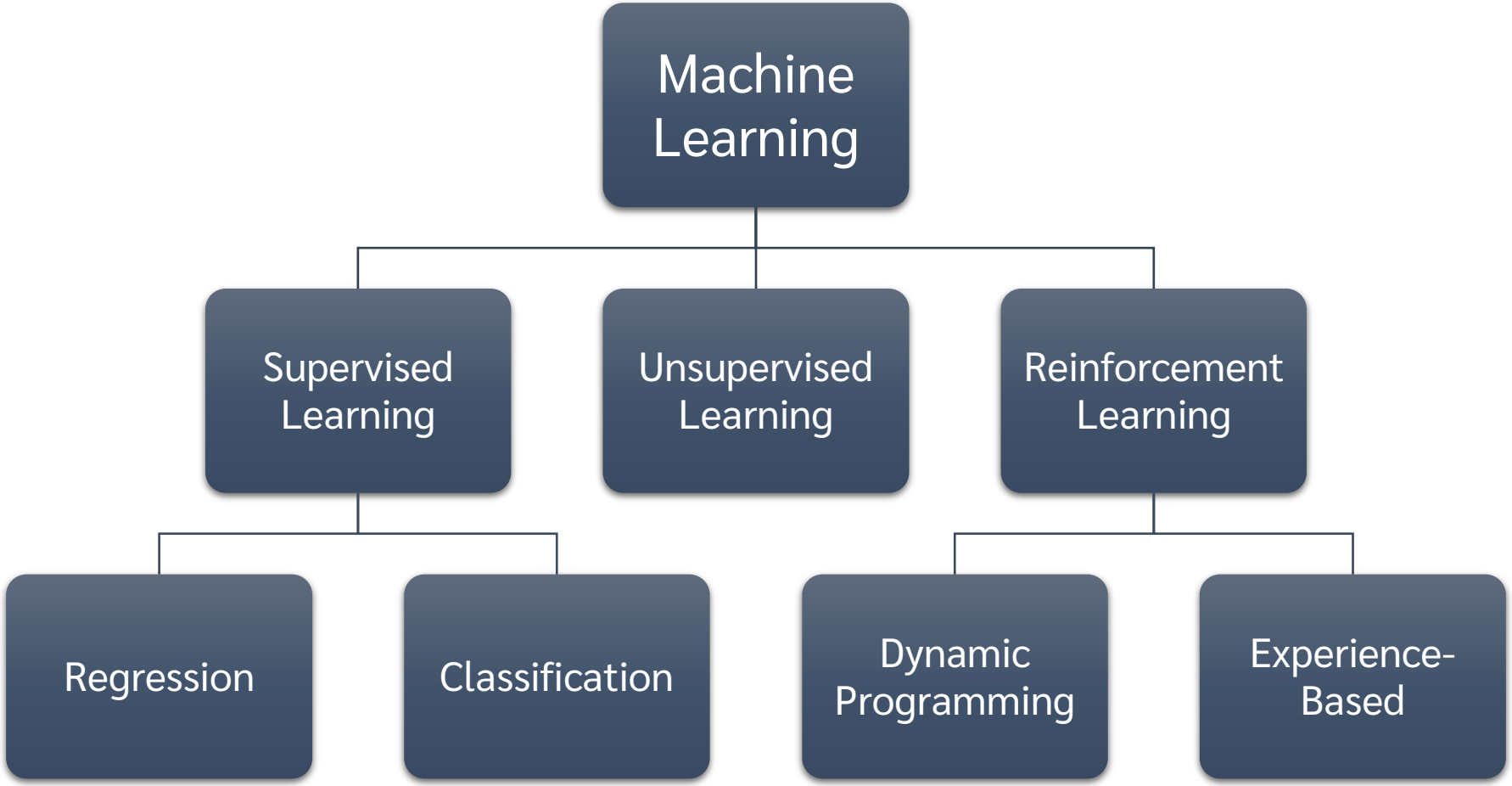
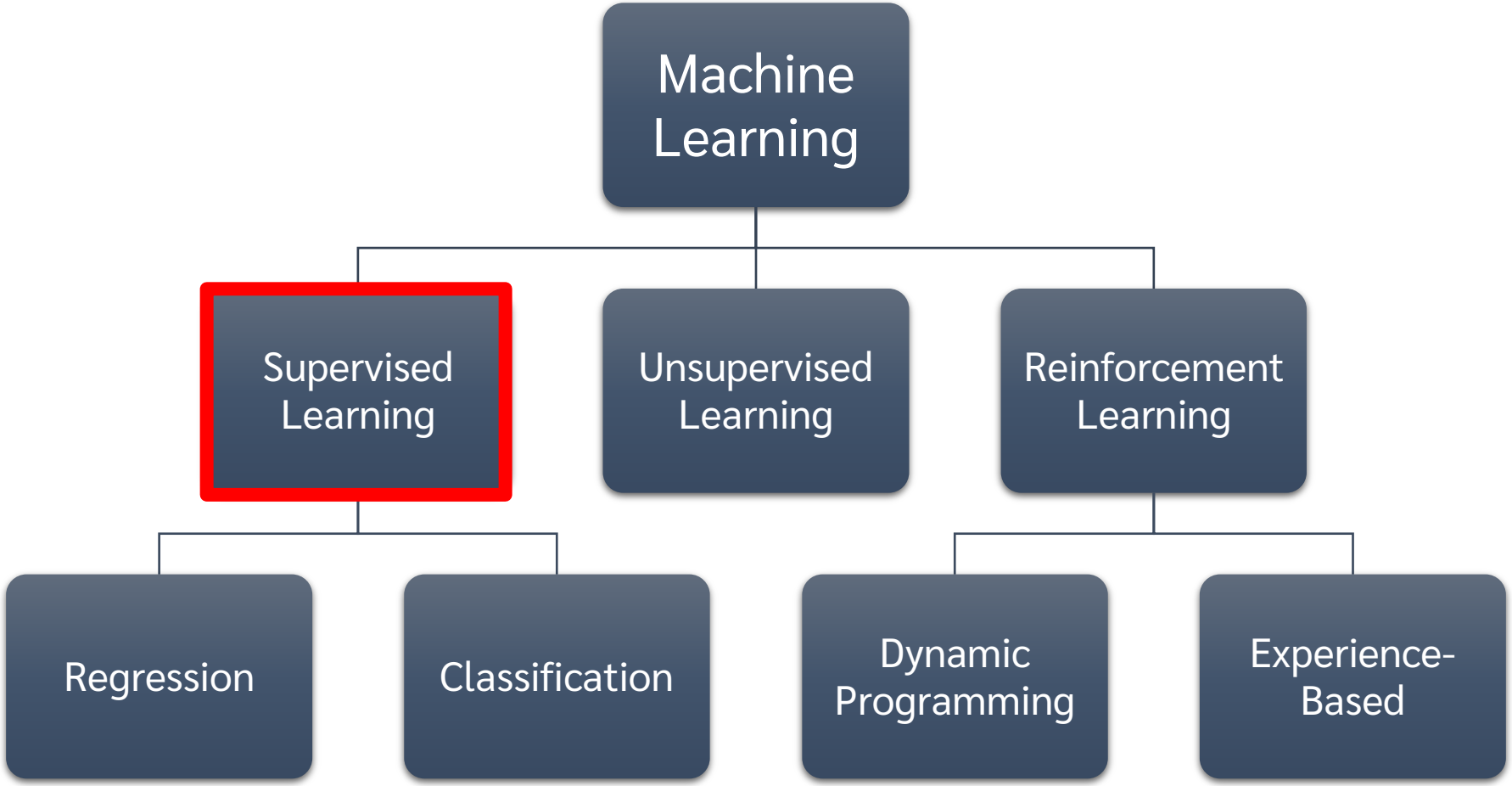
Introduction

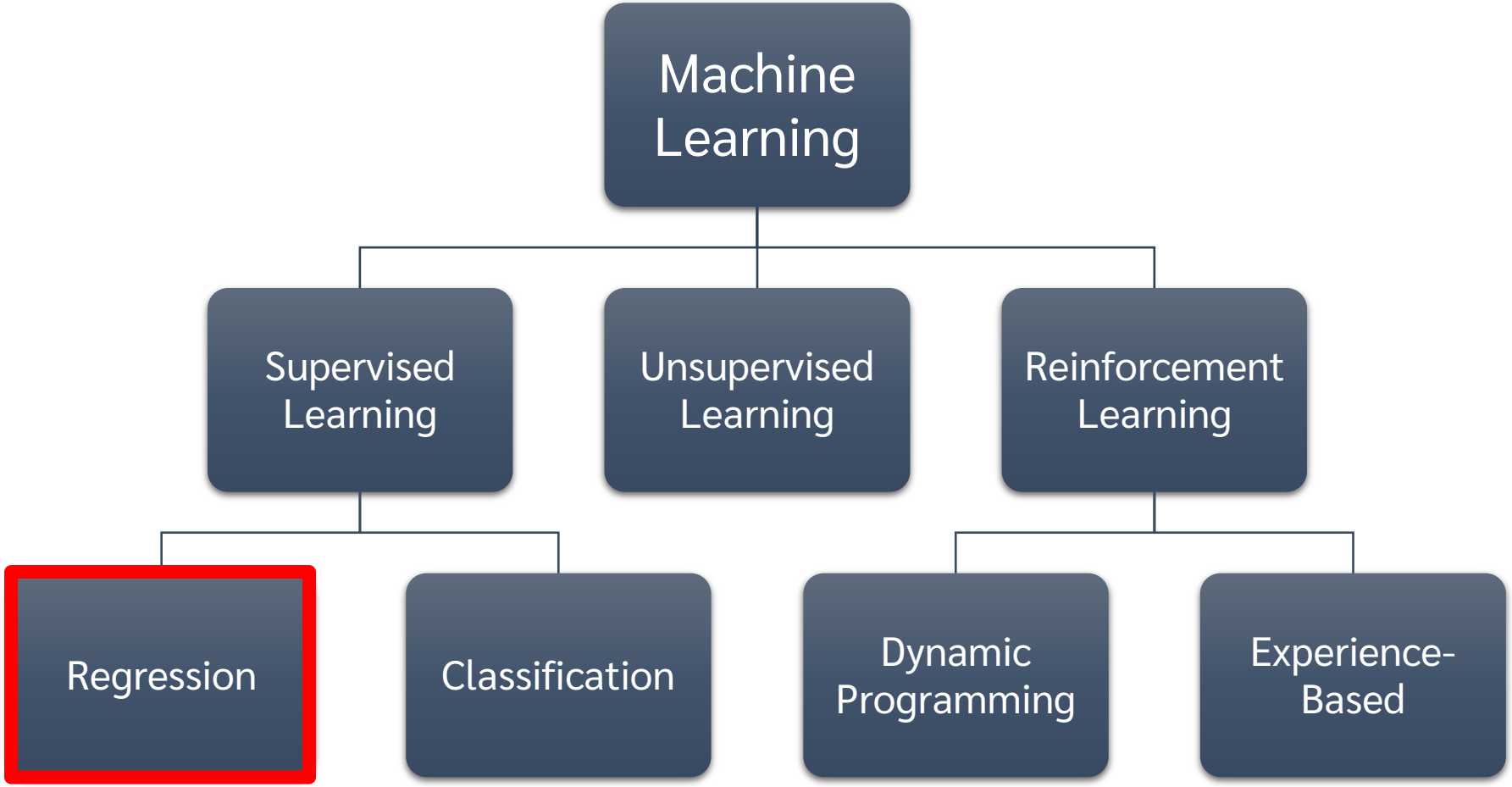
Machine learning



Machine learning



Machine learning



Regression

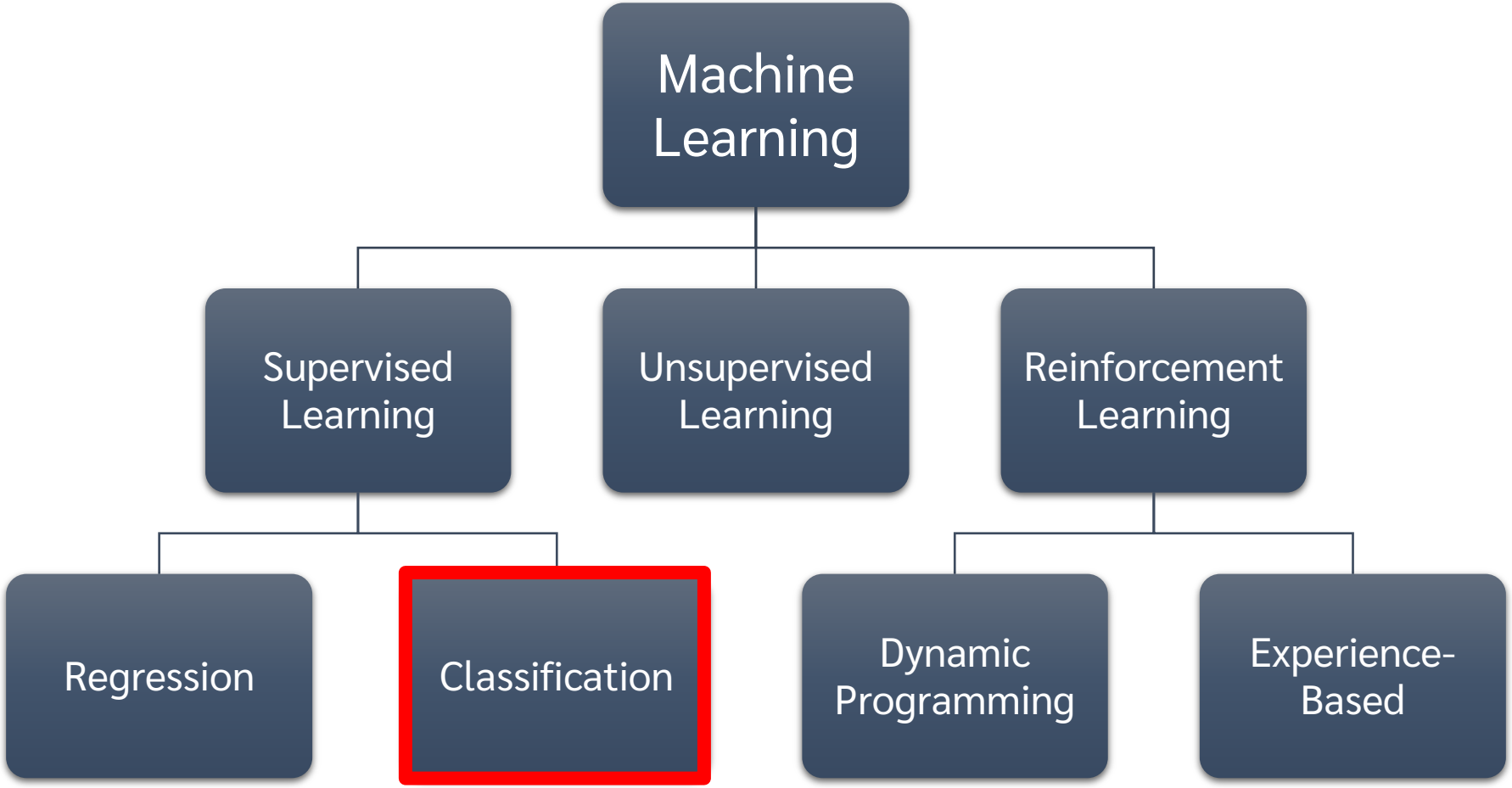
Feature

	ขนาดบ้าน	จำนวนชั้น
1	195	2
2	210	1
3	100	2
4	150	2
5	125	1

Target

	ราคา
1	10,000,000
2	18,000,000
3	6,000,000
4	8,000,000
5	?

Machine learning



Classification

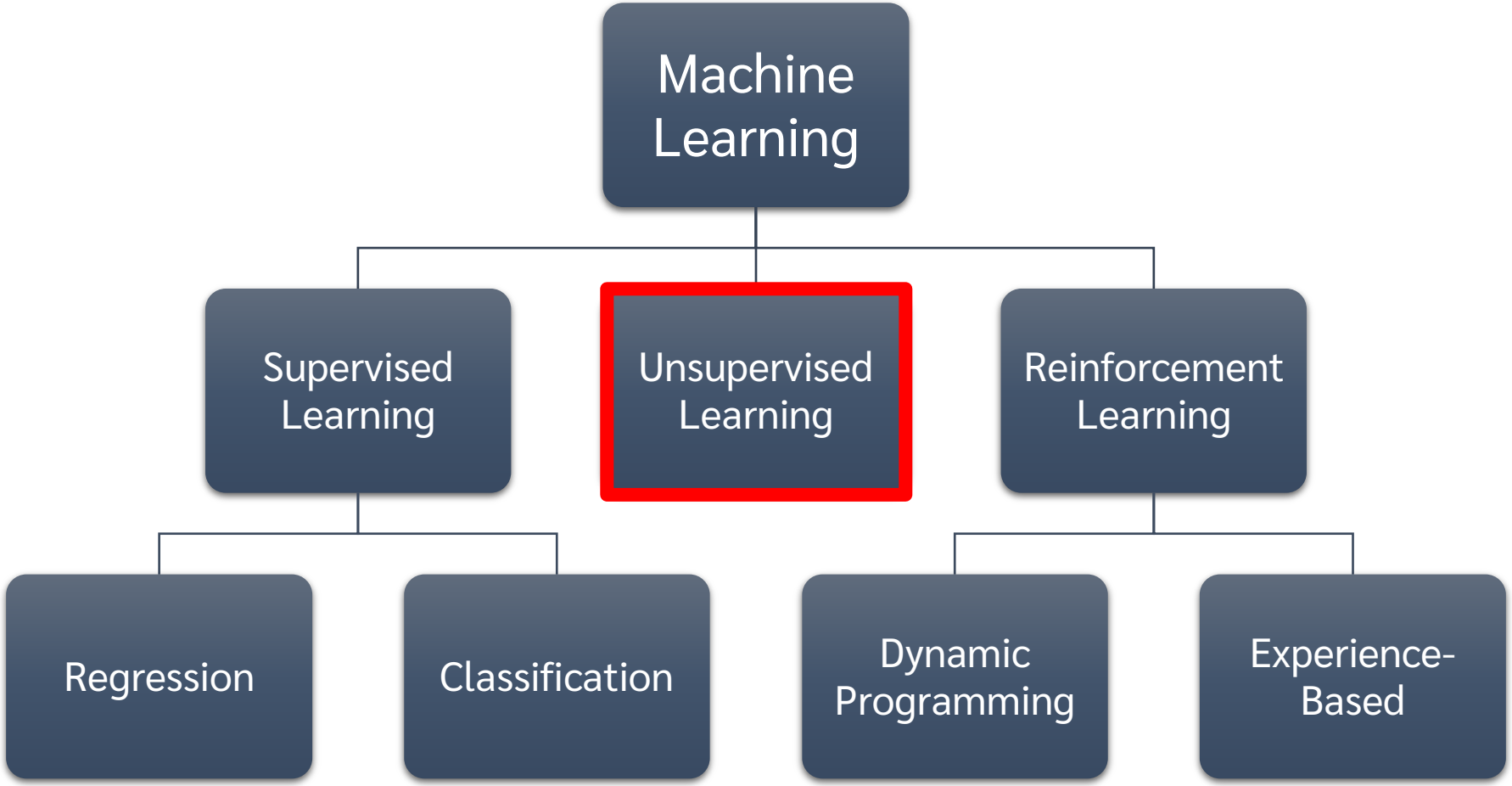
Feature

	อายุ (ปี)	ความดัน (mmHg)
1	30	130
2	70	160
3	40	140
4	70	138
5	45	132

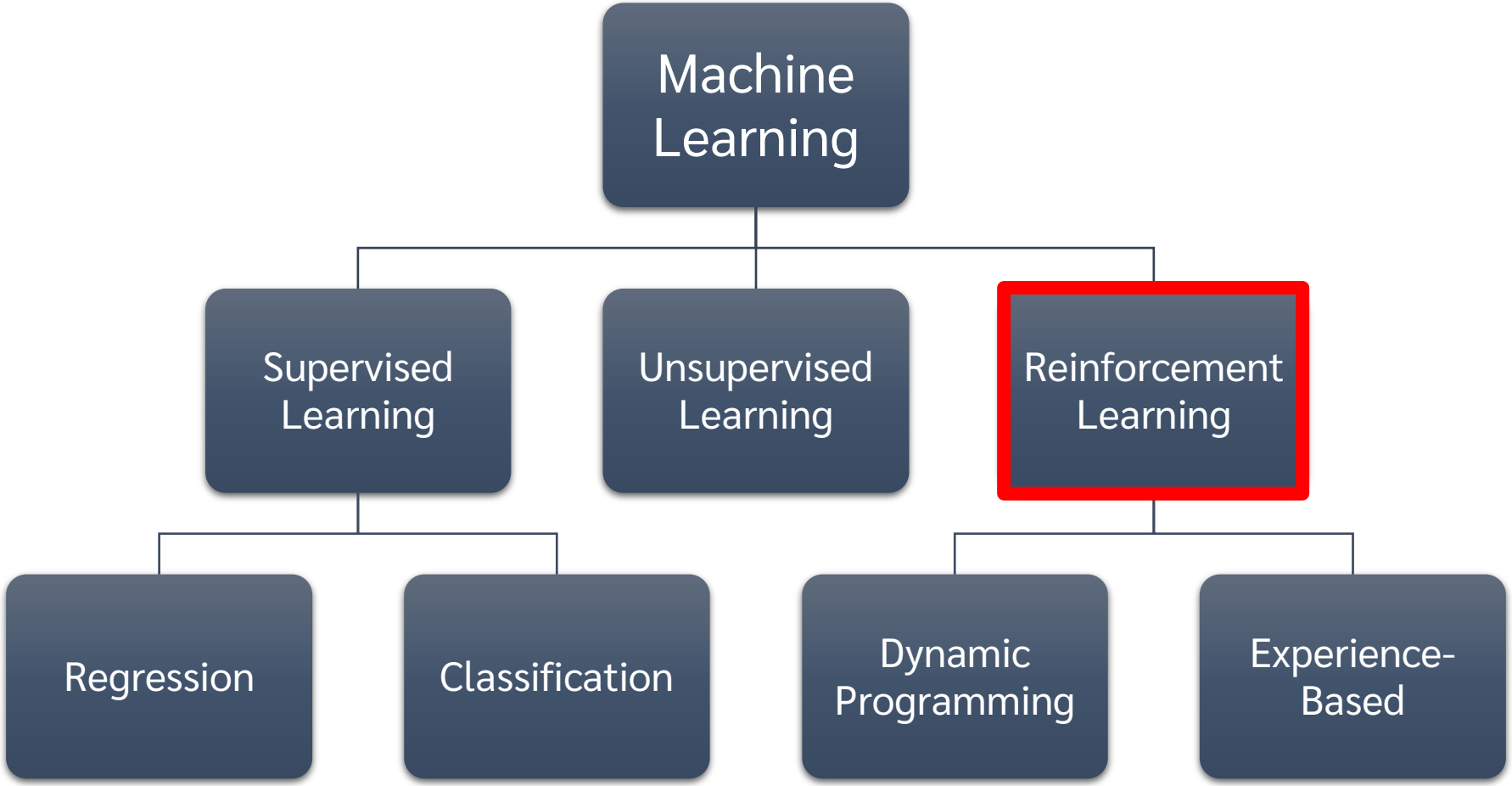
Target

	เป็นเบาหวาน ?
1	ไม่เป็น
2	เป็น
3	ไม่เป็น
4	ไม่เป็น
5	?

