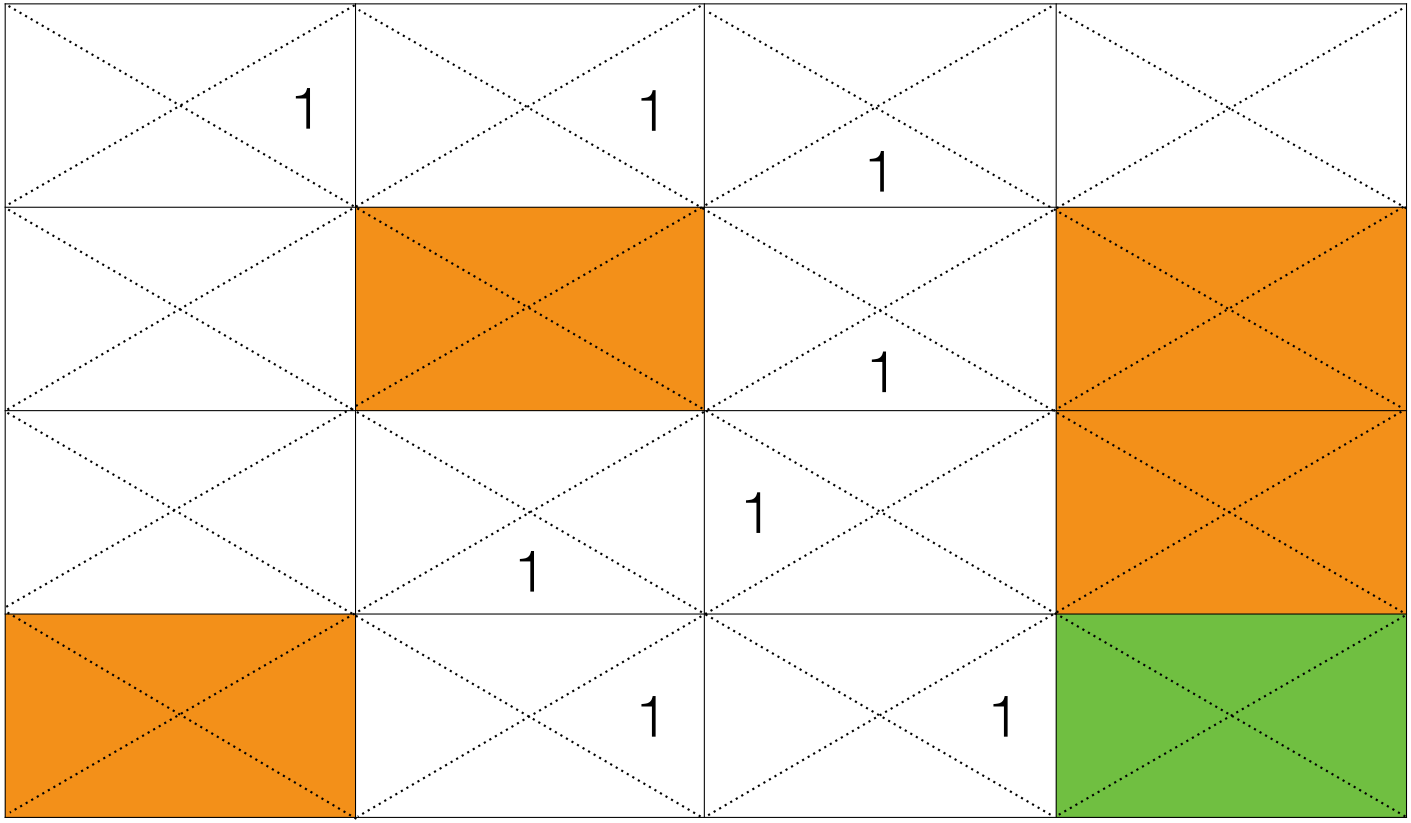


$Q(s_{14}, a_{right}) = r = 1$

$Q(s_{13}, a_{right}) = r + \max(Q(s_{14}, a)) = 0 + \max(0, 0, 1, 0) = 1$

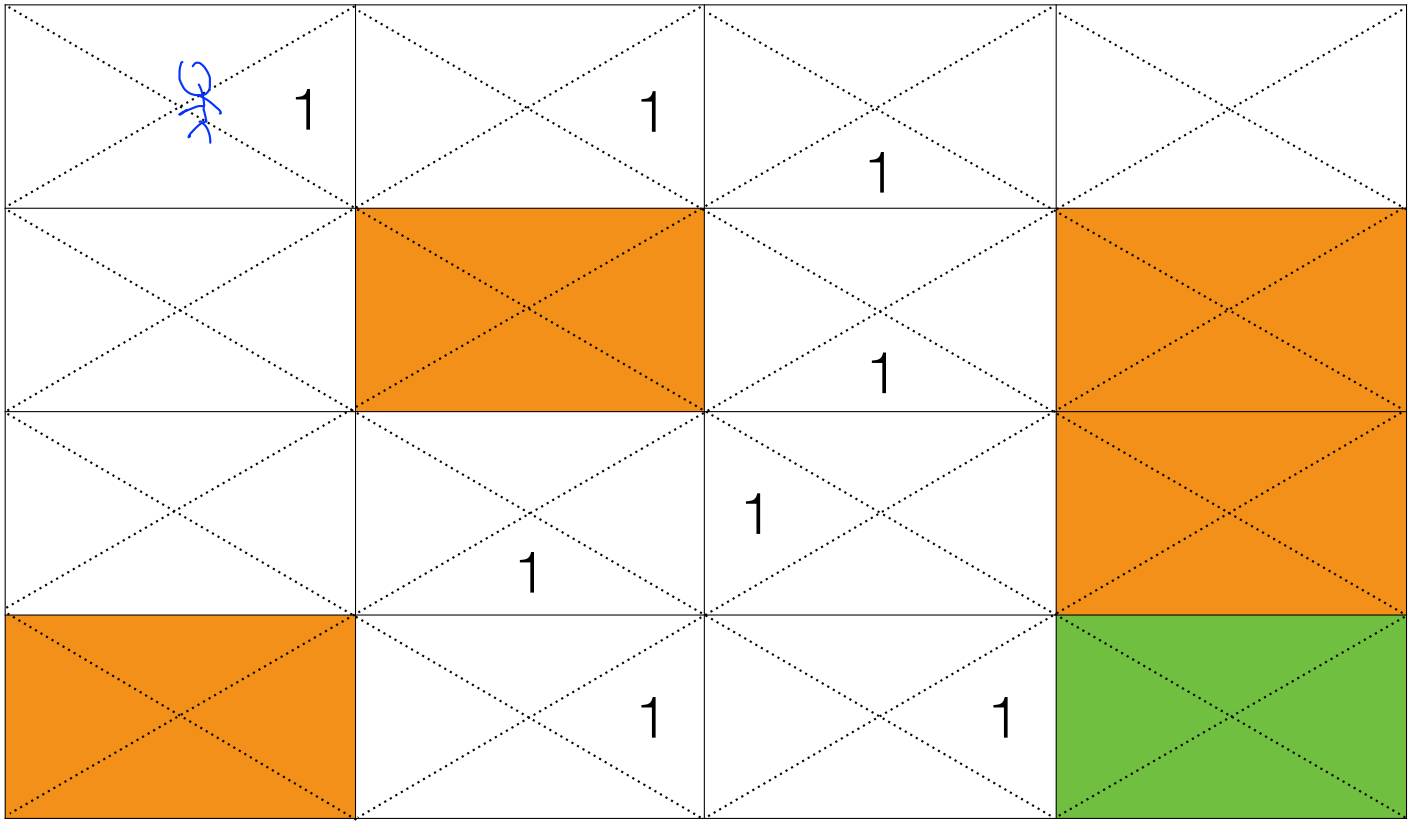
Learning Q(s, a) Table: one success!

initial Q values are 0



Learning Q(s, a) Table: optimal policy

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



Dummy Q-learning algorithm

For each s, a initialize table entry $\hat{Q}(s, a) \leftarrow 0$

Observe current state s

Do forever:

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s, a)$ as follows:

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

- $s \leftarrow s'$

Machine Learning, T. Mitchell, McGraw Hill, 1997

Next
Lab: Dummy Q-learning Table