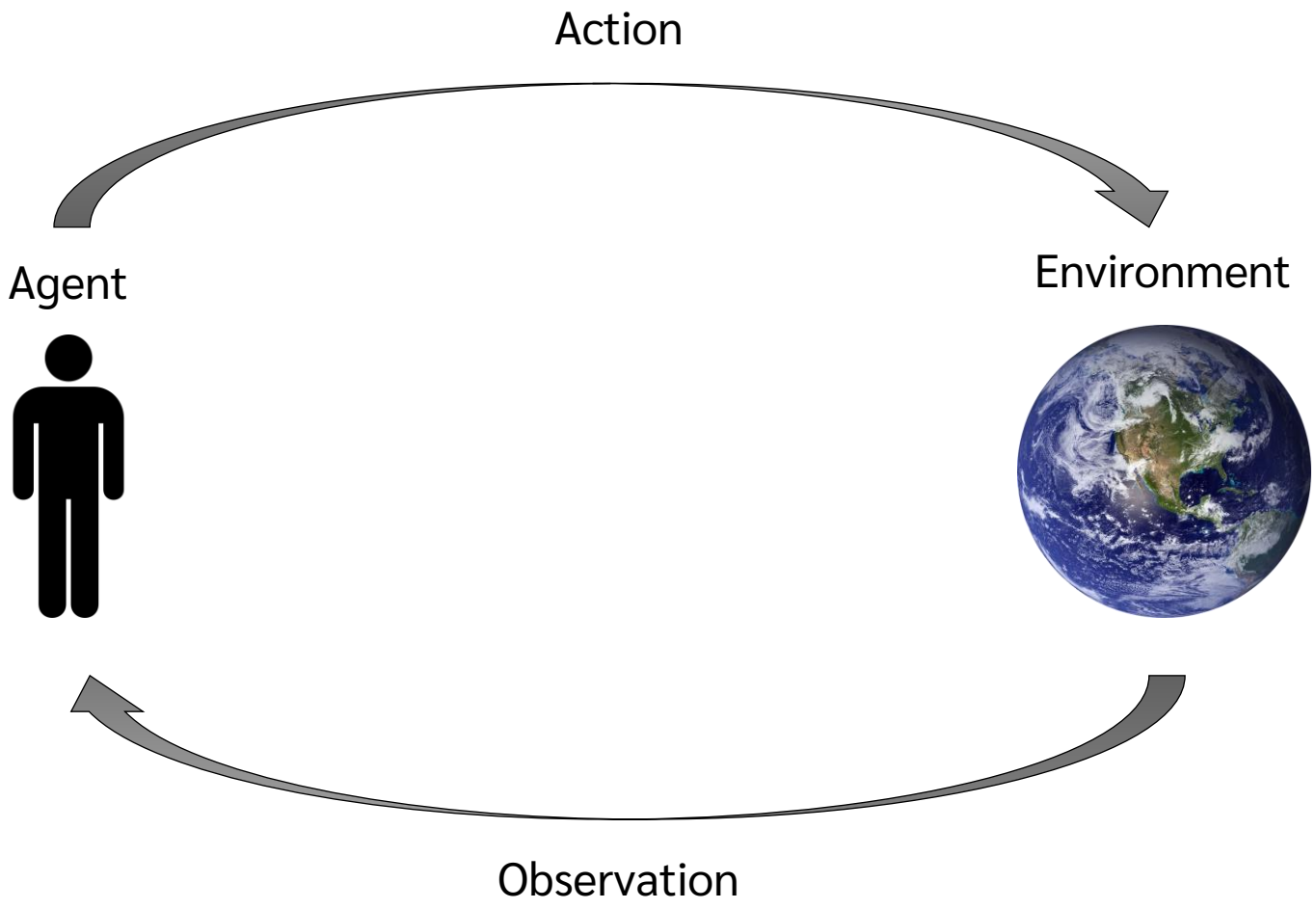
Machine learning



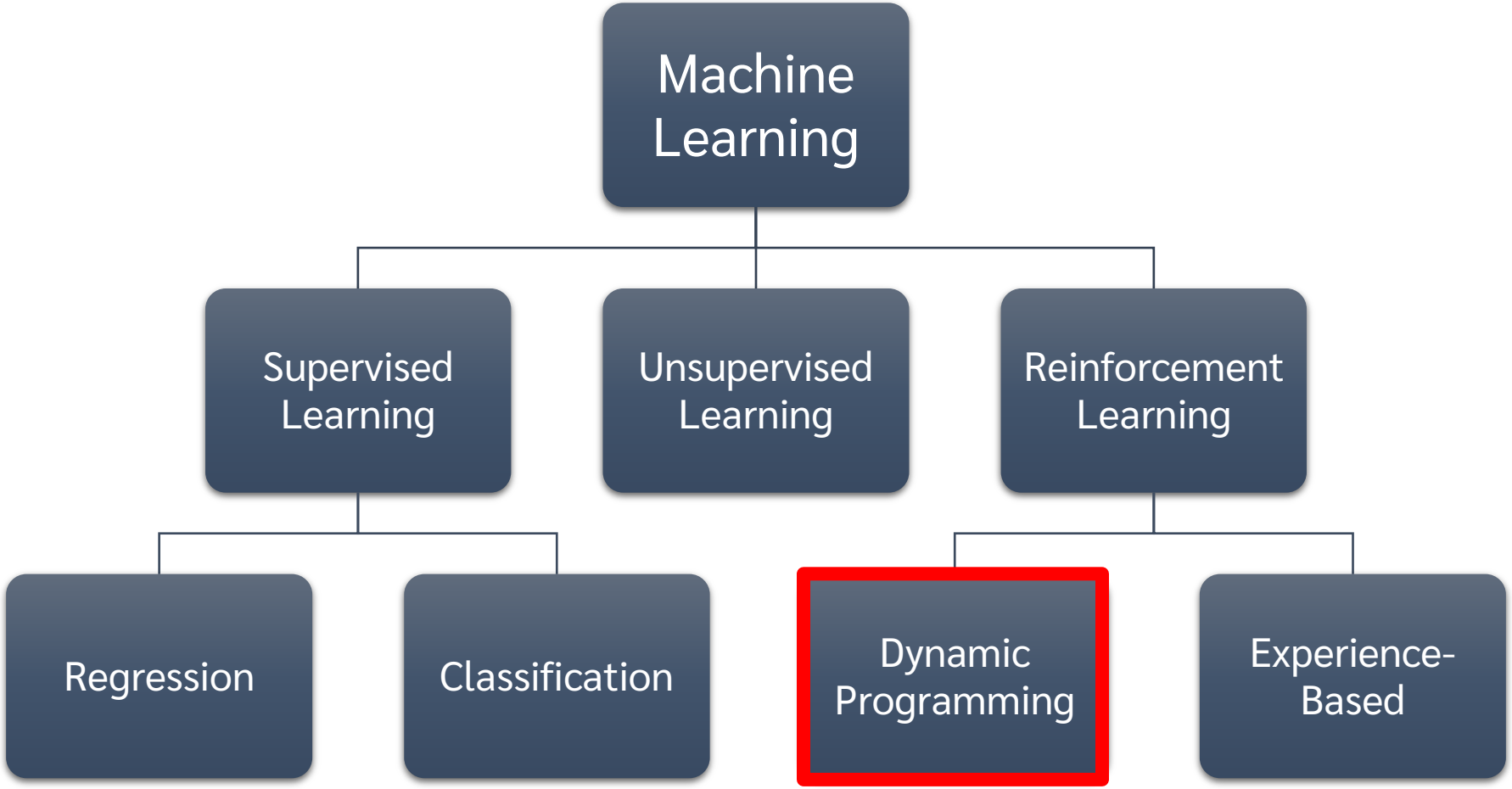
Machine learning



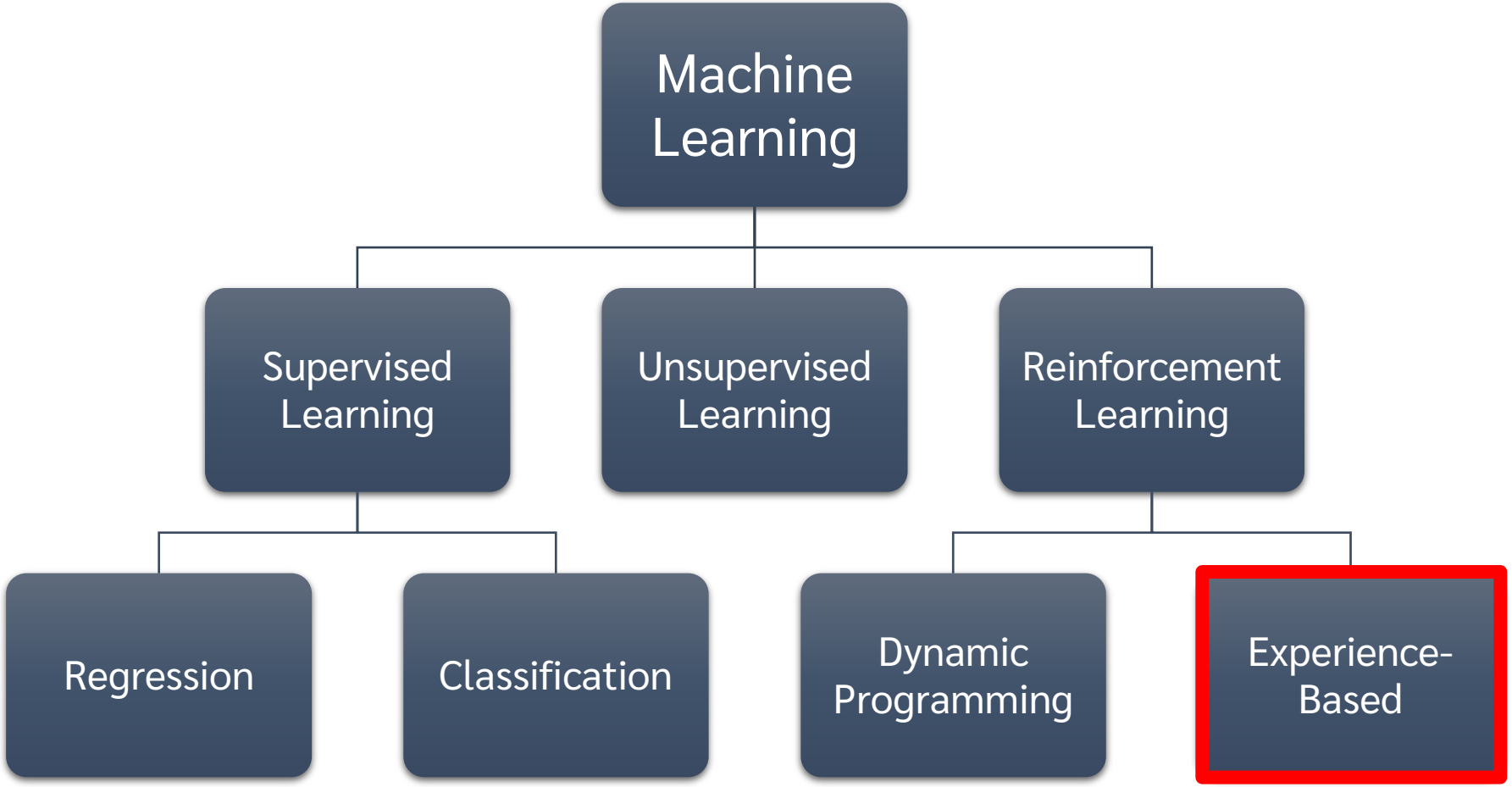
Reinforcement learning



Machine learning



Machine learning



Goal

สร้าง Agent ที่เล่นเกม Atari ได้ในระดับ Super-Human Level

การสร้าง Agent ที่เล่นเกมได้เก่งสำคัญอย่างไร

การสร้าง Agent ที่เล่นเกมได้เก่งสำคัญอย่างไร

สามารถนำความรู้ที่ได้จากการสร้างเกม
ไปใช้แก้ปัญหาที่ซับซ้อนขึ้นในโลกความเป็นจริงได้

ทำไมเราถึงควรเริ่มจากการสร้าง Agent เพื่อเล่นเกม

ทำไมเราถึงควรเริ่มจากการสร้าง Agent เพื่อเล่นเกม

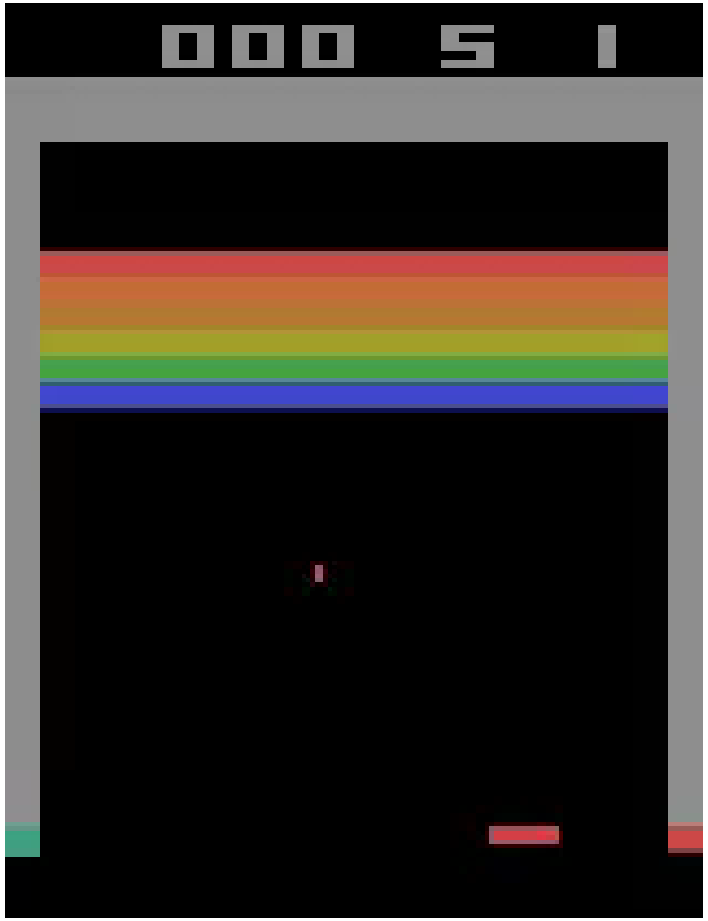
Environment ของเกม มันง่าย ไม่ซับซ้อน

จึงเหมาะกับการเริ่มต้นศึกษา RL

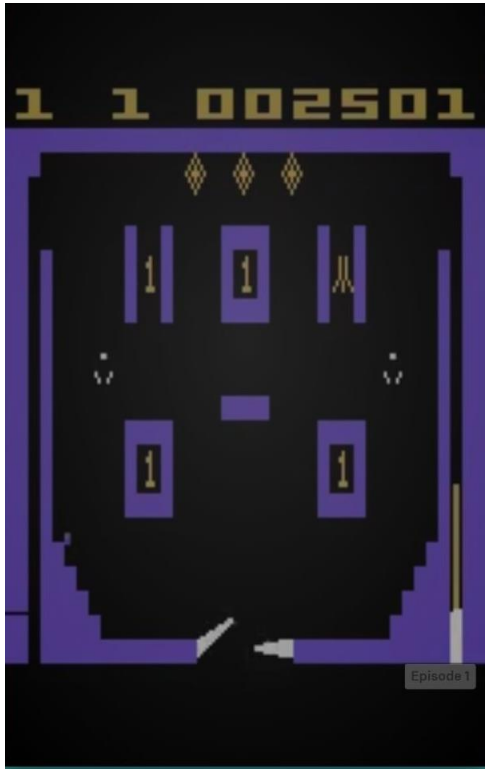
Big Picture

- Theory
  - Fundamentals of Reinforcement learning
  - Reinforcement learning algorithms
  - Q-learning
  - Deep Q-Networks
- Practice
  - Deep Q-Networks for Atari Breakout
  - Deep Q-Networks for Atari Game

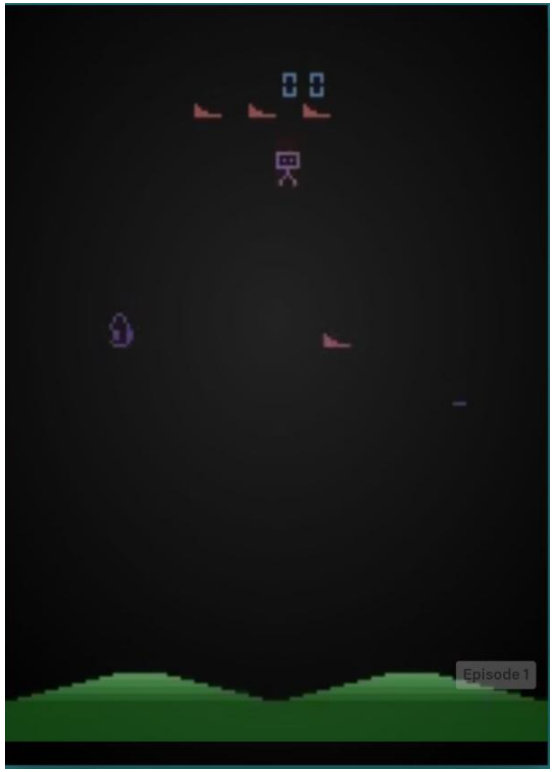
Atari Breakout



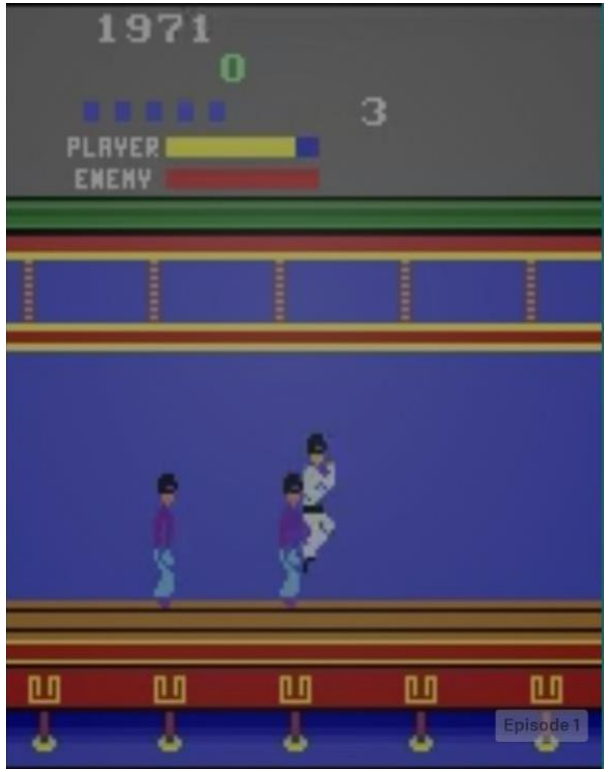
Atari Games



Pinball

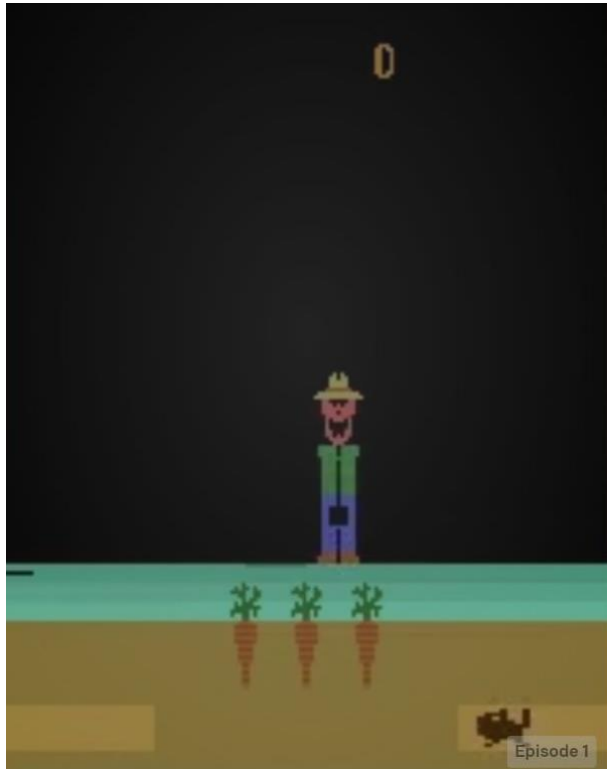


Stargunner



KungFuMaster

Atari Games



Gopher

Freeway

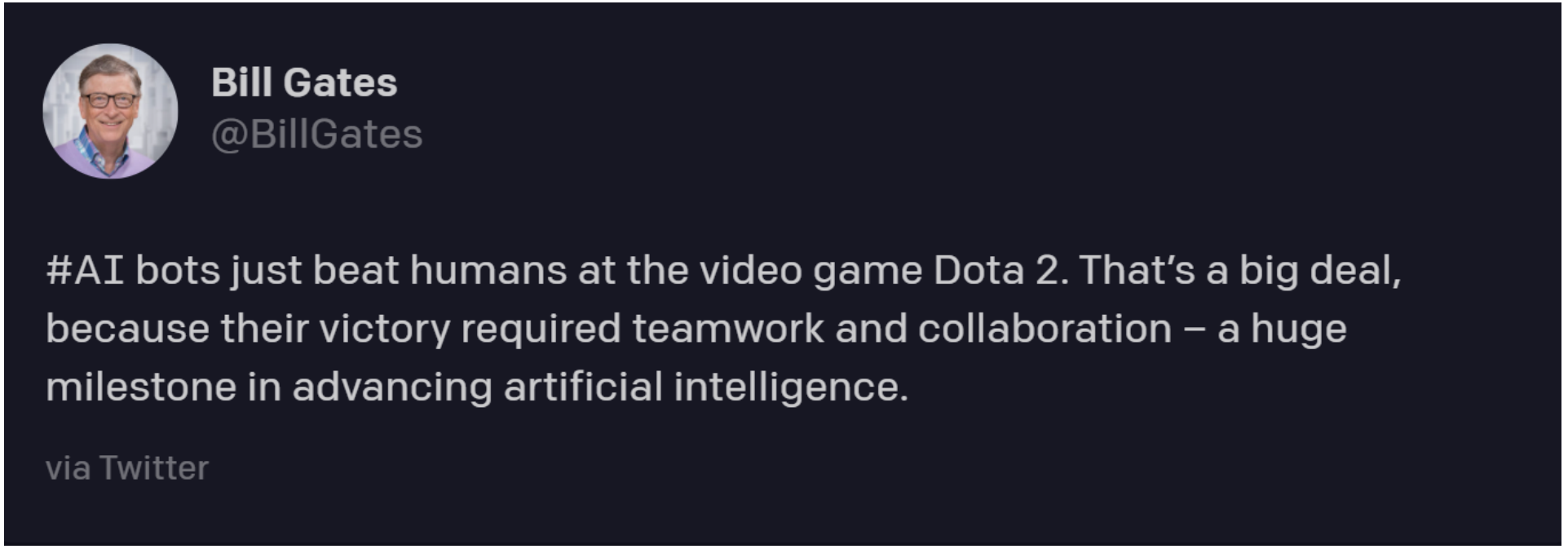
Alpha Go

https://www.alphagomovie.com/

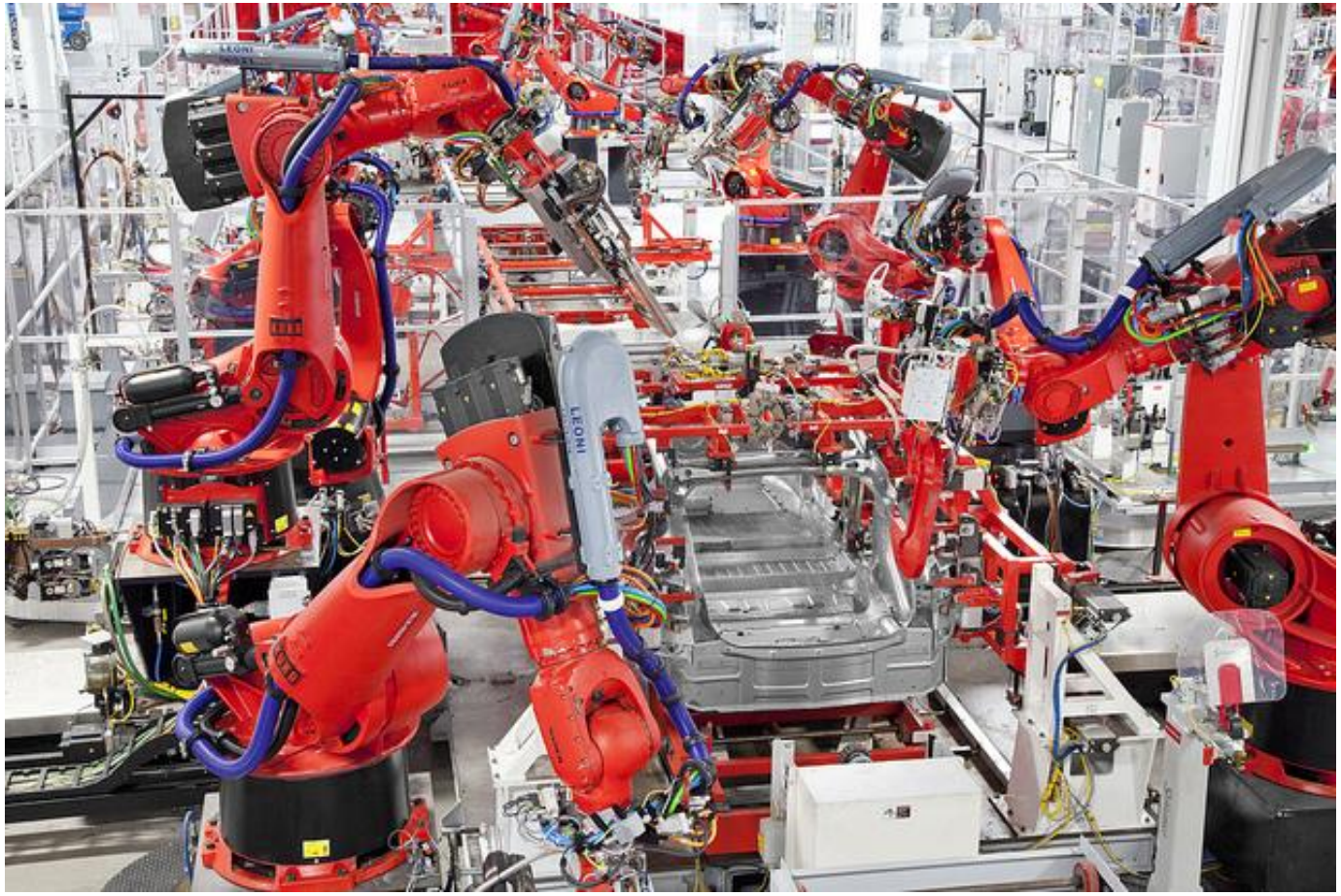
OpenAI Five

reference: https://openai.com/projects/five/

OpenAI Five


Bill Gates
@BillGates
#AI bots just beat humans at the video game Dota 2. That's a big deal, because their victory required teamwork and collaboration – a huge milestone in advancing artificial intelligence.
via Twitter

Robotics



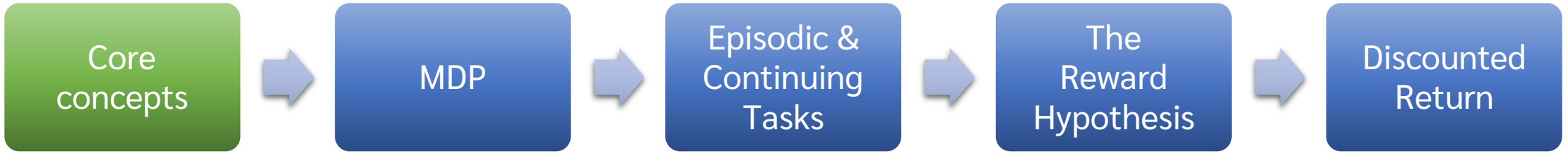
reference: https://spectrum.ieee.org/automaton/robotics/industrial-robots/five-myths-and-facts-about-robotics

Self-driving car

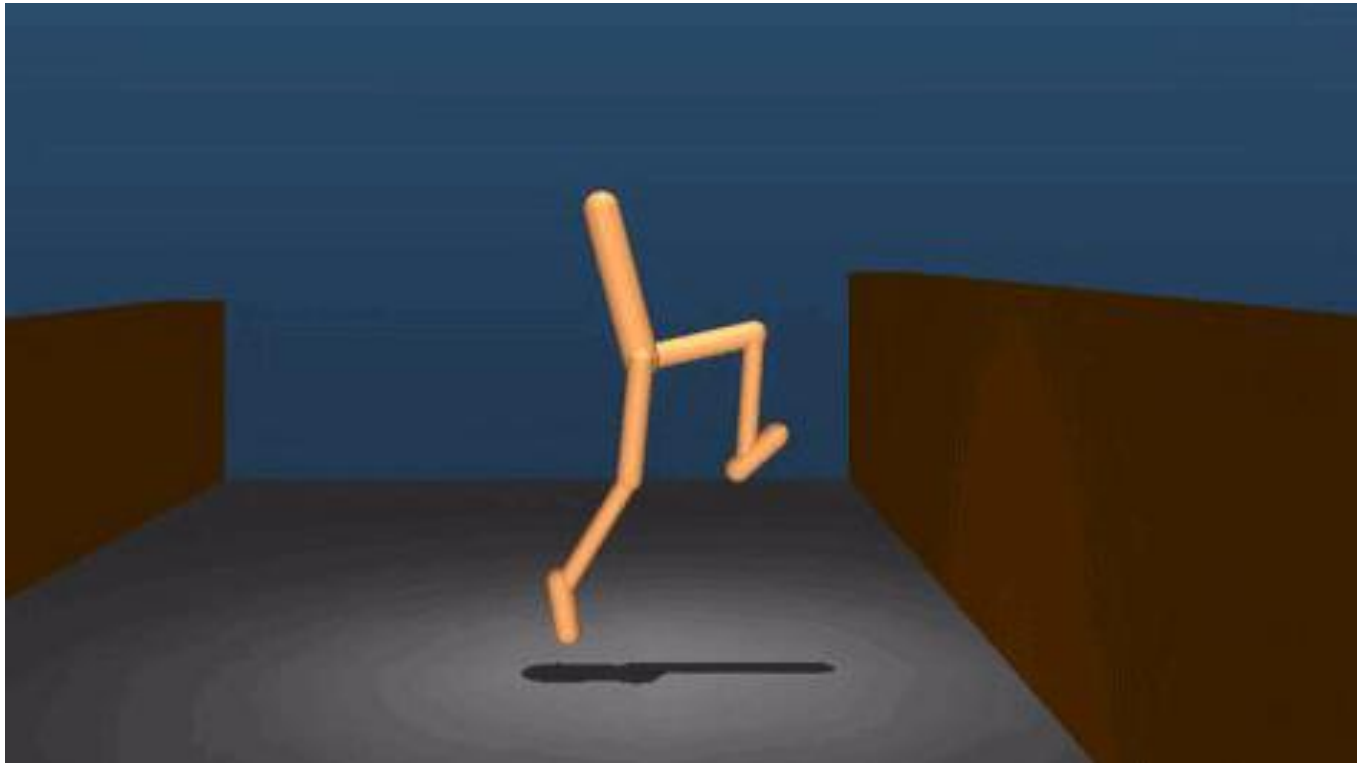
Stock trading

Fundamentals of Reinforcement Learning

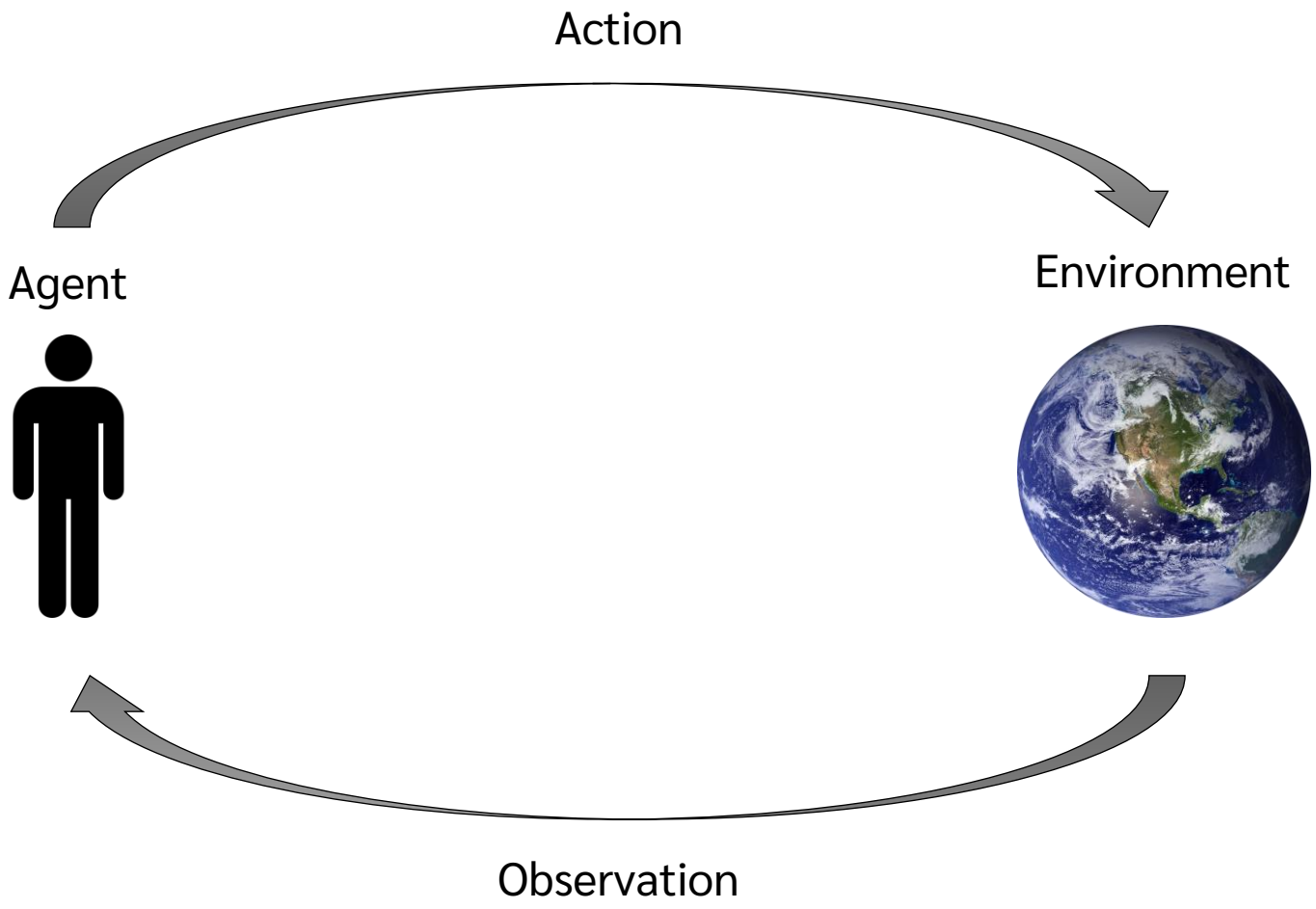
Contents


Core concepts
MDP
Episodic & Continuing Tasks
The Reward Hypothesis
Discounted Return

What is Reinforcement Learning ?



Core Concepts

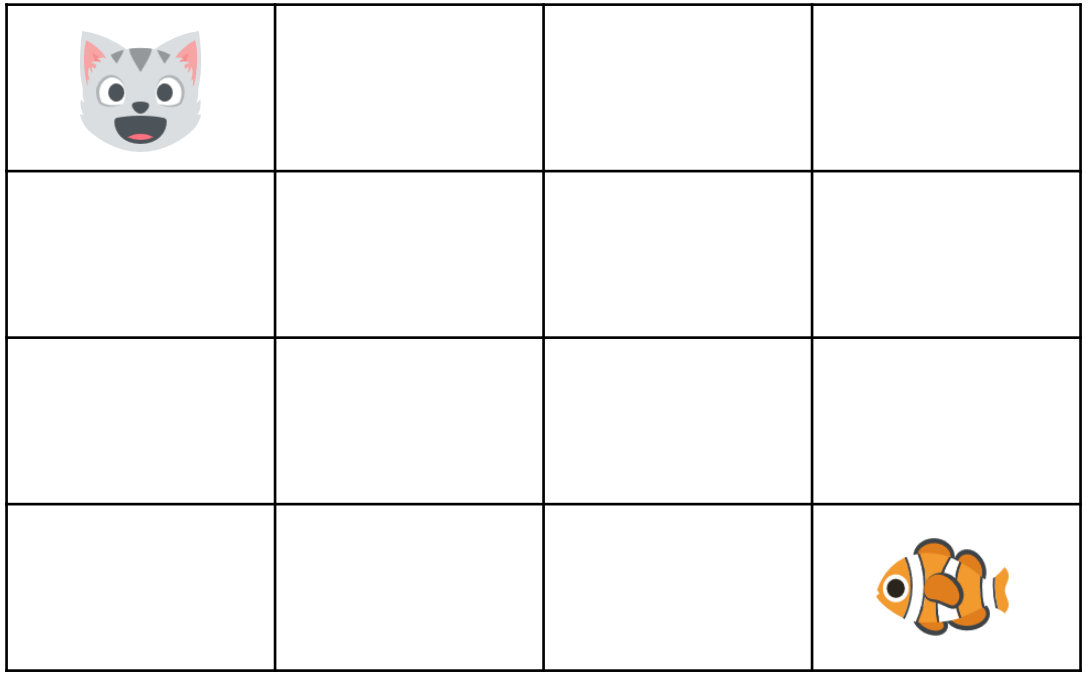


Gridworld

$$\mathcal{S} = \{$$
$$(0,0), (0,1), (0,2), (0,3),$$
$$(1,0), (1,1), (1,2), (1,3),$$
$$(2,0), (2,1), (2,2), (2,3),$$
$$(3,0), (3,1), (3,2), (3,3)$$
$$\}$$

$$\mathcal{A} = \{ up, down, left, right \}$$

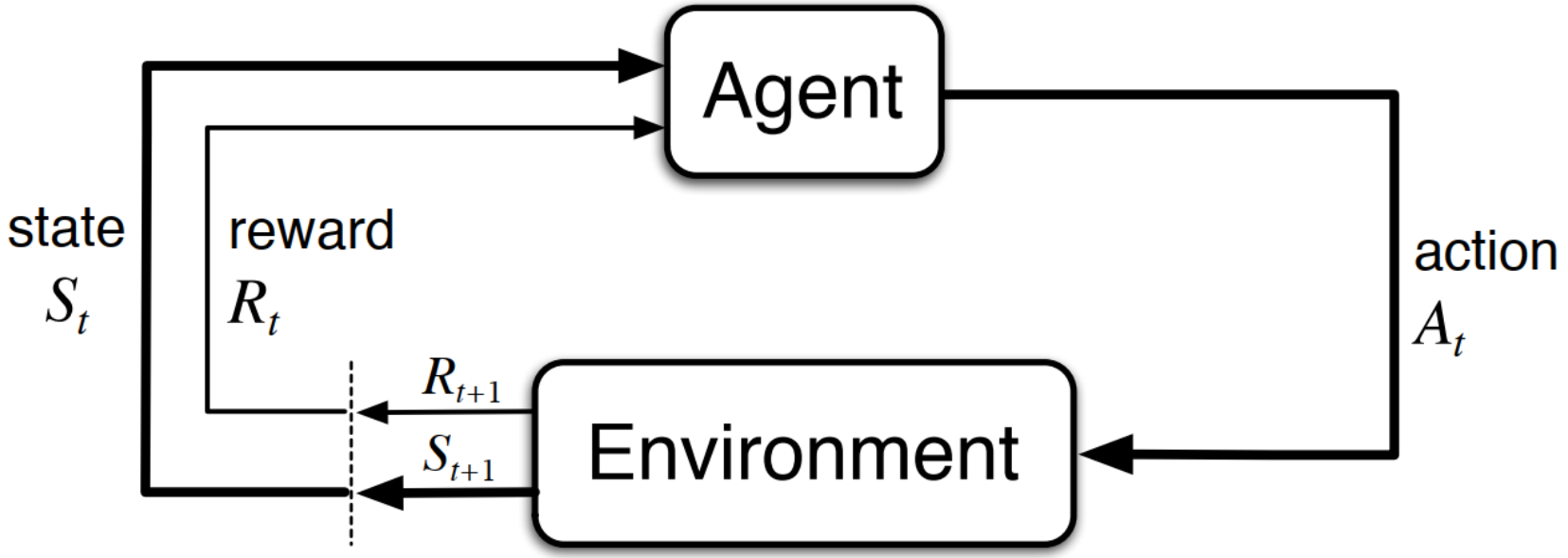
$$\mathcal{R} = \{ -1 , 10 \}$$



Goal

Maximize expected cumulative reward

Agent–Environment Interaction



reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, R_T, S_T$$

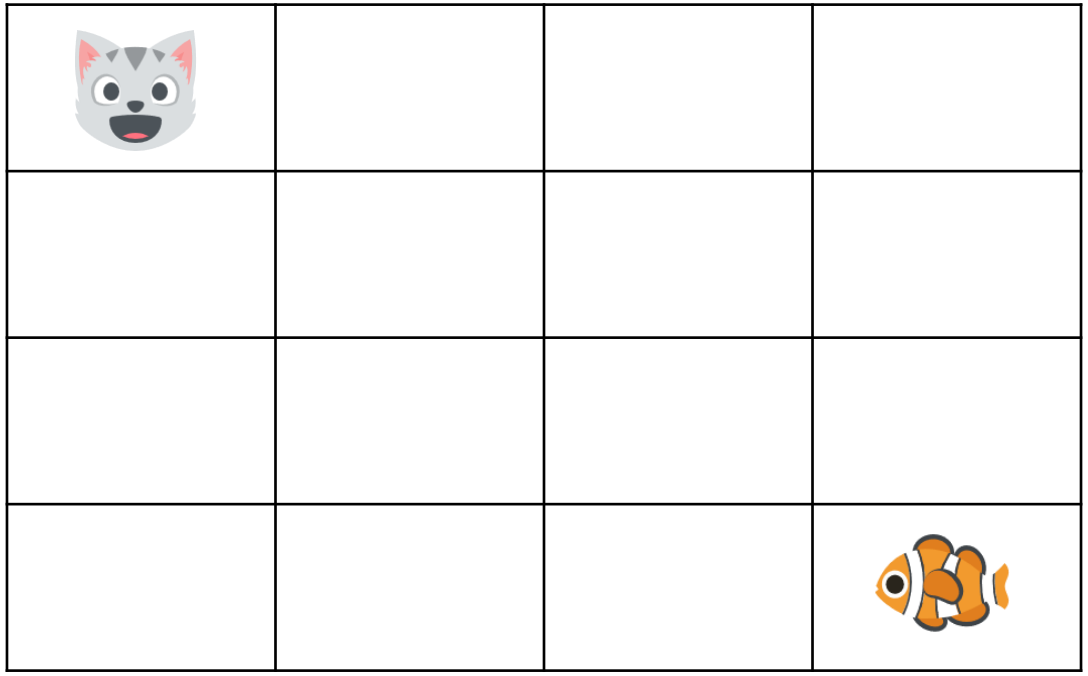
Gridworld

t = 0

$S_0 = (0,0)$

$A_0 = right$

$R_1 = -1$



Gridworld

t = 1

$S_1 = (0,1)$

$A_1 = down$

$R_2 = -1$

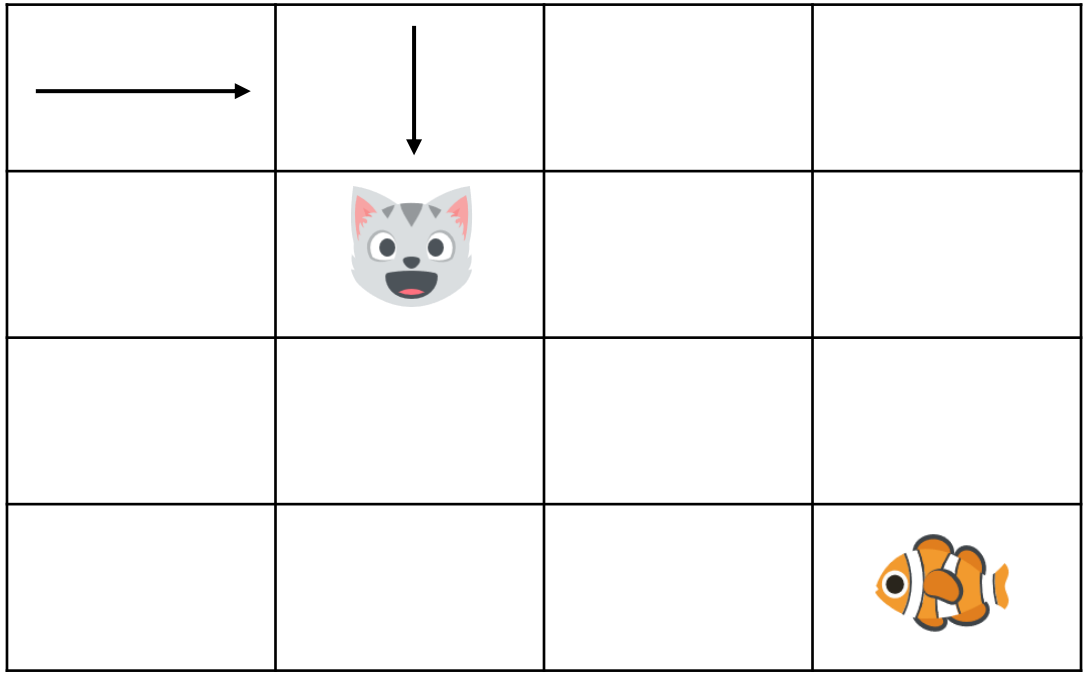
Gridworld

t = 2

$S_2 = (1,1)$

$A_2 = right$

$R_3 = -1$



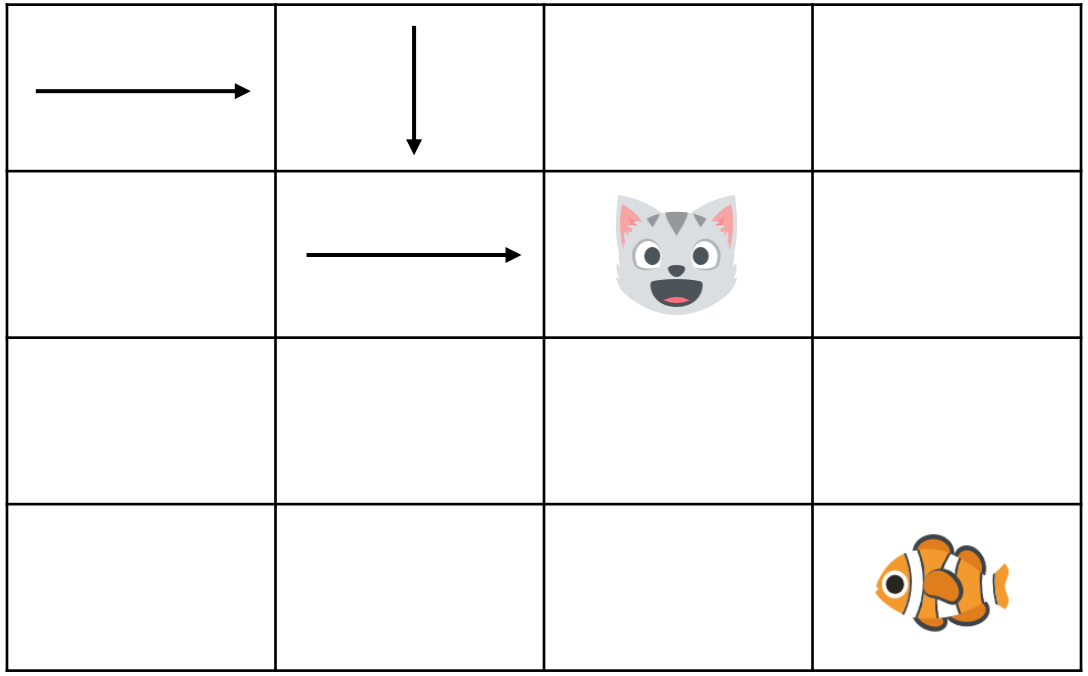
Gridworld

t = 3

$S_3 = (1,2)$

$A_3 = right$

$R_4 = -1$



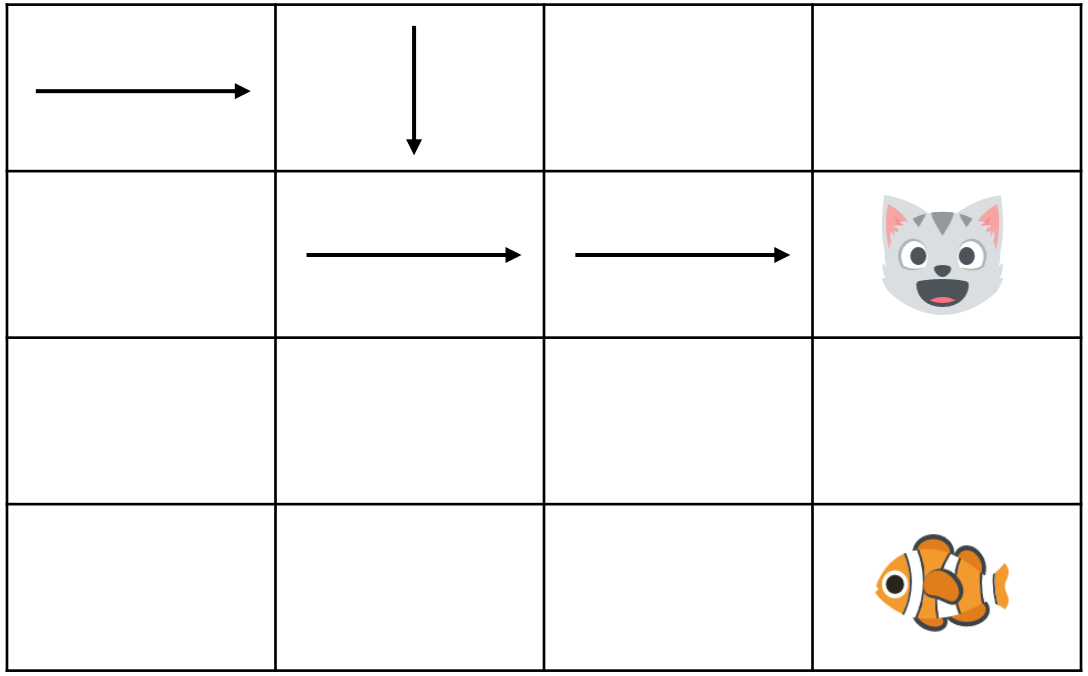
Gridworld

t = 4

$S_4 = (1,3)$

$A_4 = down$

$R_5 = -1$



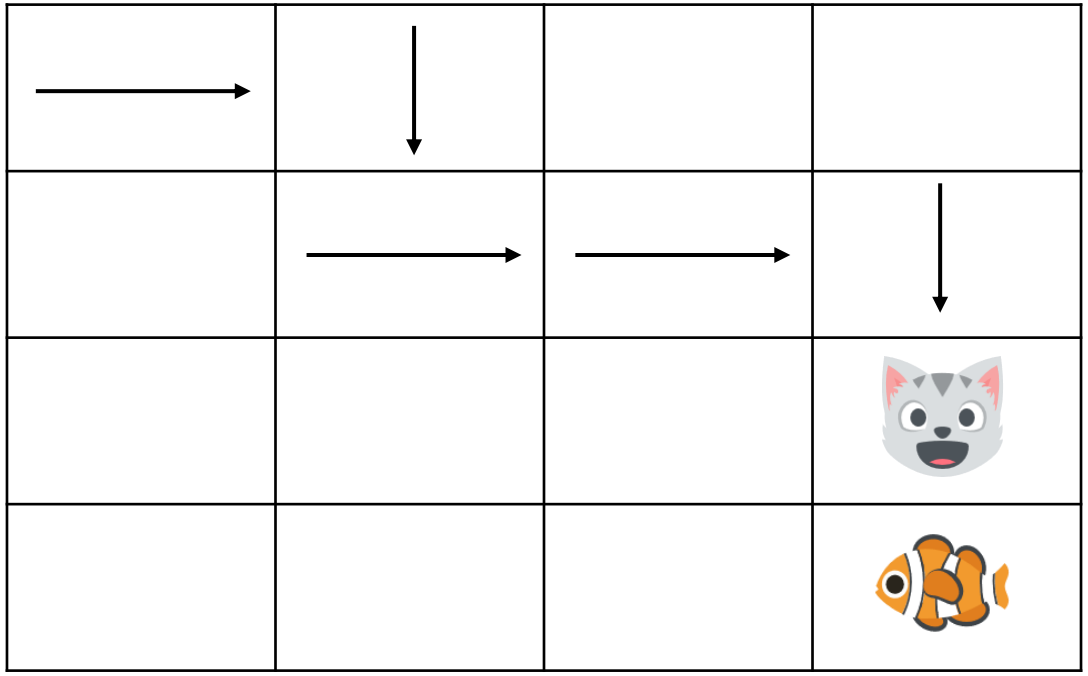
Gridworld

$S_5 = (2,3)$

$A_5 = down$

$R_6 = 10$

t = 5

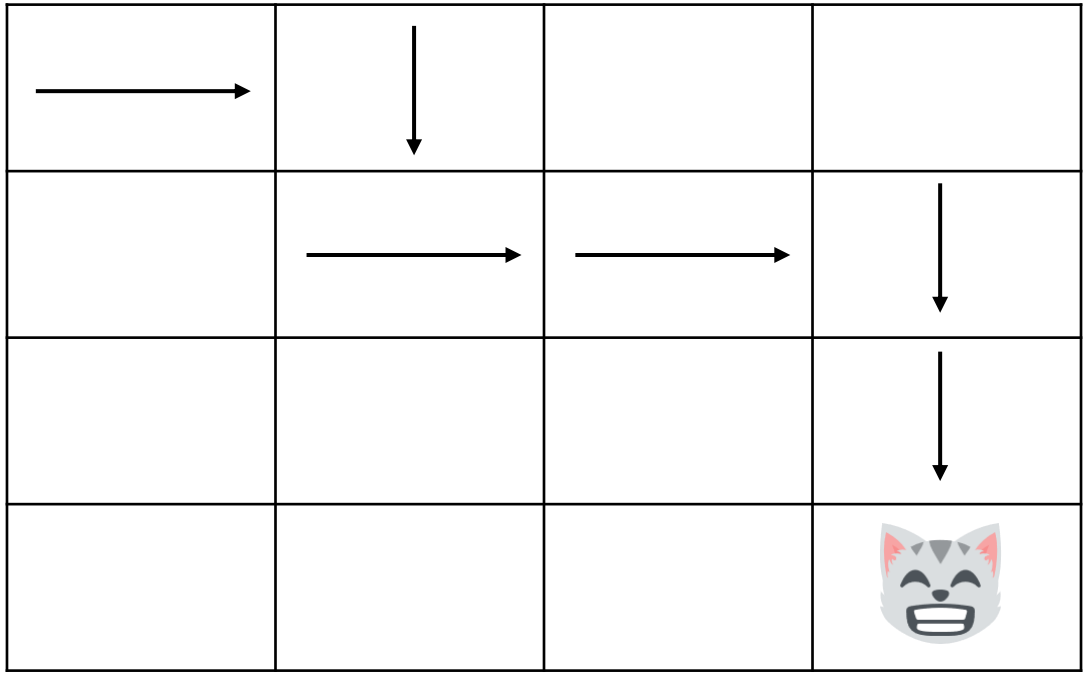


Gridworld

t = 6

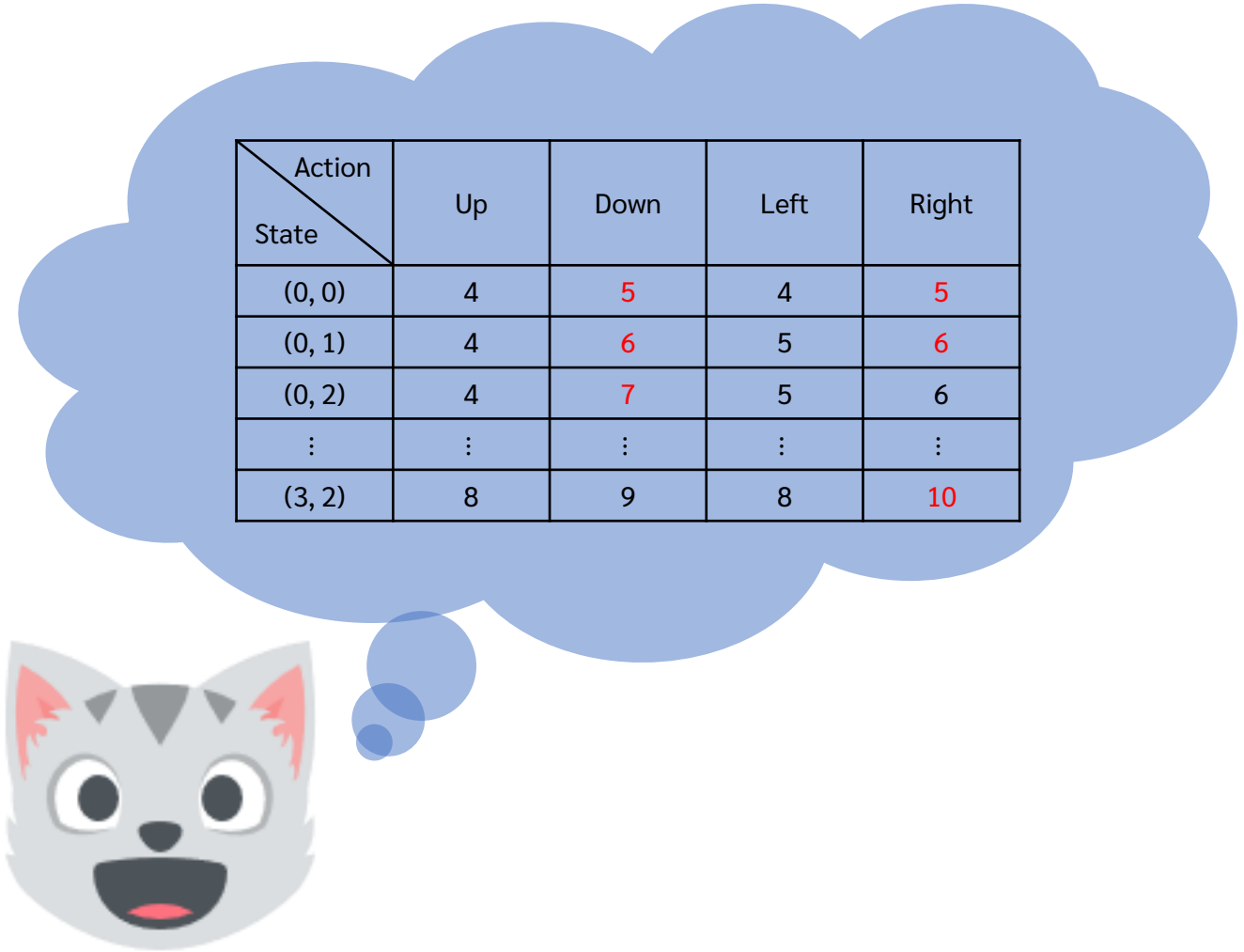
$S_6 = (3,3)$

$Teminal\ state$



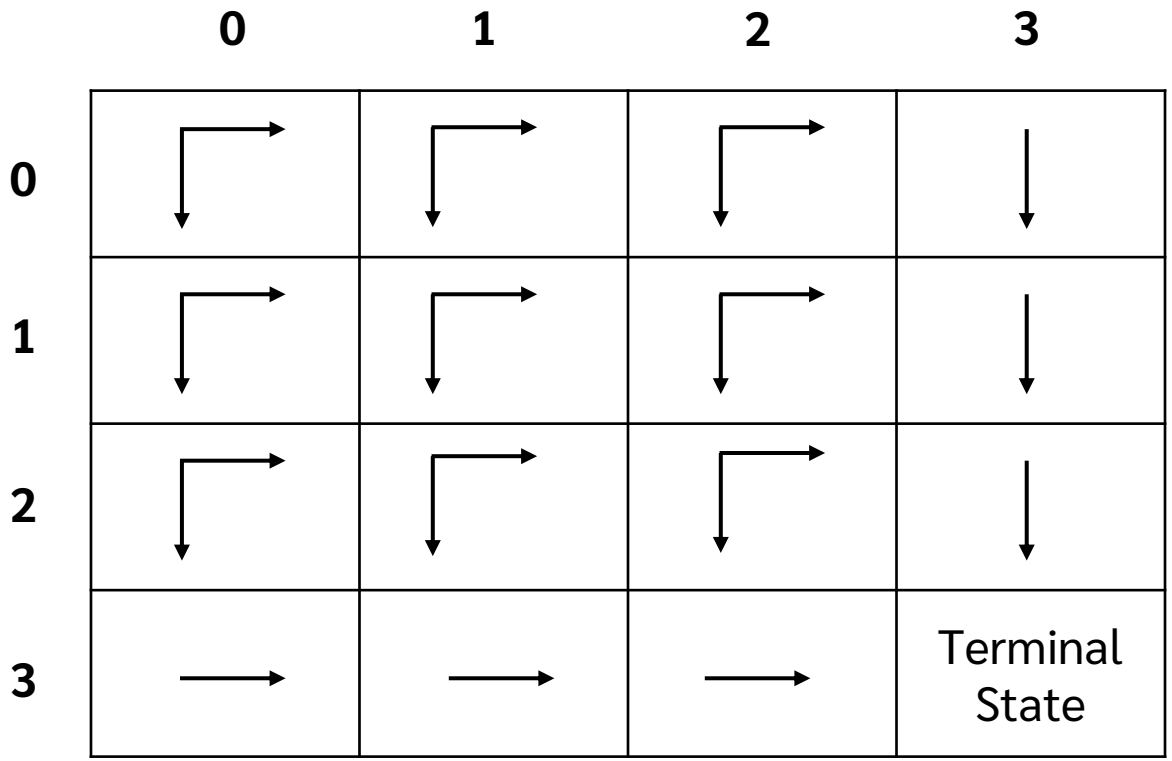
$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, S_3, A_3, R_4, S_4, A_4, R_5, S_5, A_5, R_6, S_6$$

Why is agent so smart ?



State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Policy



Q table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

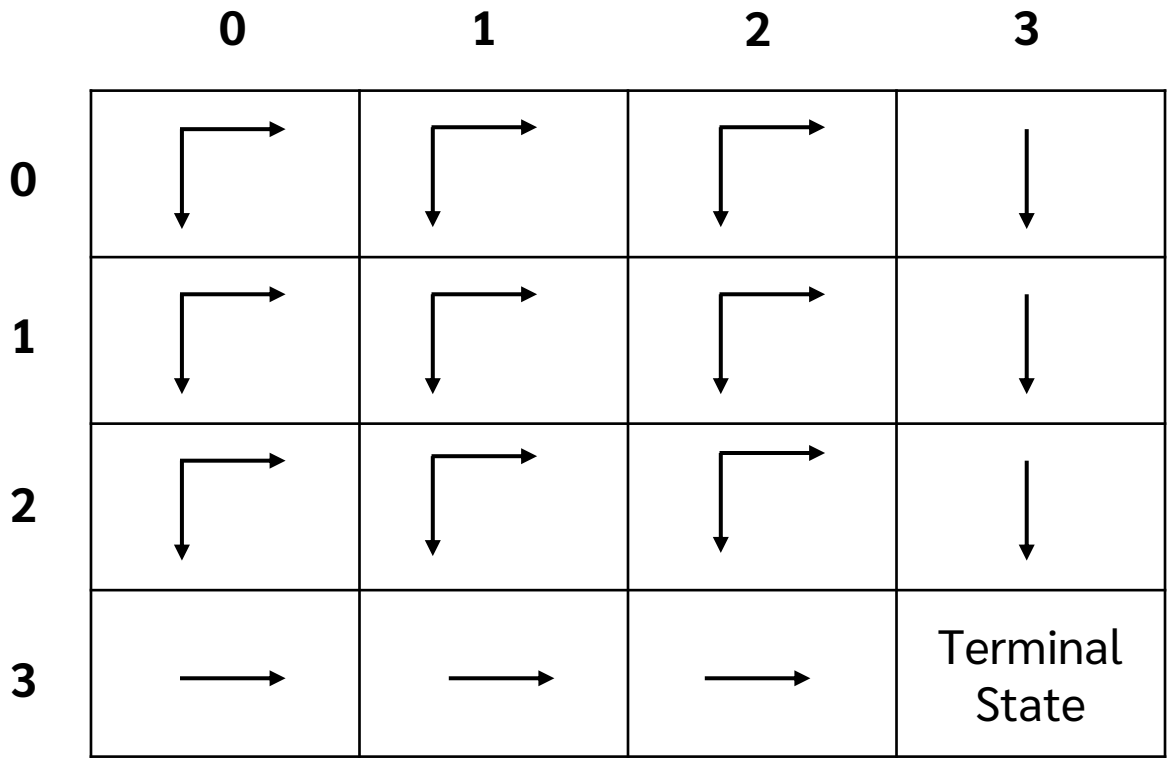
Q table

	0	1	2	3
0	4 4 5 5	4 5 6 6	4 5 7 7	4 5 6 7
1	4 5 6 6	5 5 7 7	6 6 8 8	7 7 8 9
2	5 6 7 7	6 6 8 8	7 7 9 9	8 8 9 10
3	6 7 8 7	7 7 9 8	8 8 10 9	Terminal State

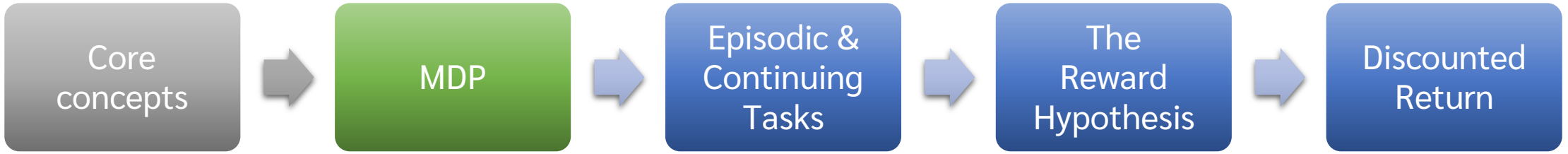
Q table

	0	1	2	3
0	4 4 5 5	4 5 6 6	4 5 7 7	4 5 6 7
1	4 5 6 6	5 5 7 7	6 6 8 8	7 7 8 9
2	5 6 7 7	6 6 8 8	7 7 9 9	8 8 9 10
3	6 7 8 7	7 7 9 8	8 8 10 9	Terminal State

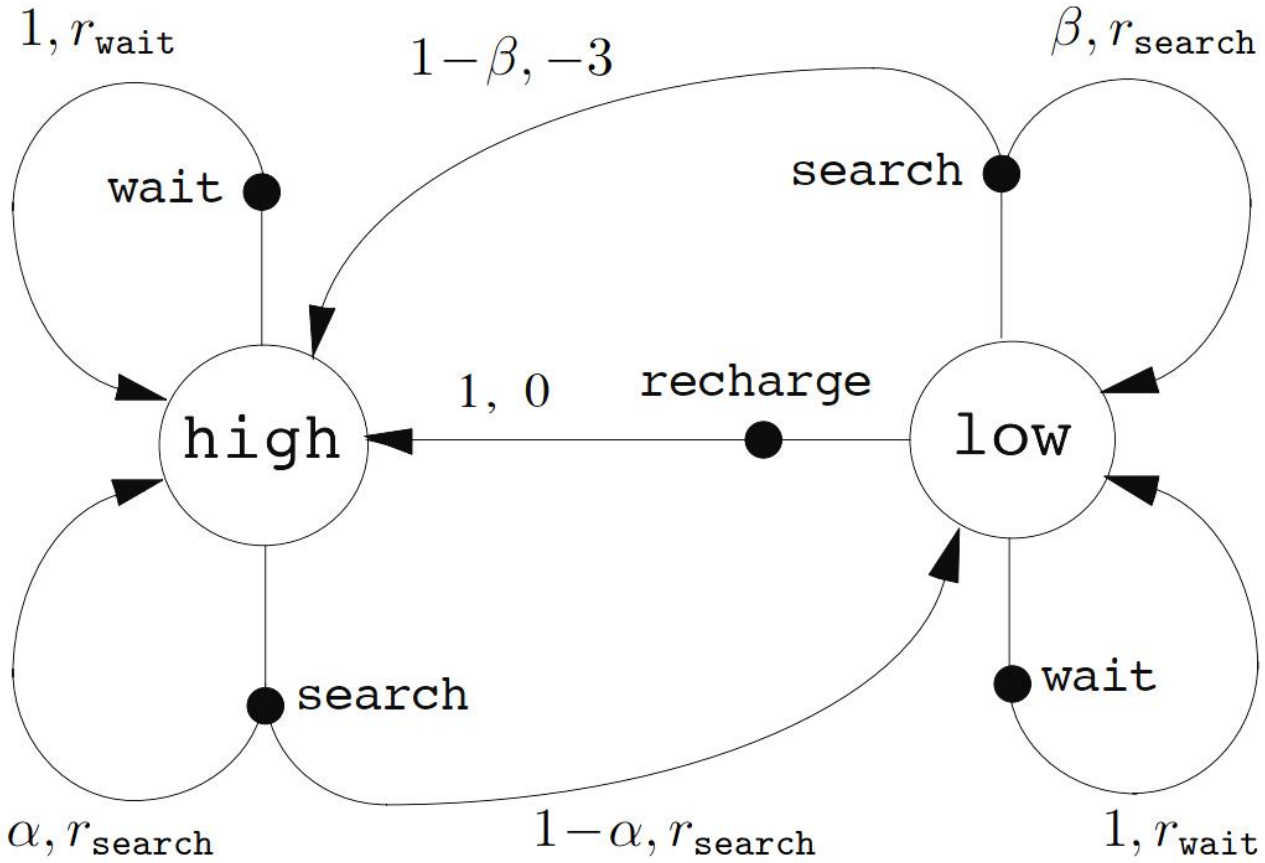
Policy



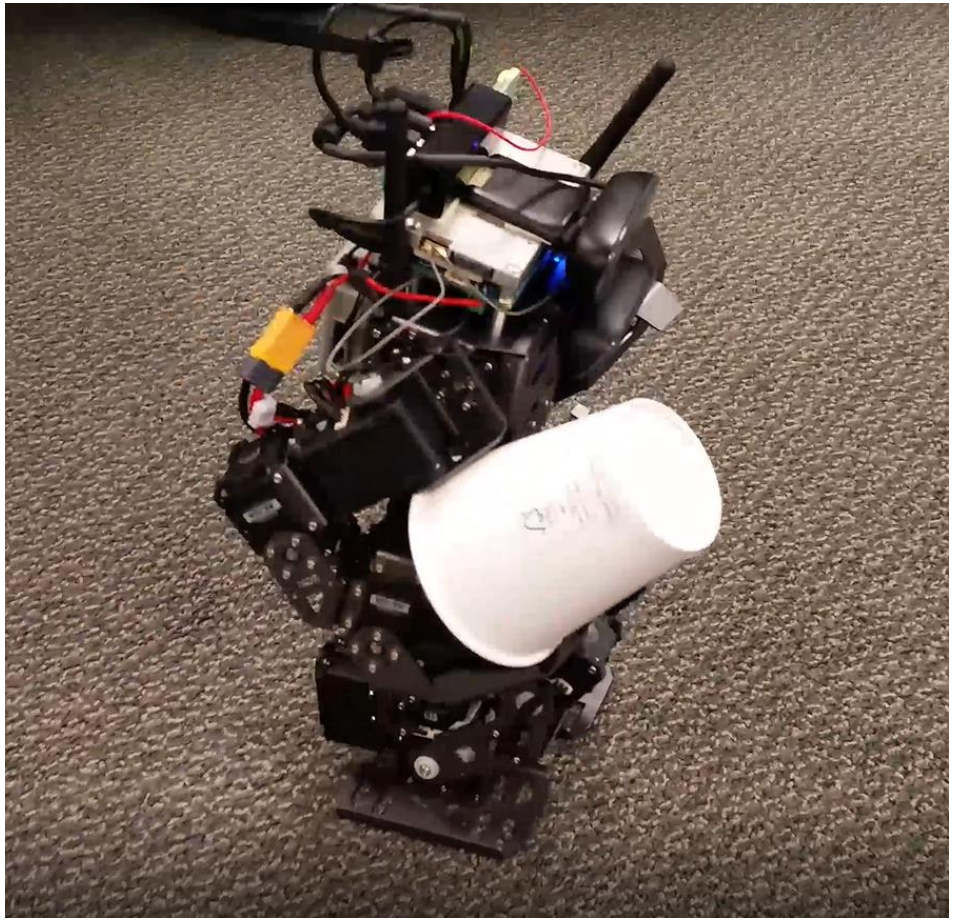
Contents


Core concepts
MDP
Episodic & Continuing Tasks
The Reward Hypothesis
Discounted Return

Markov Decision Process

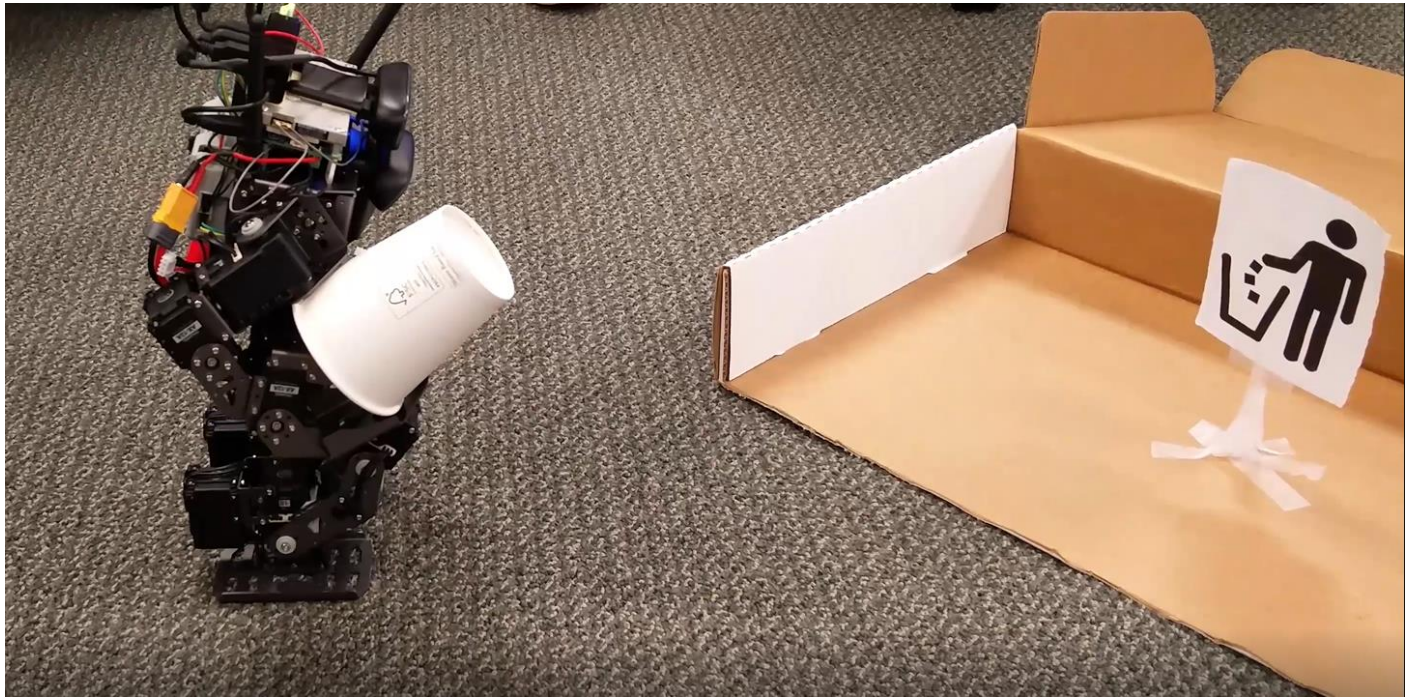


Recycling Robot



reference: https://news.developer.nvidia.com/intelligent-trash-pick-up-robots-coming-to-an-office-near-you/

Recycling Robot

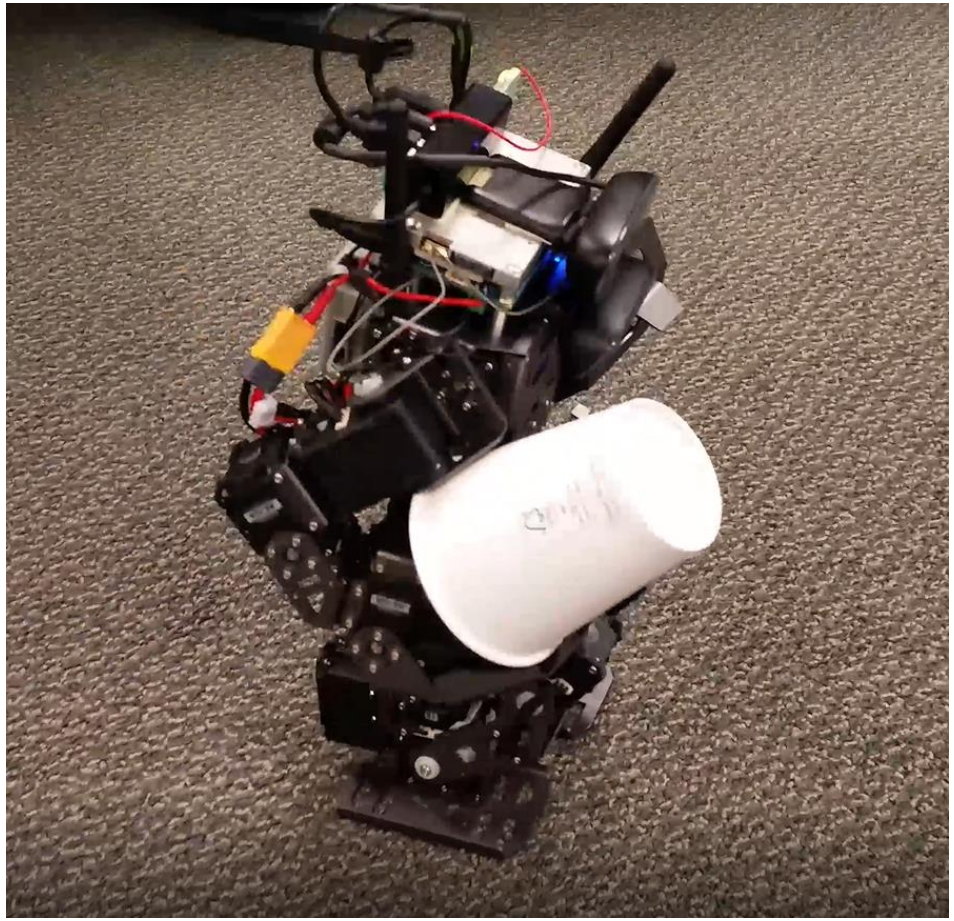


reference: https://news.developer.nvidia.com/intelligent-trash-pick-up-robots-coming-to-an-office-near-you/

Recycling Robot

$\mathcal{S} = \{ high, low \}$

$\mathcal{A} = \{ search, recharge, wait \}$



reference: https://news.developer.nvidia.com/intelligent-trash-pick-up-robots-coming-to-an-office-near-you/

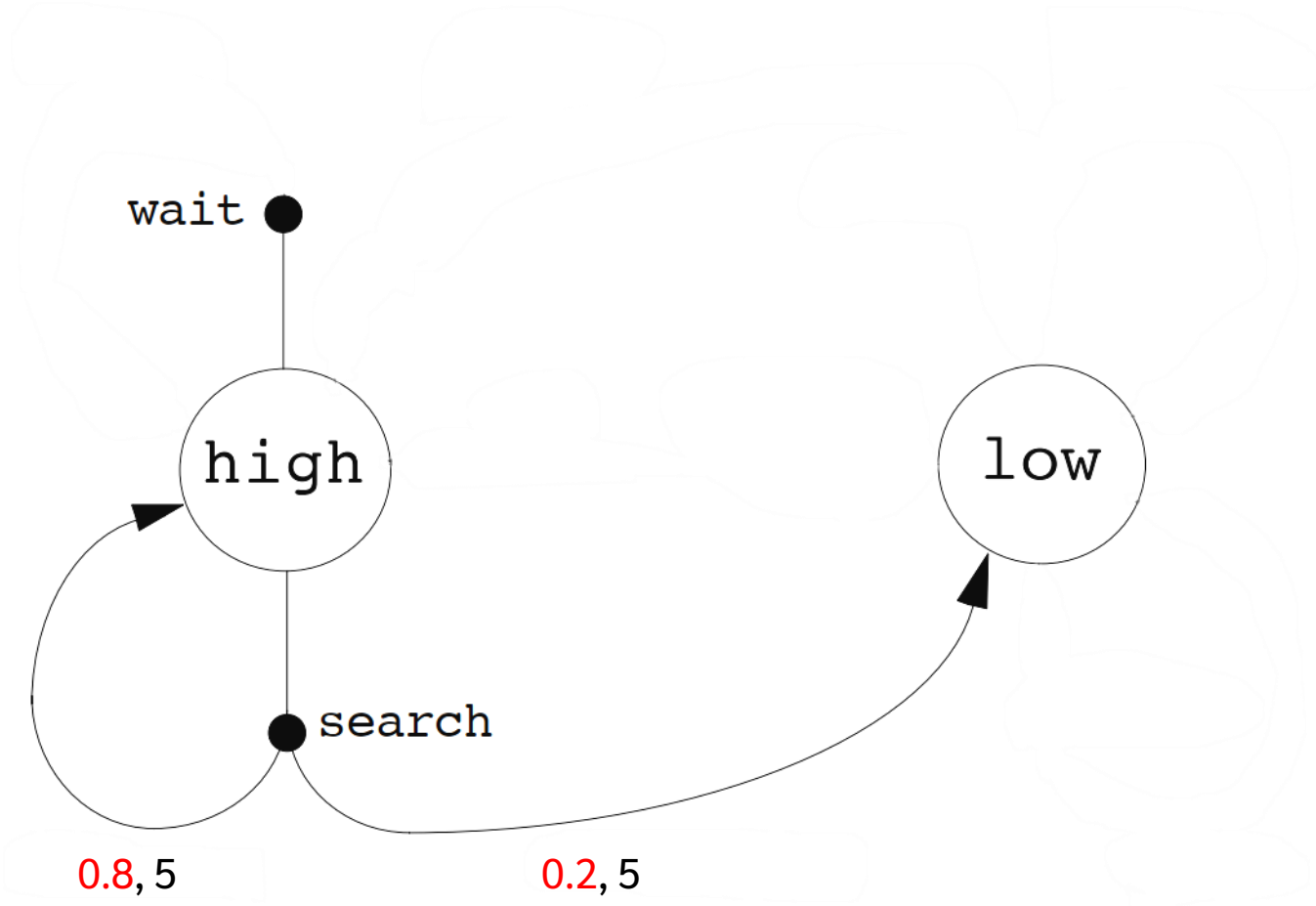
Recycling Robot

reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot

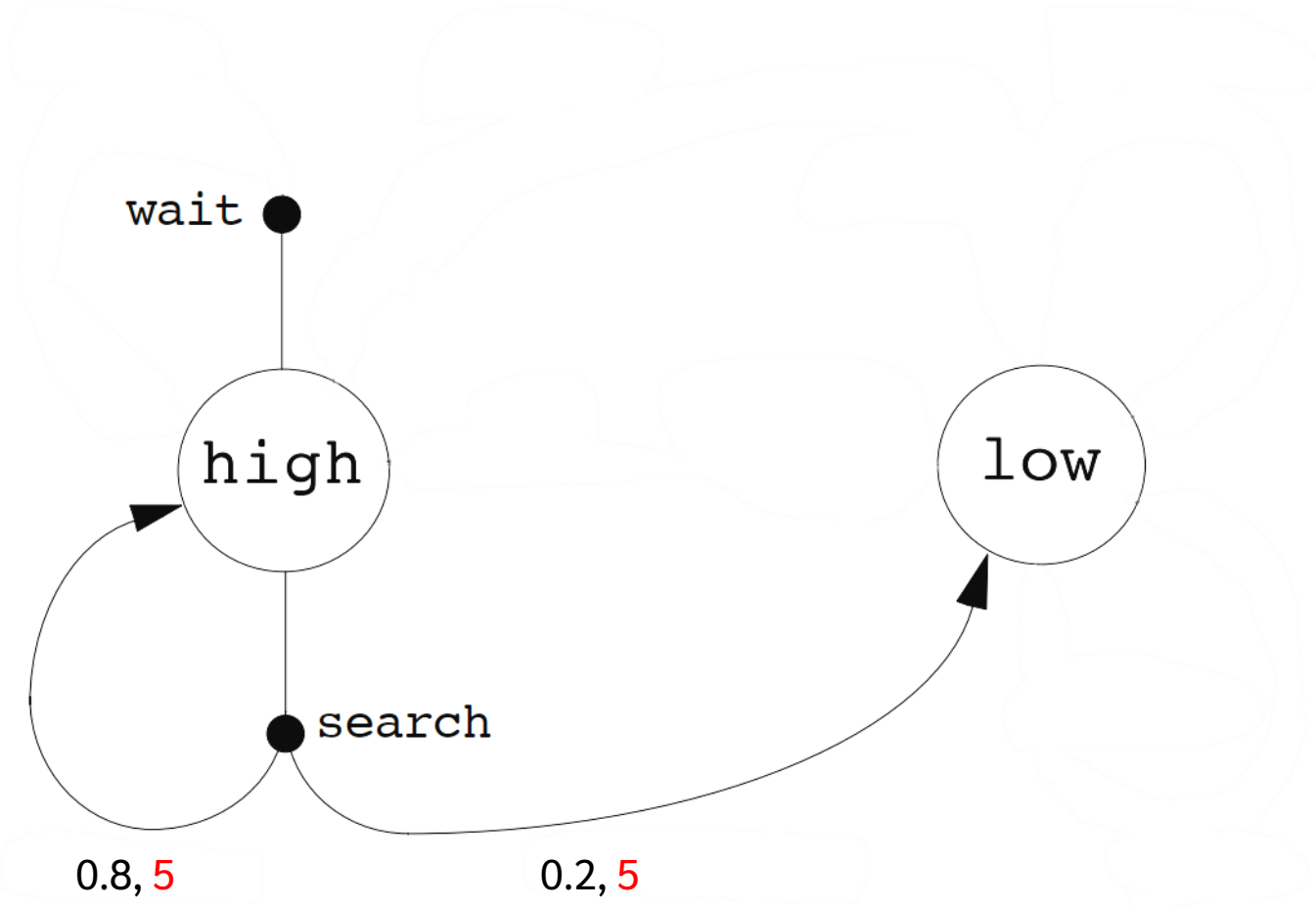
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



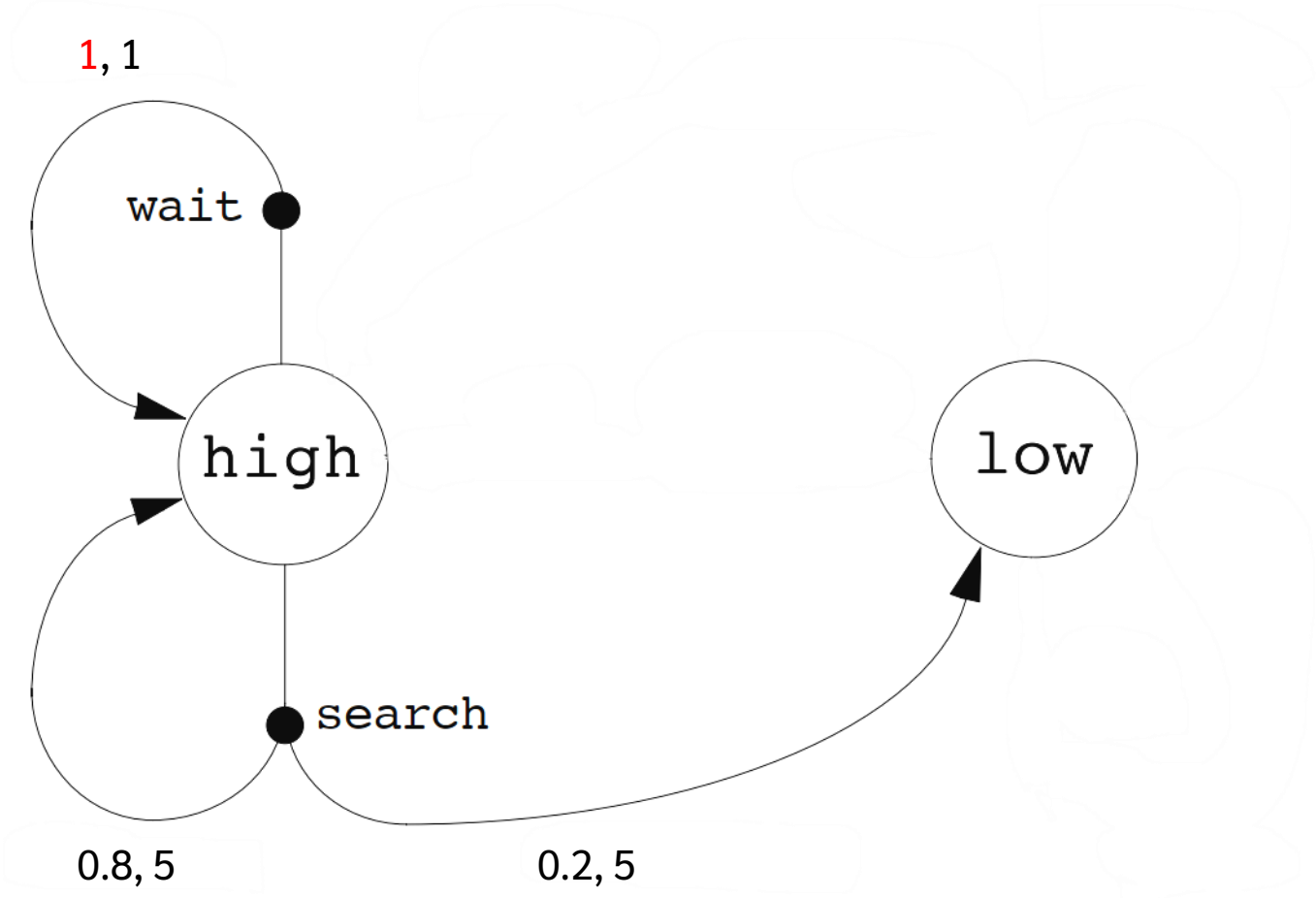
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



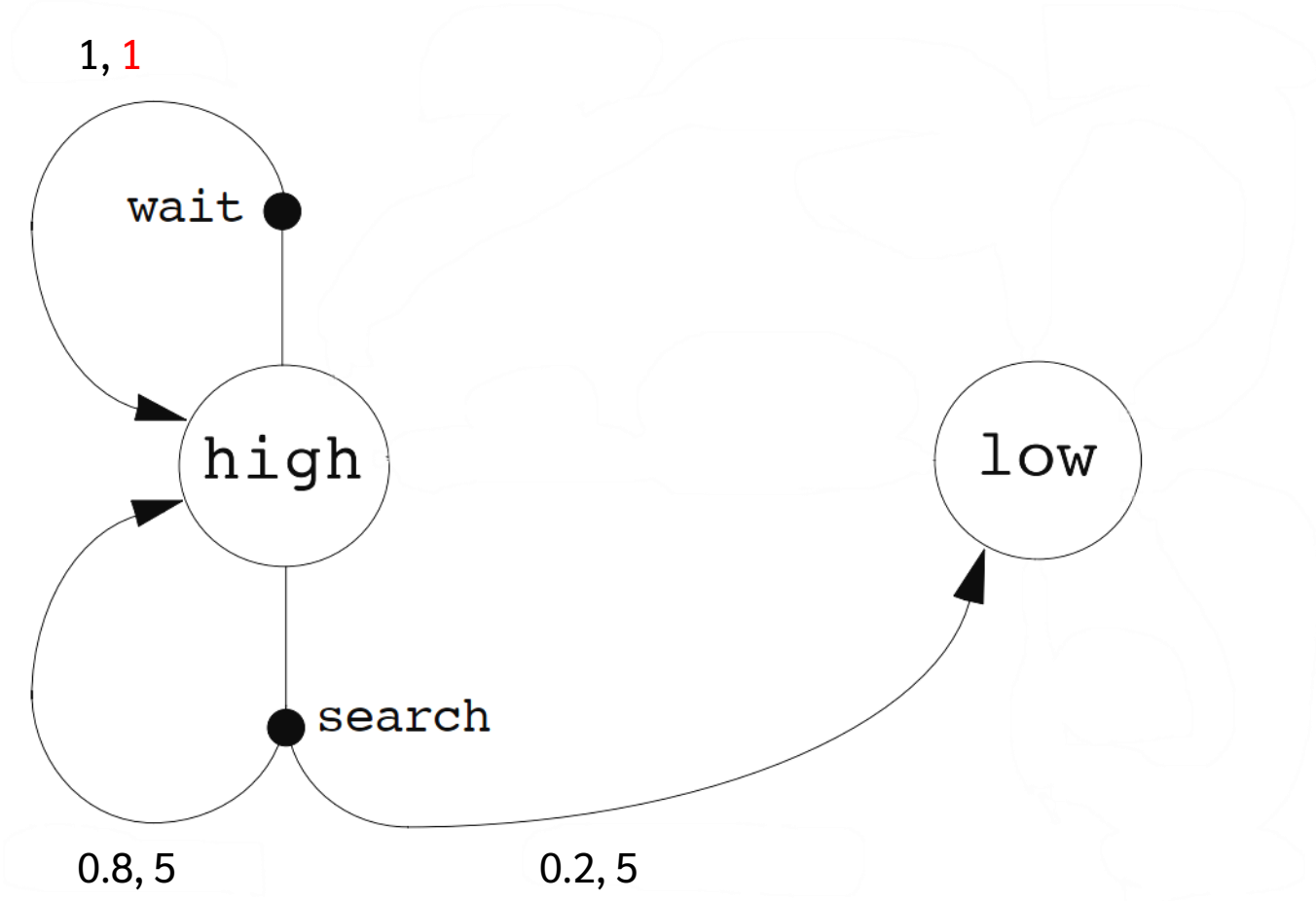
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



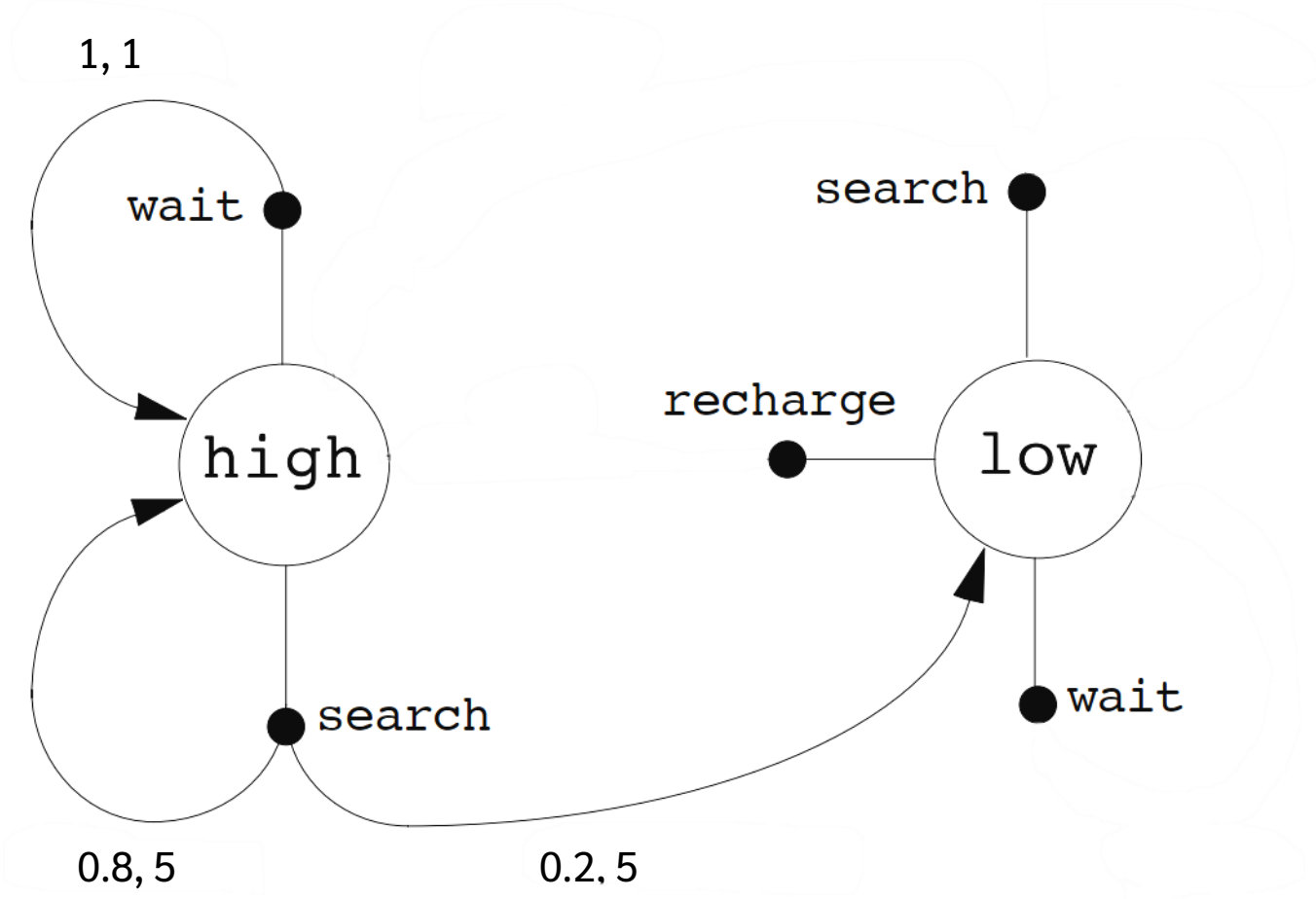
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



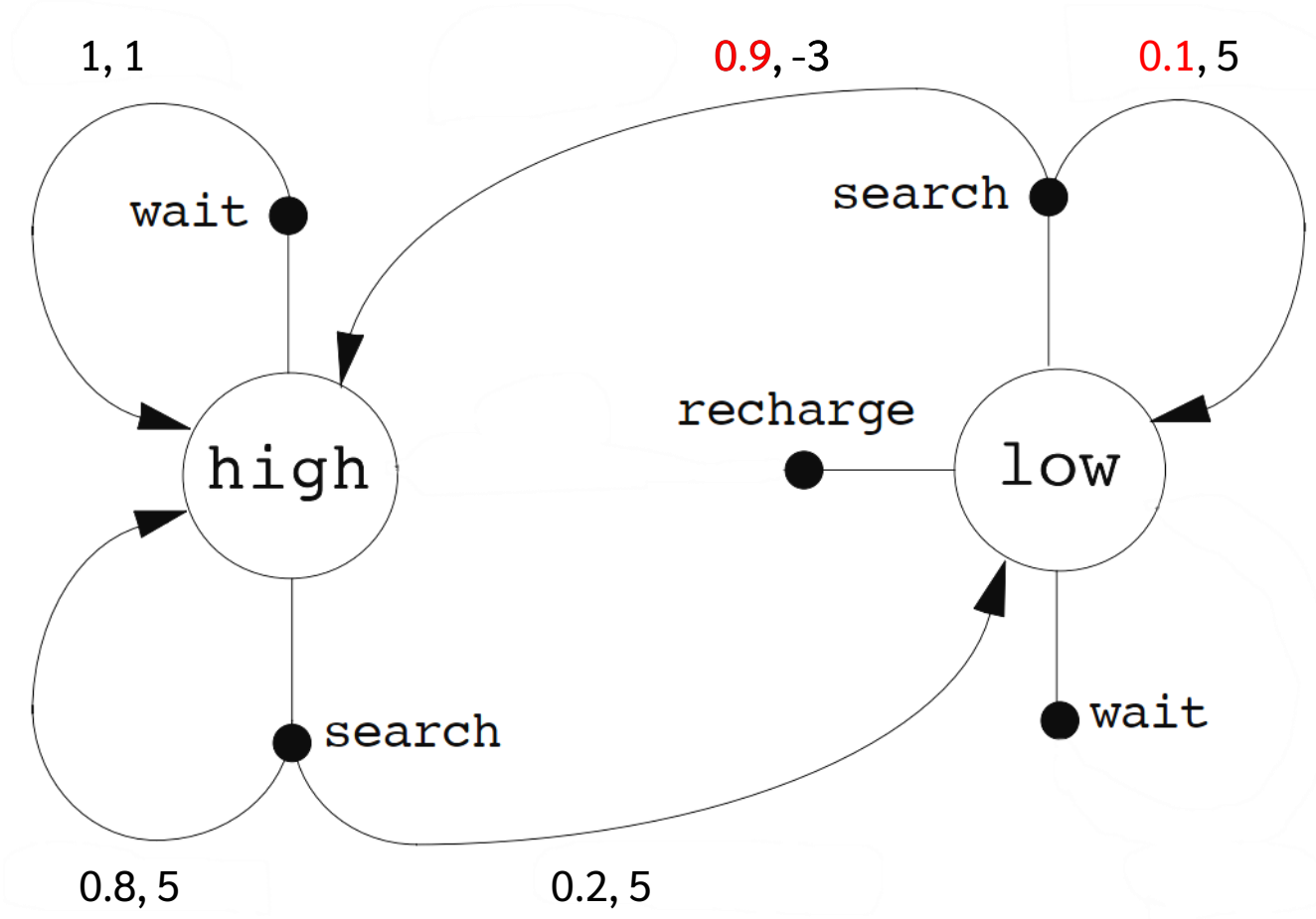
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



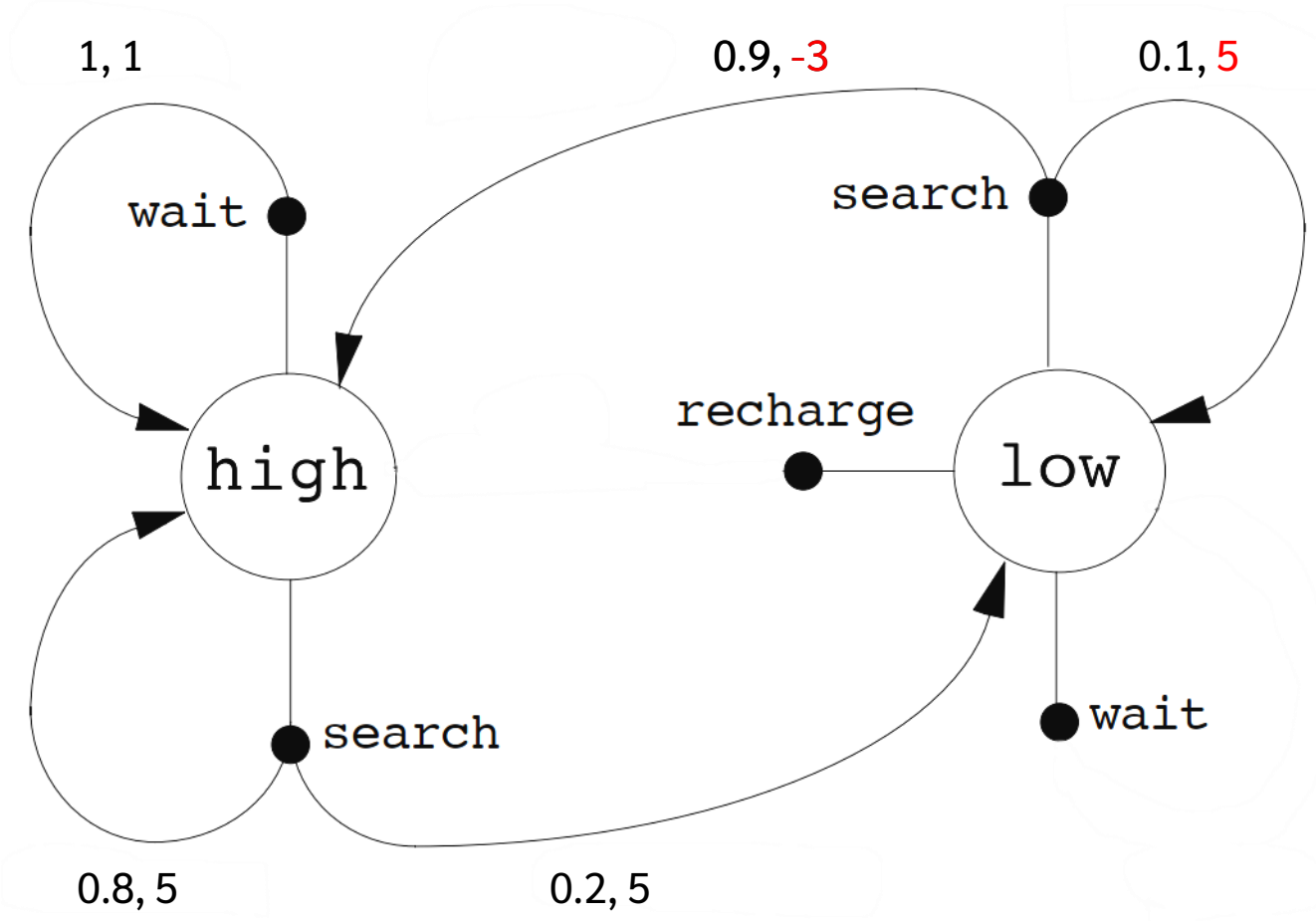
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



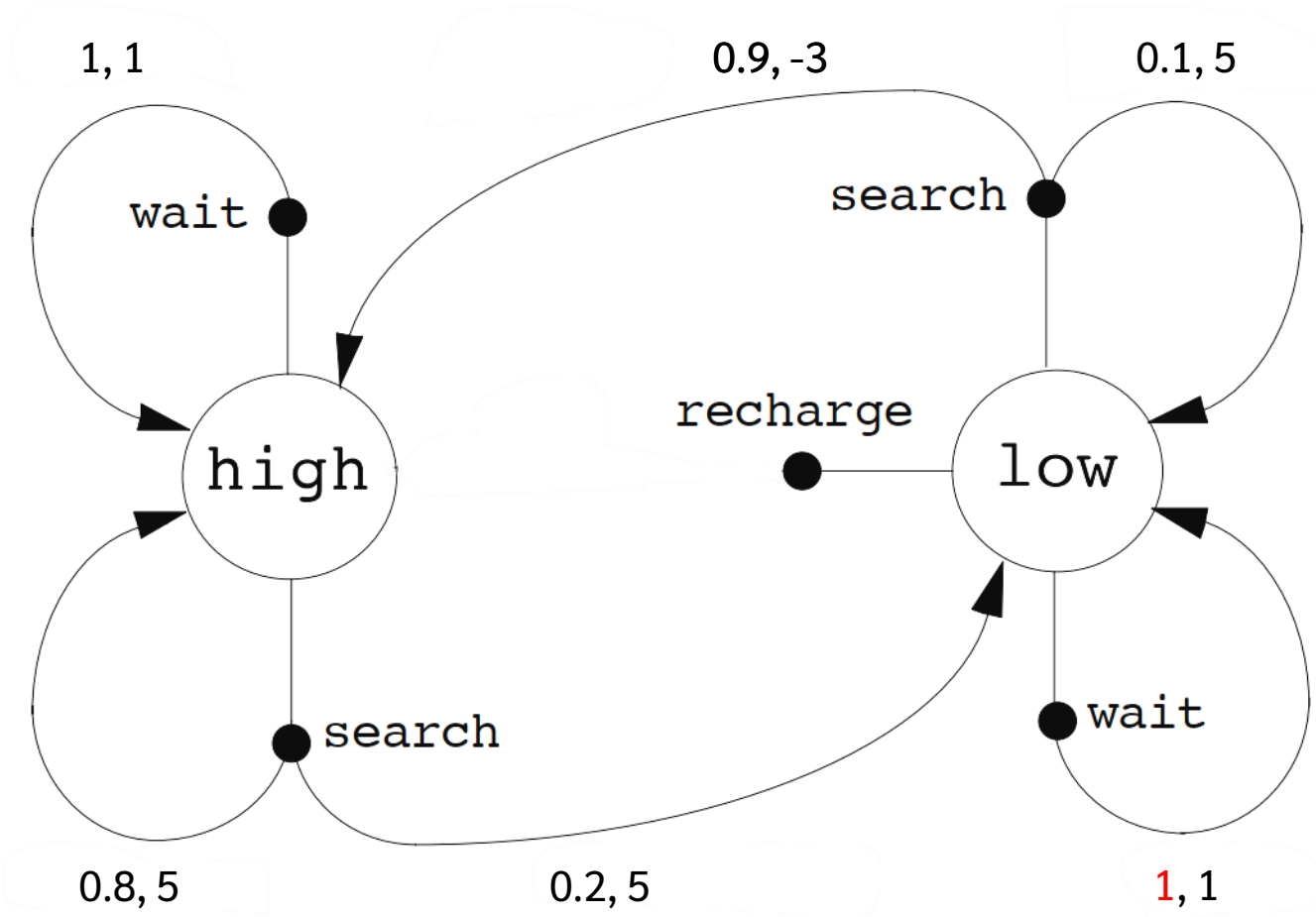
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



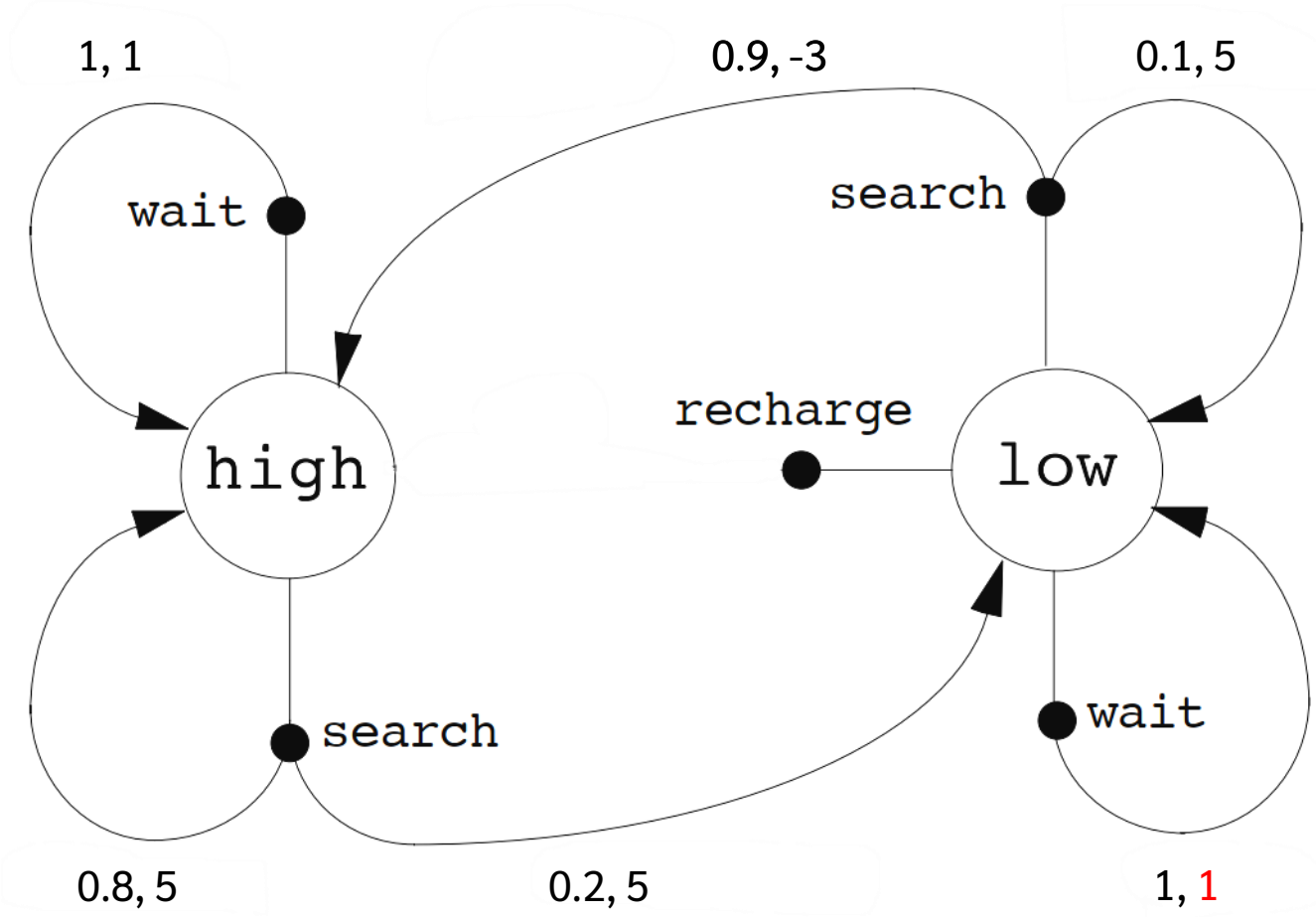
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



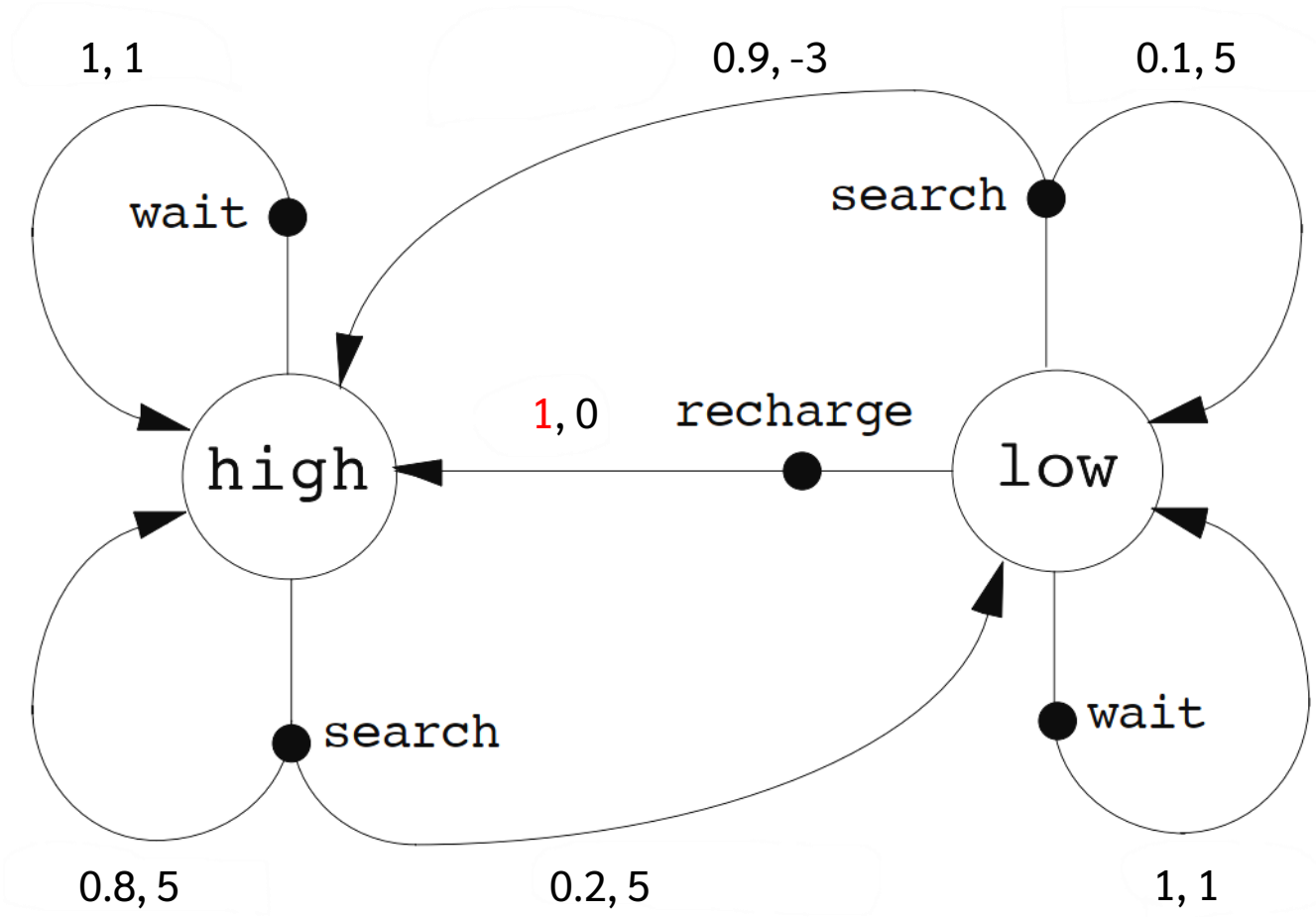
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



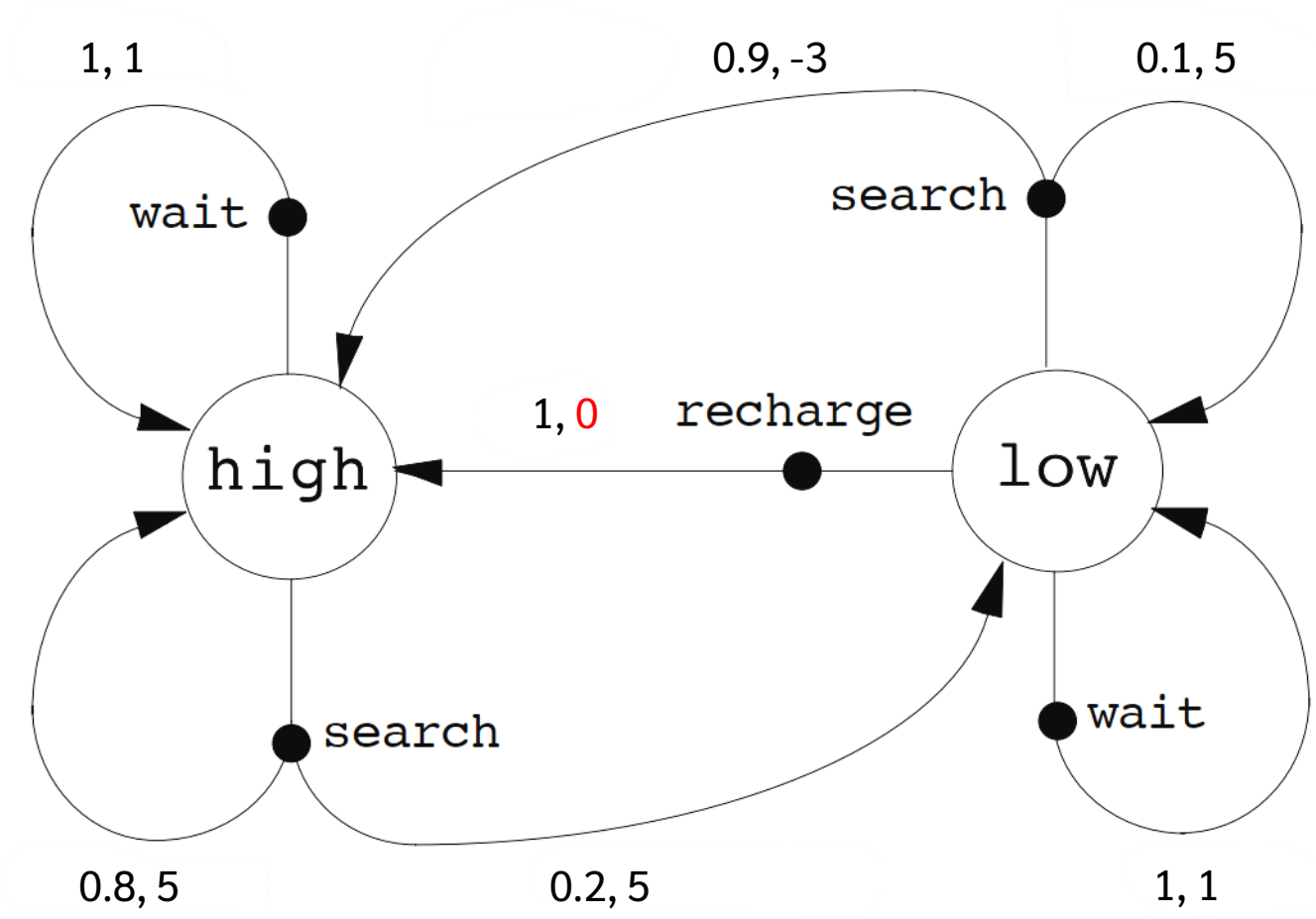
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



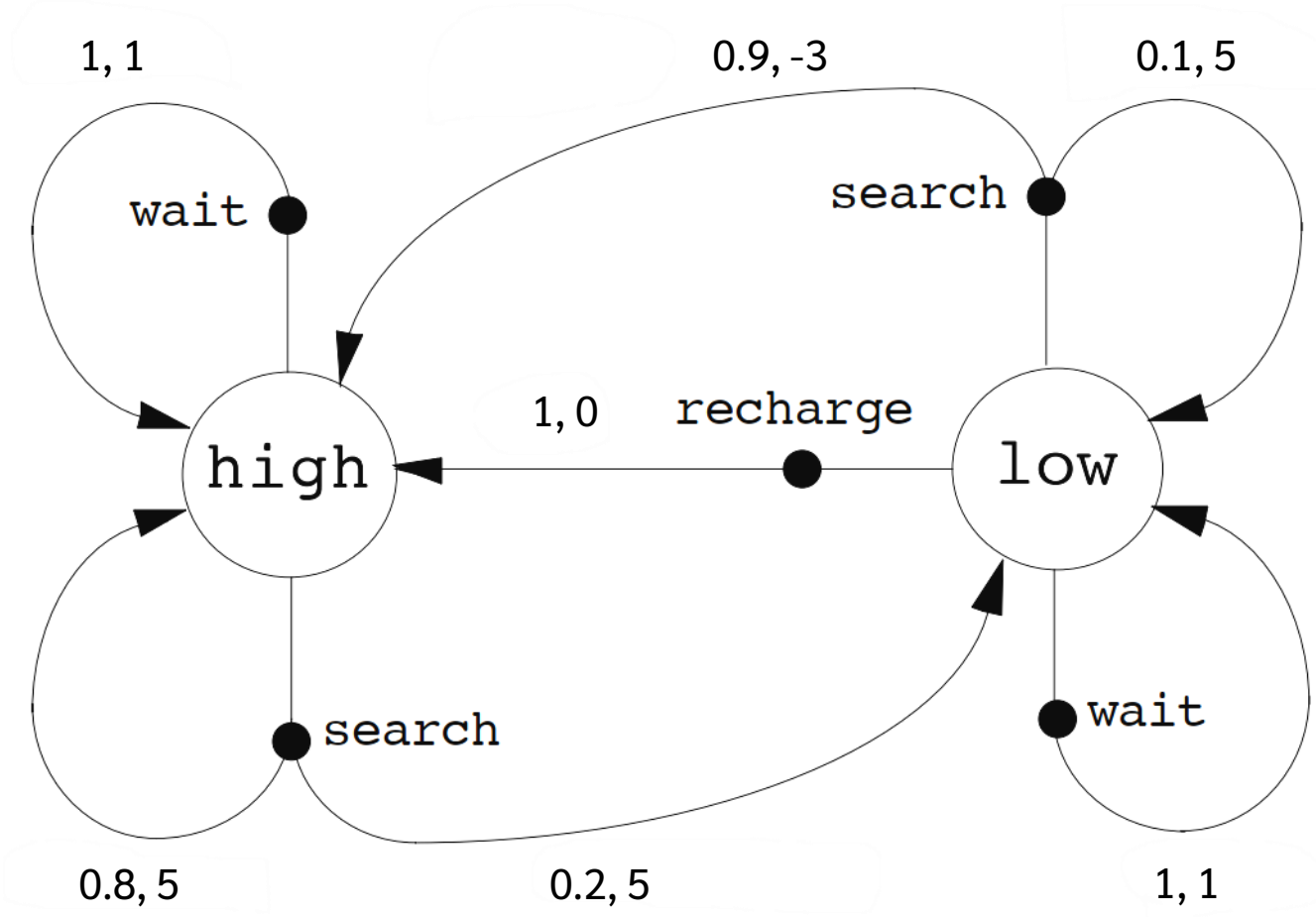
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot



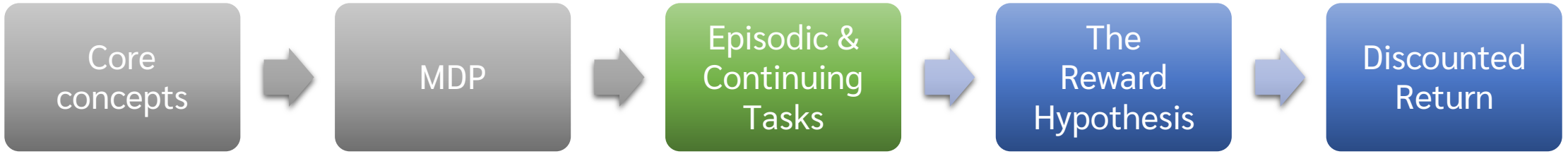
reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Recycling Robot

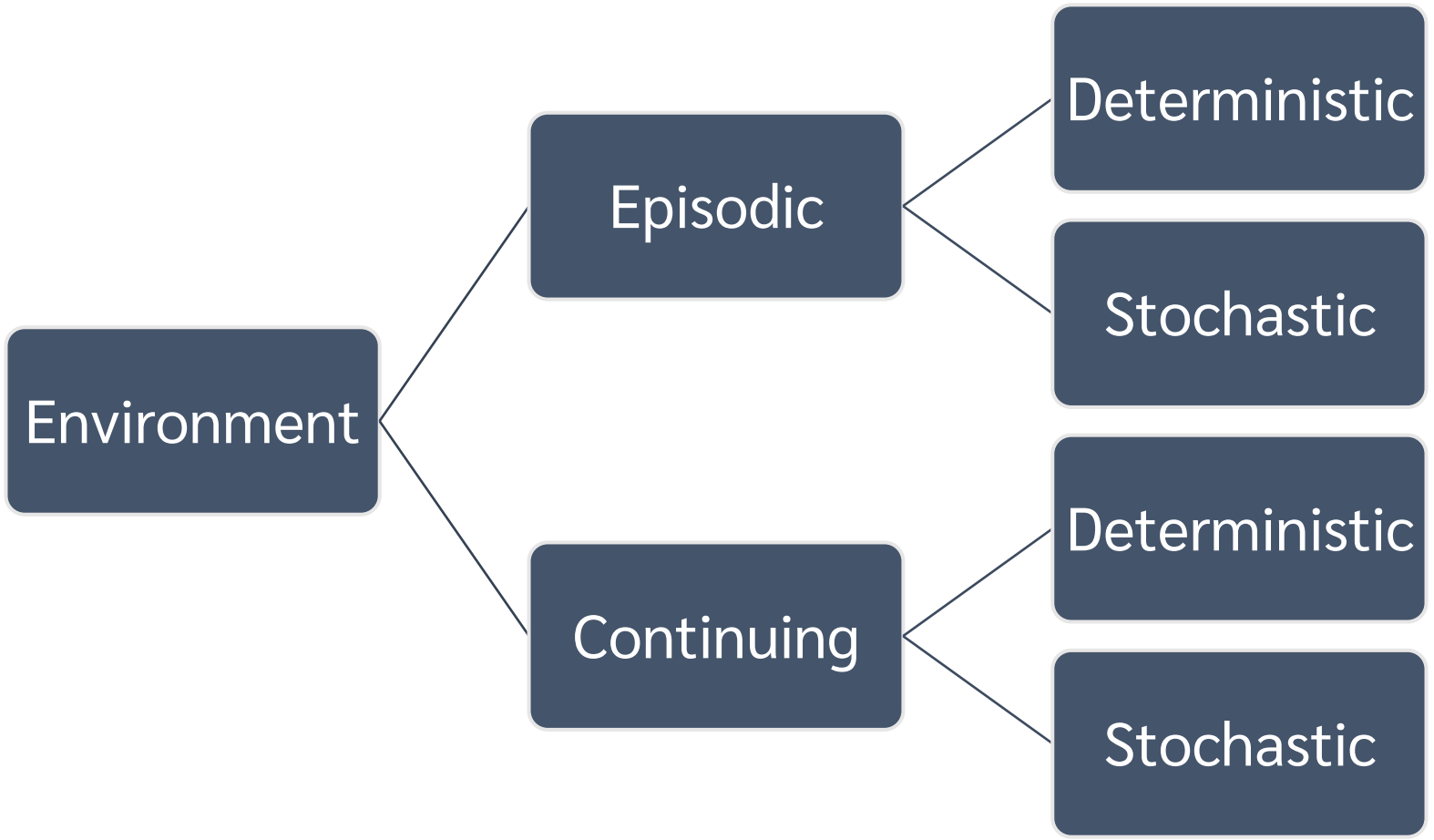


reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Contents


Core concepts
MDP
Episodic & Continuing Tasks
The Reward Hypothesis
Discounted Return

Environment



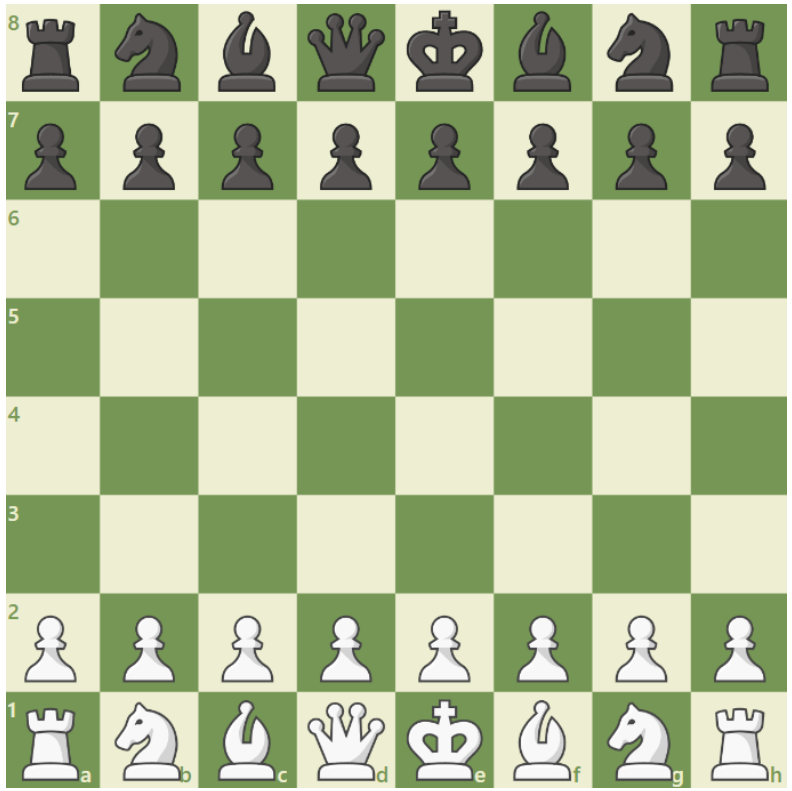
Episodic Task

$$S_0, A_0, R_1, S_1, A_1, \ldots, R_T, S_T$$

Chess: Deterministic Episodic task

reference: https://www.chess.com/

Chess: Deterministic Episodic task

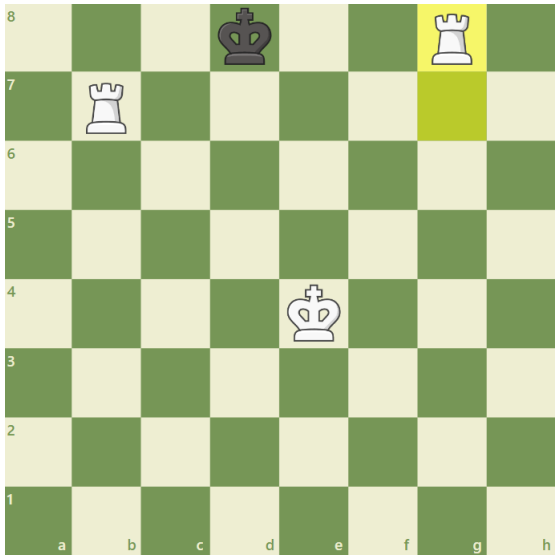


reference: https://www.chess.com/

Blackjack: Stochastic Episodic task

reference: https://www.247blackjack.com/

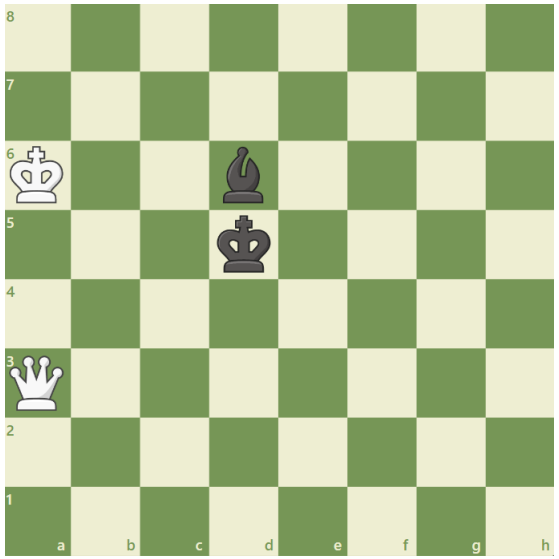
Episode



Episode 1

White won

$S_0, A_0, R_1, S_1, A_1, \ldots, R_T, S_T$

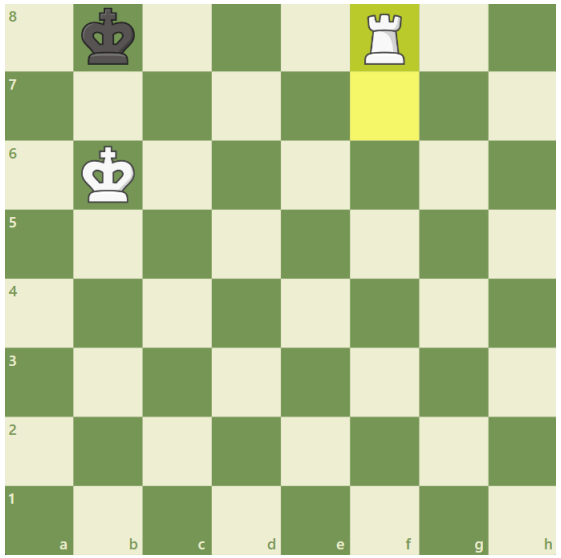


Episode 2

Draw

$S_0, A_0, R_1, S_1, A_1, \ldots, R_T, S_T$

...



Episode N

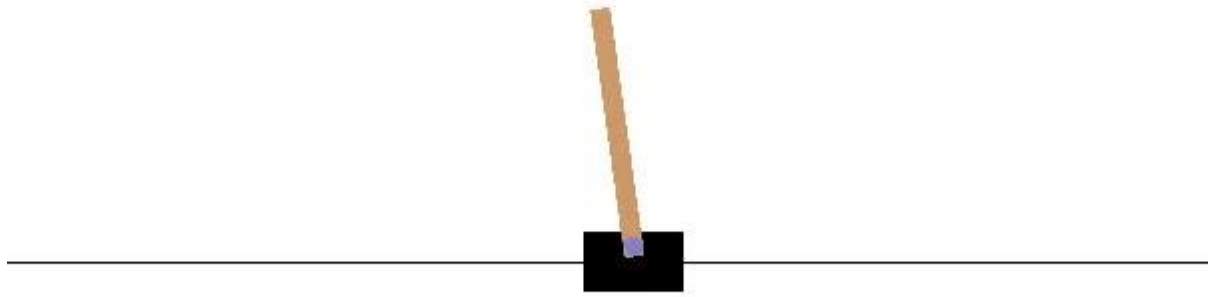
White won

$S_0, A_0, R_1, S_1, A_1, \ldots, R_T, S_T$

Continuing Task

$$S_0, A_0, R_1, S_1, A_1, \ldots$$

CartPole: Deterministic Continuing task



Stock trading: Stochastic Continuing task

Contents


Core concepts
MDP
Episodic & Continuing Tasks
The Reward Hypothesis
Discounted Return

Reward Hypothesis

That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

reference: Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Reward Hypothesis

การให้ reward กับ agent จะต้องสอดคล้องกับเป้าหมาย

Chess

reference: https://www.chess.com/

Chess

reference: https://www.chess.com/

เป้าหมาย: agent เล่น chess เก่งมาก ๆ

การให้ reward:

ชนะ reward +1

แพ้ reward –1

เสมอ reward 0

Chess

reference: https://www.chess.com/

เป้าหมาย: agent เล่น chess เก่งมาก ๆ
และ ชนะเร็วที่สุด

การให้ reward:

ทุก time step ให้ reward -1

ชนะ reward +100

แพ้ reward –100

เสมอ reward 0

Chess

reference: https://www.chess.com/

เป้าหมาย: agent เล่น chess เก่งมาก ๆ
และ เมื่อจบเกมต้องการให้
ตัวหมากฝั่งเราเหลือเยอะสุด

การให้ reward:

ทุกครั้งที่เสียตัวหมากไป reward -1
ชนะ reward +100
แพ้ reward –100
เสมอ reward 0

Contents


Core concepts
MDP
Episodic & Continuing Tasks
The Reward Hypothesis
Discounted Return

Return

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

Discounted Return

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

Discount rate $\gamma \in [0,1]$

Discounted Return

$\gamma = 1$

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$

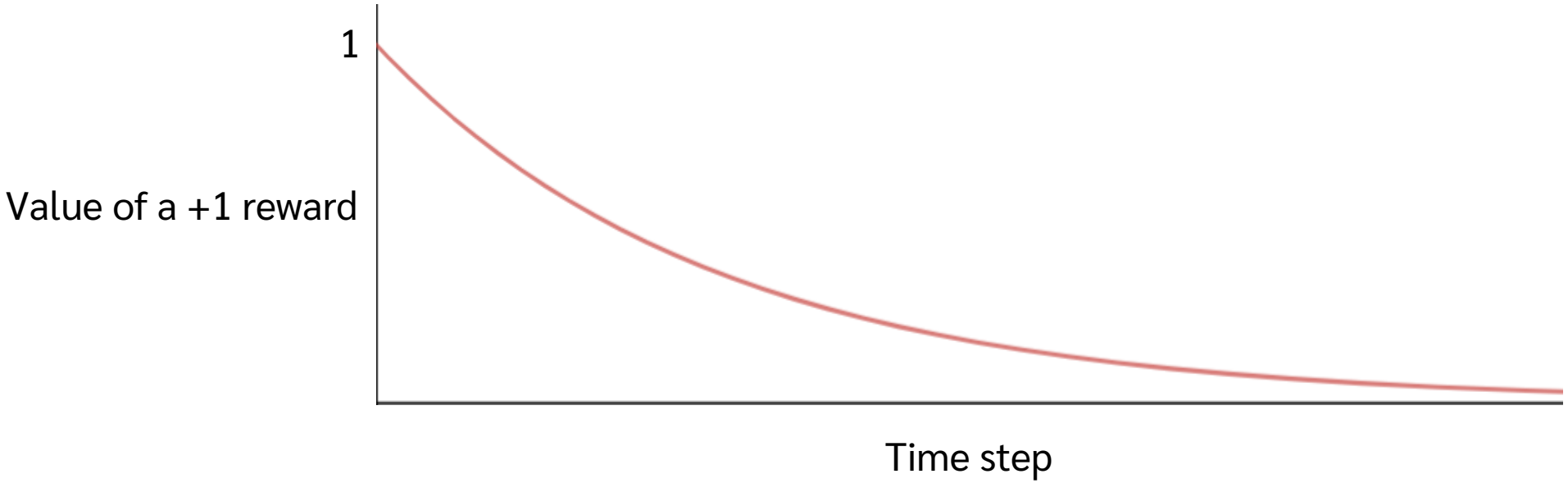
$\gamma = 0$

$$G_t = R_{t+1}$$

Discounted Return

$\gamma = 0.95$

$$G_t = R_{t+1} + (0.95)R_{t+2} + (0.90)R_{t+3} + \cdots$$



คำถาม: $\gamma = 1$ ดีที่สุดใช่หรือไม่

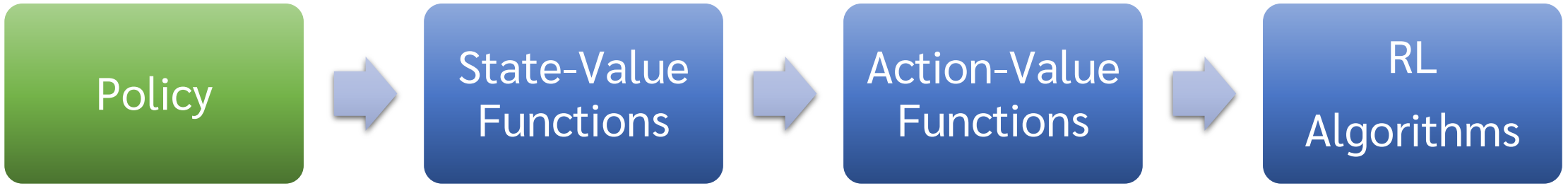
คำถาม: $\gamma = 1$ ดีที่สุดใช่หรือไม่

คำตอบ: ไม่ใช่

ค่า γ เท่าไหร่ดีนั้นขึ้นกับปัญหาที่เราจะแก้

Reinforcement learning Algorithms

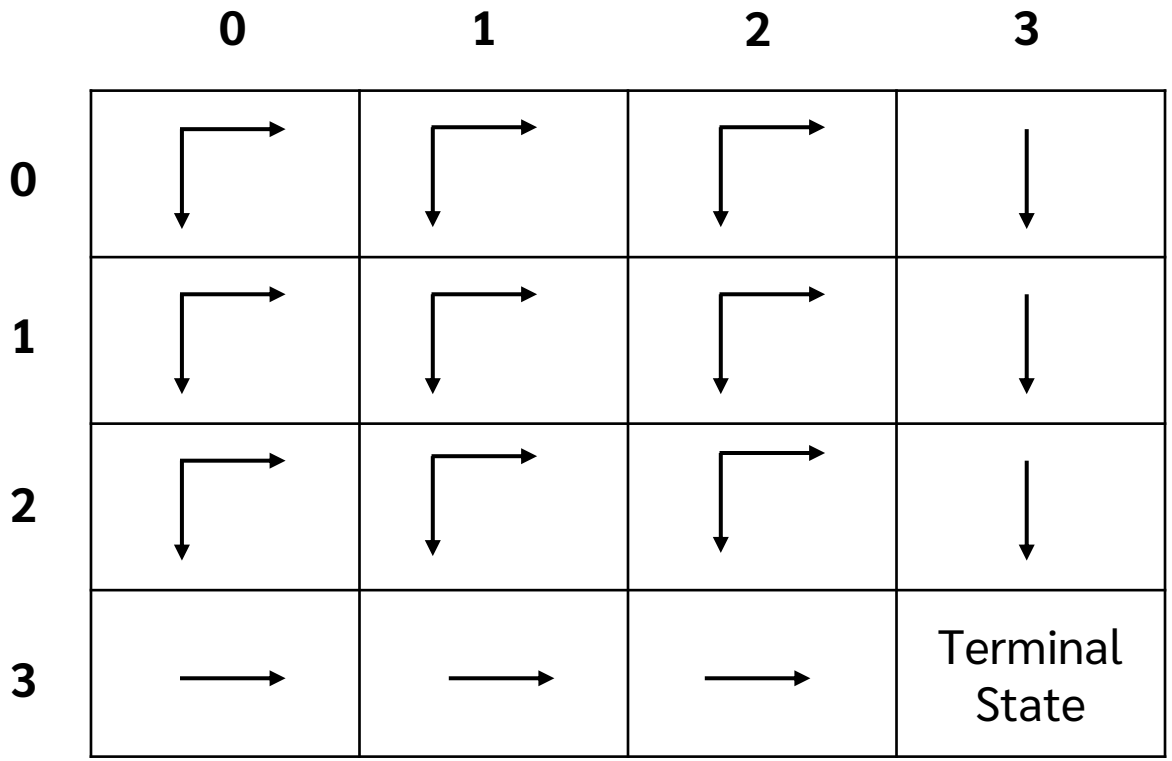
Contents


Policy
State-Value Functions
Action-Value Functions
RL Algorithms

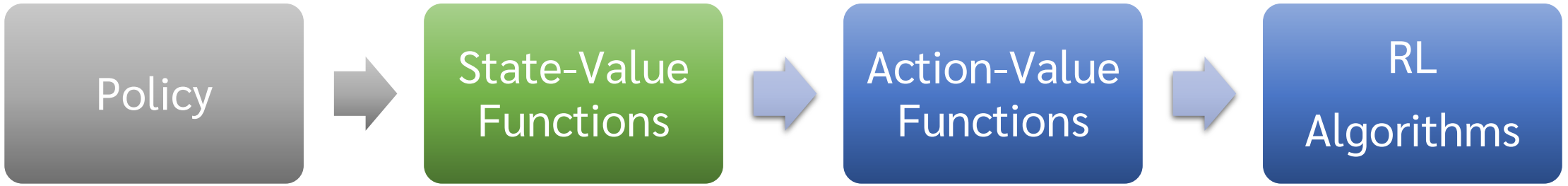
Policy

$$\pi : state \rightarrow action$$

Policy



Contents


Policy
State-Value Functions
Action-Value Functions
RL Algorithms

State-Value Functions

ค่าของ state s จาก policy π คือ

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

State-Value Functions

ค่าของ state s จาก policy π คือ

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

ในแต่ละ state s

State-Value Functions

ค่าของ state s จาก policy π คือ

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

ในแต่ละ state s

จะมีค่า expected return

State-Value Functions

ค่าของ state s จาก policy π คือ

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

ในแต่ละ state s

จะมีค่า expected return

ถ้า agent เริ่มต้นที่ state s

State-Value Functions

ค่าของ state s จาก policy π คือ

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

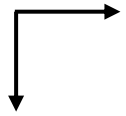
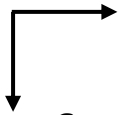
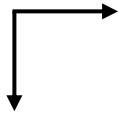
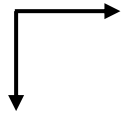
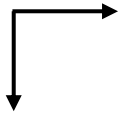
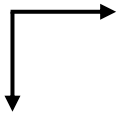
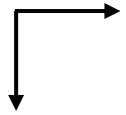
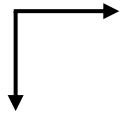
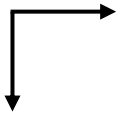
ในแต่ละ state s

จะมีค่า expected return

จากการใช้ policy ในการเลือก action ในทุก ๆ time step

ถ้า agent เริ่มต้นที่ state s

Example

	0	1	2	3
0	 5	 6	 7	 8
1	 6	 7	 8	 9
2	 7	 8	 9	 10
3	 8	 9	 10	Terminal State

Example

	0	1	2	3
0				
1				
2				
3				Terminal State

Agent อยู่ ณ state (2, 3)

Agent ตัดสินใจ take action:

1. Down
2. Left → Down → Right

คำถามคือ

Agent ควรเลือก action ด้วยวิธีใด

Example

คำตอบคือ เลือก action ที่ทำให้ Expected return สูงสุด

Example

คำตอบคือ เลือก action ที่ทำให้ Expected return สูงสุด

สมมติ ให้ $\gamma=1$

Example

คำตอบคือ เลือก action ที่ทำให้ Expected return สูงสุด

สมมติ ให้ $\gamma=1$

Agent ตัดสินใจ take action:

1. Down

   Expected return = 10

Example

คำตอบคือ เลือก action ที่ทำให้ Expected return สูงสุด

สมมติ ให้ $\gamma=1$

Agent ตัดสินใจ take action:

1. Down

   Expected return = 10

2. Left Down Right

   Expected return = [(-1) + (-1) + 10] / 3 = 8 / 3 = 2.67

Example

คำตอบคือ เลือก action ที่ทำให้ Expected return สูงสุด

สมมติ ให้ $\gamma=1$

Agent ตัดสินใจ take action:

1. Down

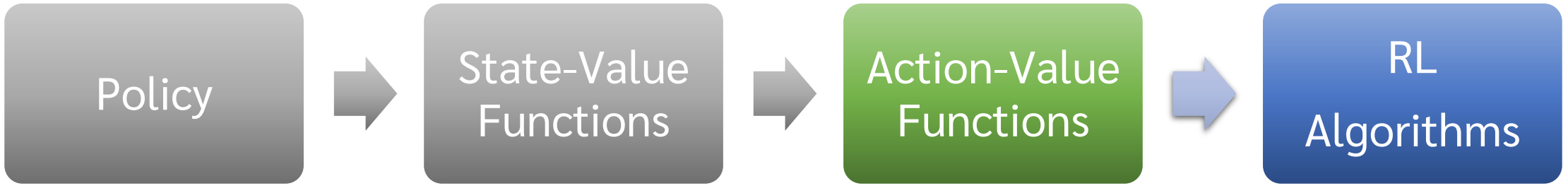
   Expected return = 10

2. Left Down Right

   Expected return = [(-1) + (-1) + 10] / 3 = 8 / 3 = 2.67

เลือก action 1 เพราะให้ค่า Expected return สูงสุด

Contents


Policy
State-Value Functions
Action-Value Functions
RL Algorithms

Action-Value Functions

ค่าของการเลือกออก action a ใน state s จาก policy π คือ

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

Action-Value Functions

ค่าของการเลือกออก action a ใน state s จาก policy π คือ

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

ในแต่ละ state s และ action a

Action-Value Functions

ค่าของการเลือกออก action a ใน state s จาก policy π คือ

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

ในแต่ละ state s และ action a

จะมีค่า expected return

Action-Value Functions

ค่าของการเลือกออก action a ใน state s จาก policy π คือ

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

ในแต่ละ state s และ action a

จะมีค่า expected return

ถ้า agent เริ่มต้นที่ state s

Action-Value Functions

ค่าของการเลือกออก action a ใน state s จาก policy π คือ

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

ในแต่ละ state s และ action a

จะมีค่า expected return

ถ้า agent เริ่มต้นที่ state s

จากนั้นเลือก action a

Action-Value Functions

ค่าของการเลือกออก action a ใน state s จาก policy π คือ

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

ในแต่ละ state s และ action a

จะมีค่า expected return

จากการใช้ policy ในการเลือก action ในทุก ๆ time step

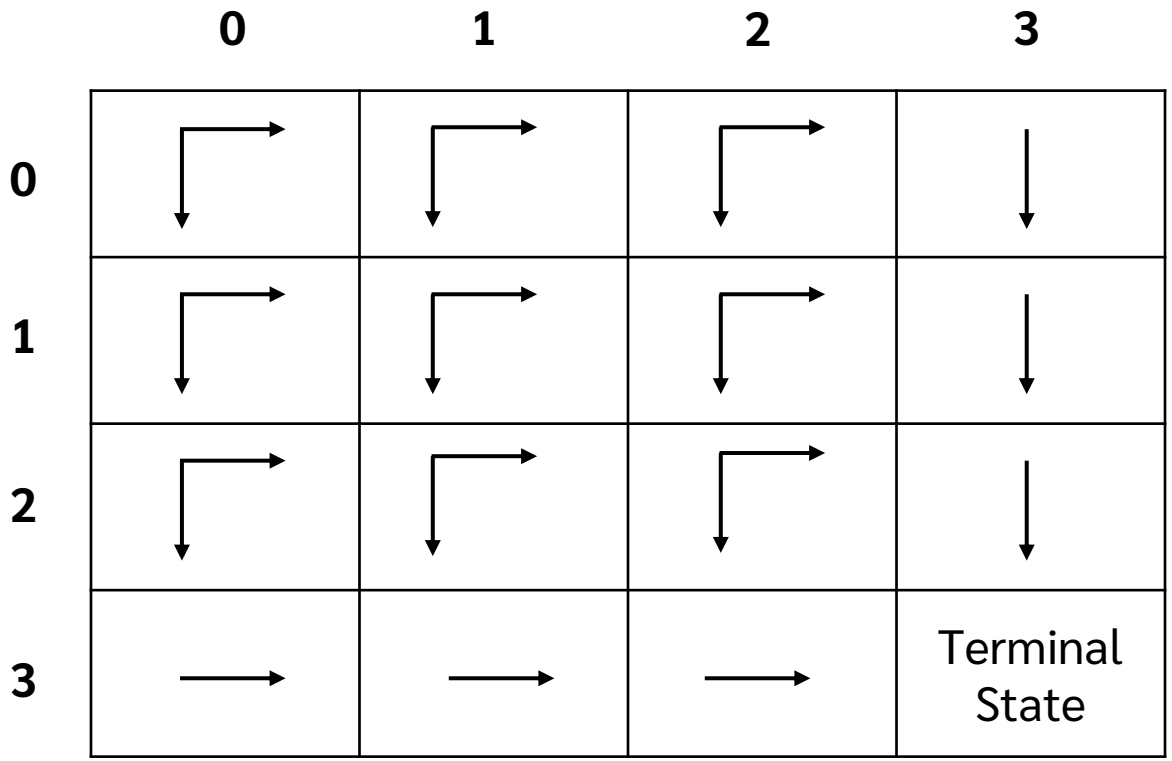
ถ้า agent เริ่มต้นที่ state s

จากนั้นเลือก action a

Example

	0	1	2	3
0	4 4 5 5	4 5 6 6	4 5 7 7	4 5 6 7
1	4 5 6 6	5 5 7 7	6 6 8 8	7 7 8 9
2	5 6 7 7	6 6 8 8	7 7 9 9	8 8 9 10
3	6 7 8 7	7 7 9 8	8 8 10 9	Terminal State

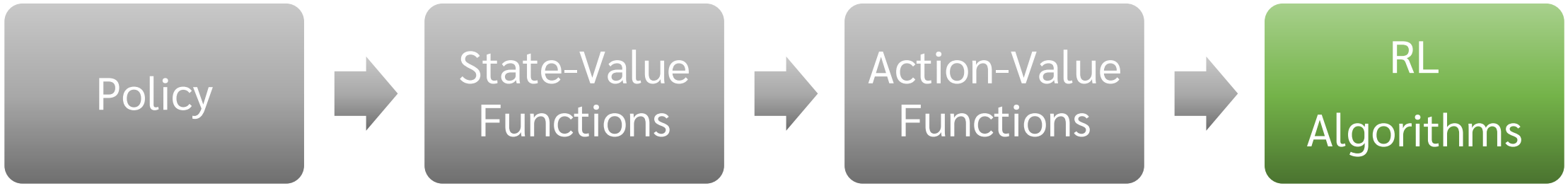
Policy



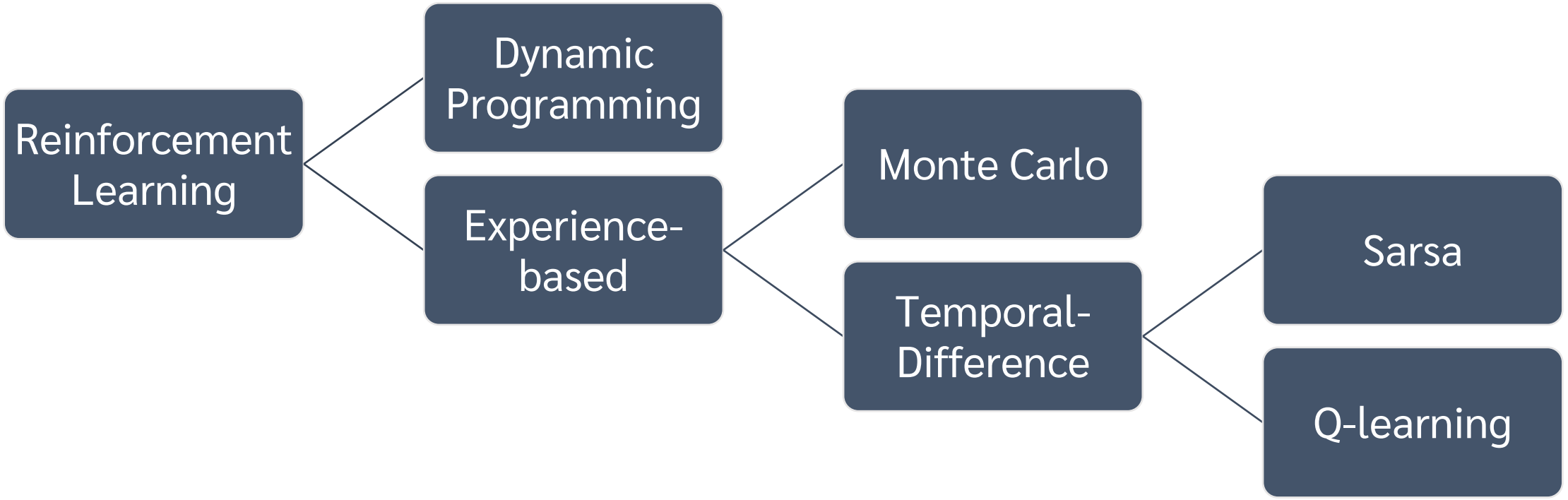
Q table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Contents


Policy
State-Value Functions
Action-Value Functions
RL Algorithms

Reinforcement Learning Algorithms



Q-Learning

Q-Learning

Episode 1: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots, R_T, S_T$

Episode 2: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots, R_T, S_T$

$\vdots$

Episode N: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots, R_T, S_T$

Q-Learning

Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Environment

Q-Learning

Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Get
Q-values

Epsilon-greedy

* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Environment

Q-Learning

Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Get
Q-values

Epsilon-greedy

Select action

A_t

Environment

* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Q-Learning

Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Get
Q-values

Epsilon-greedy

Select action

$(S_t, A_t, R_{t+1}, S_{t+1})$

Receive experience

* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Environment

A_t

R_{t+1}, S_{t+1}

Q-Learning

Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Improve Q-table

Get Q-values

Epsilon-greedy

Select action

$(S_t, A_t, R_{t+1}, S_{t+1})$

Receive experience

* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Environment

A_t

R_{t+1}, S_{t+1}

Q-Learning

Q-table

Improve Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Get Q-values

Epsilon-greedy

Select action

$(S_t, A_t, R_{t+1}, S_{t+1})$

Receive experience

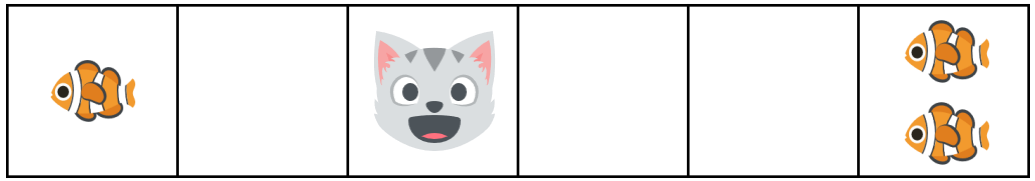
* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Environment

A_t

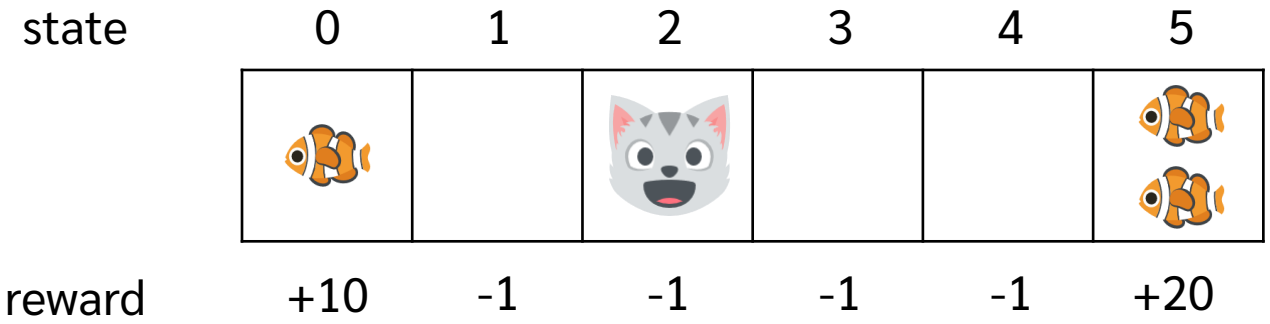
R_{t+1}, S_{t+1}

Grid World

state	0	1	2	3	4	5
						
reward	+10	-1	-1	-1	-1	+20

Grid World

Episode 10,000



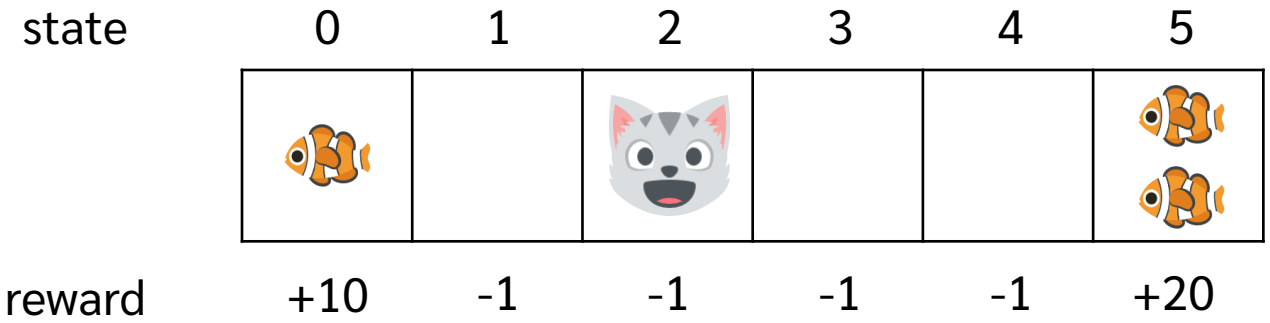
S_0	A_0	R_1	S_1	A_1	R_2	S_2
2	←	-1	1	←	10	0

Q table

	←	→
0	0	0
1	5	0
2	2	0
3	0	0
4	0	0
5	0	0

Grid World

Episode 10,000



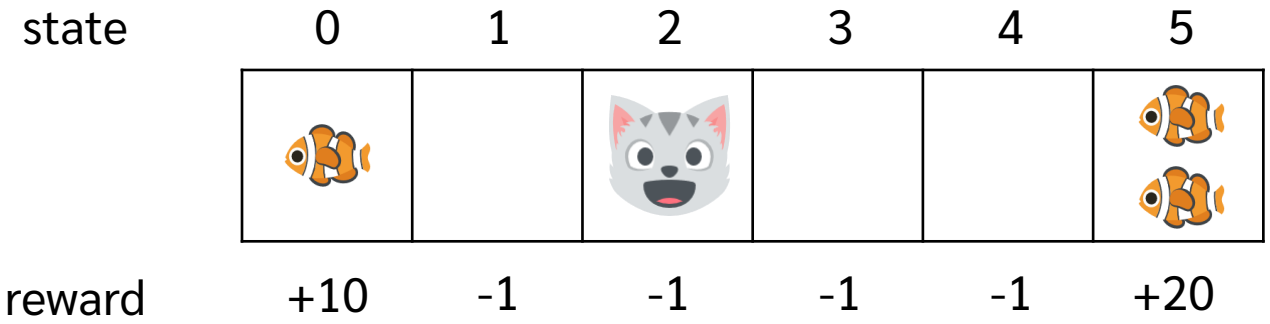
S_0	A_0	R_1	S_1	A_1	R_2	S_2
2	⬅	-1	1	⬅	10	0

Q table

	⬅	➡
0	0	0
1	**6**	0
2	**2.1**	0
3	0	0
4	0	0
5	0	0

Grid World

Episode 20,000



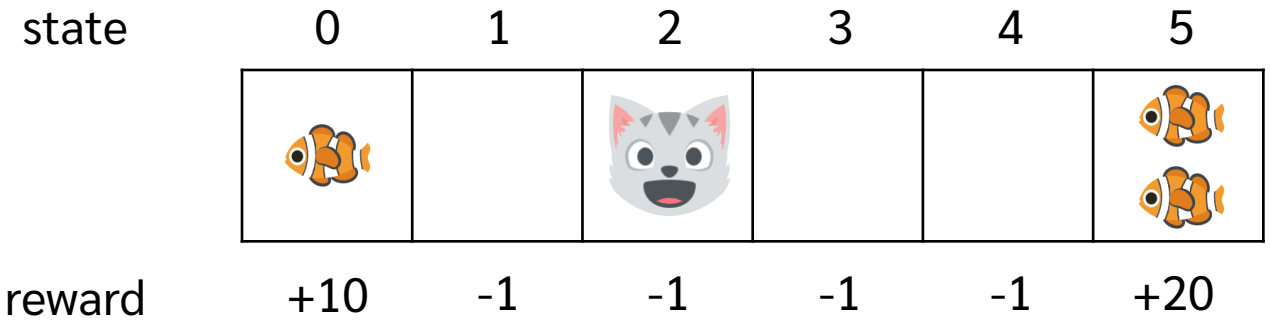
S_0	A_0	R_1	S_1	A_1	R_2	S_2
2	⬅	-1	1	⬅	10	0

Q table

	⬅	➡
0	0	0
1	8	0
2	3.1	0
3	0	0
4	0	0
5	0	0

Grid World

Episode 20,000



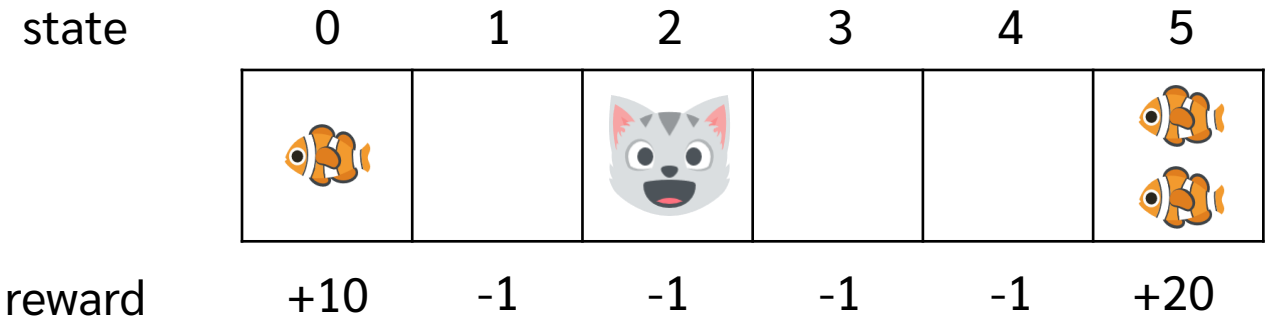
S_0	A_0	R_1	S_1	A_1	R_2	S_2
2	⬅	-1	1	⬅	10	0

Q table

	⬅	➡
0	0	0
1	**9**	0
2	**3.2**	0
3	0	0
4	0	0
5	0	0

Grid World

Episode 30,000



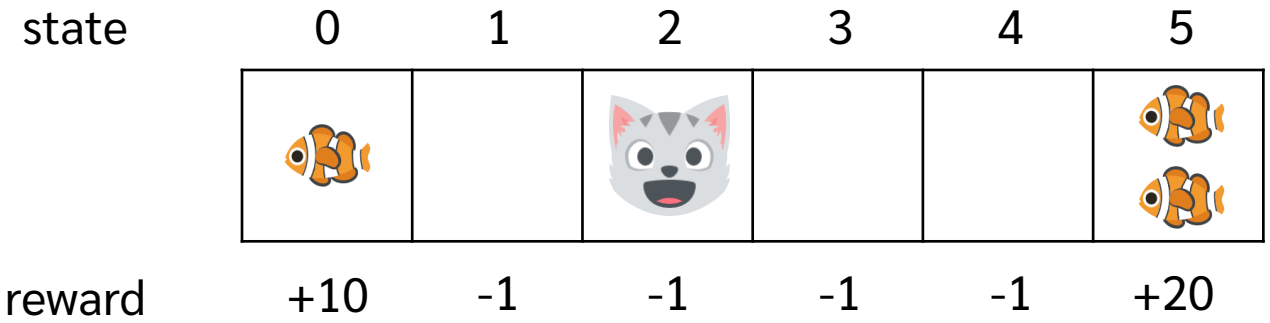
S_0	A_0	R_1	S_1	A_1	R_2	S_2
2	⬅	-1	1	⬅	10	0

Q table

	⬅	➡
0	0	0
1	9.5	0
2	3.8	0
3	0	0
4	0	0
5	0	0

Grid World

Episode 30,000



S_0	A_0	R_1	S_1	A_1	R_2	S_2
2	⬅	-1	1	⬅	10	0

Q table

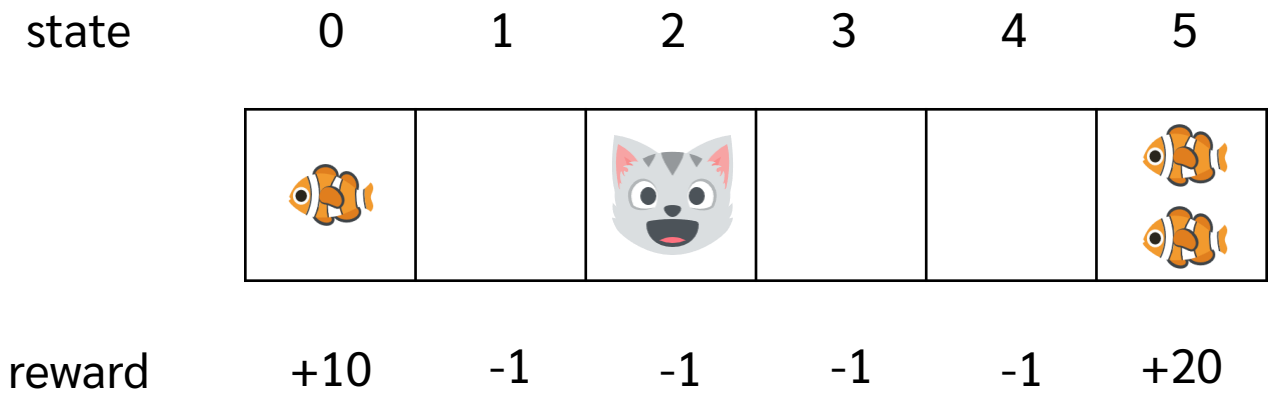
	⬅	➡
0	0	0
1	**10**	0
2	**4**	0
3	0	0
4	0	0
5	0	0

Epsilon-Greedy

เลือก action โดยการสุ่ม
ด้วยความน่าจะเป็น ε

เลือก action จาก Q-table
ด้วยความน่าจะเป็น $1 - \varepsilon$

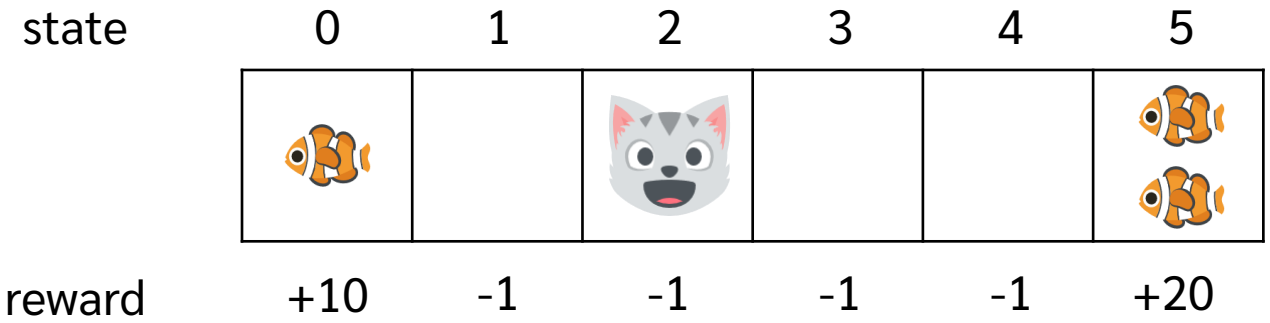
Grid World



$\varepsilon = 0.3$

Grid World

Episode 1



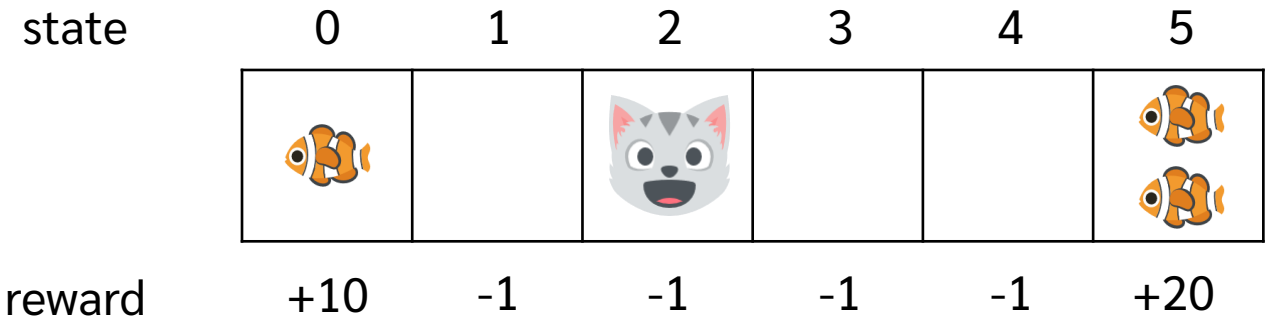
S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

Q table

	←	→
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0

Grid World

Episode 1



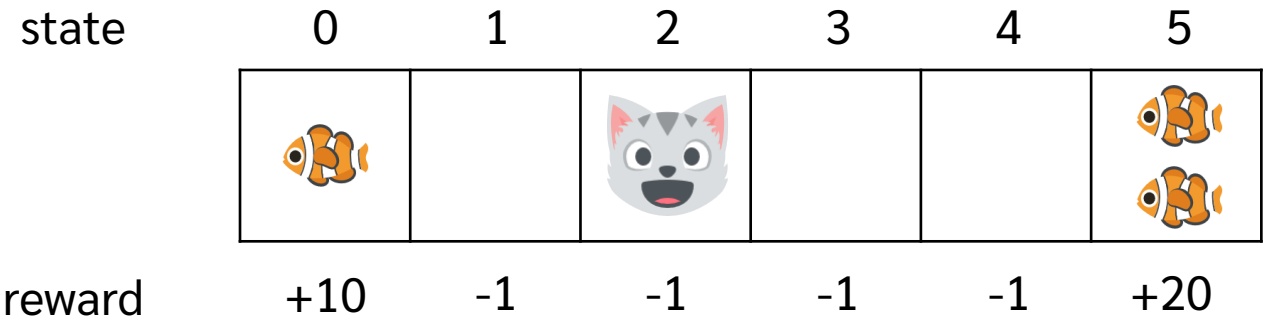
S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

Q table

	←	→
0	0	0
1	0	0
2	0	**0.2**
3	0	**0.2**
4	**-0.2**	**2**
5	0	0

Grid World

Episode 2



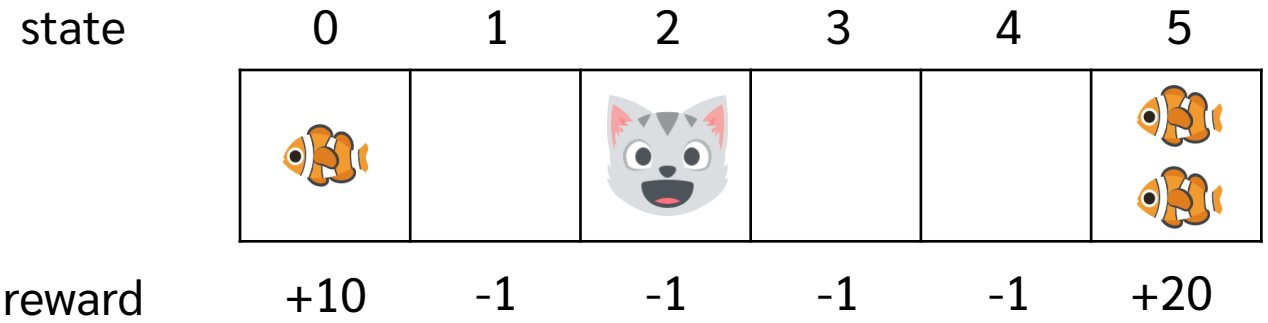
S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4
2	➡	-1	3	⬅	-1	2	⬅	-1	1	⬅	10	0

Q table

	⬅	➡
0	0	0
1	0	0
2	0	0.2
3	0	0.1
4	-0.2	2
5	0	0

Grid World

Episode 2



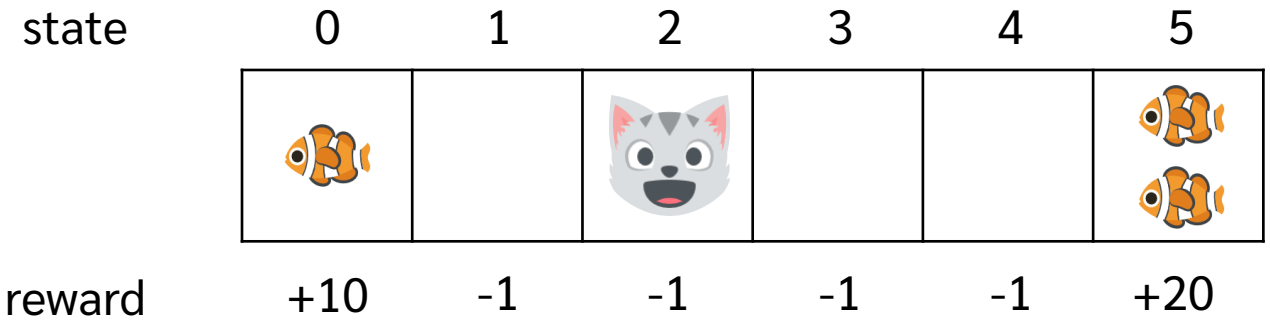
S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4
2	➡	-1	3	⬅	-1	2	⬅	-1	1	⬅	10	0

Q table

	⬅	➡
0	0	0
1	**1**	0
2	**0.1**	**0.1**
3	**-0.1**	0.1
4	-0.2	2
5	0	0

Grid World

Episode N



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3
2	➡	-1	3	➡	-1	4	➡	20	5

Q table

	⬅	➡
0	0	0
1	10	2
2	1	2
3	1	2
4	1	20
5	0	0

Q-Learning

Q-table

Improve
Q-table

State \ Action	Up	Down	Left	Right
(0, 0)	4	5	4	5
(0, 1)	4	6	5	6
(0, 2)	4	7	5	6
⋮	⋮	⋮	⋮	⋮
(3, 2)	8	9	8	10

Get
Q-values

Epsilon-greedy

Select action

$(S_t, A_t, R_{t+1}, S_{t+1})$

Receive experience

* เริ่มต้นให้ค่าของ Q-table ทุกช่องเป็น 0

Environment

A_t

R_{t+1}, S_{t+1}

Q-Learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

Q-Learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

$$(S_t, A_t, R_{t+1}, S_{t+1})$$

Q-Learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

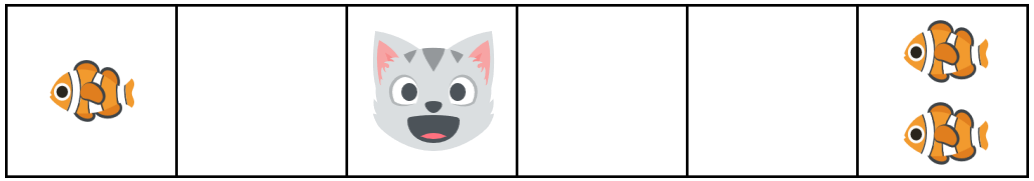
updated value

learning rate

target value

old value

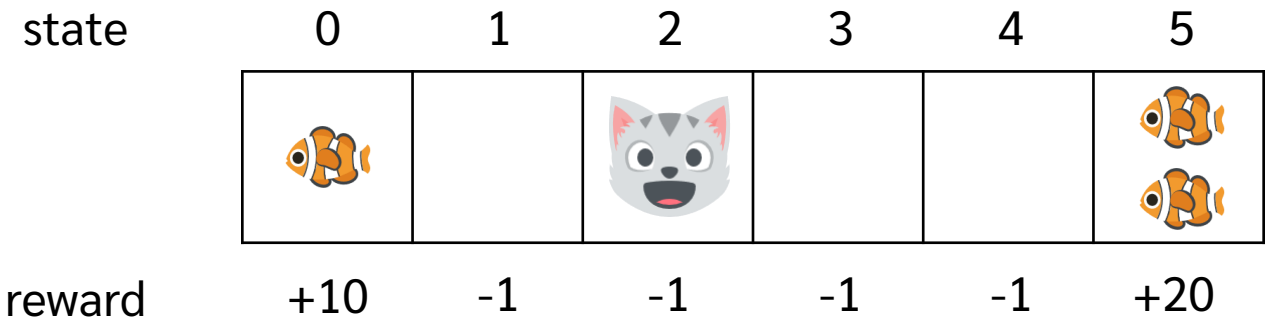
Grid World

state	0	1	2	3	4	5
						
reward	+10	-1	-1	-1	-1	+20

$\varepsilon = 0.2$

$\gamma = 1$

Grid World

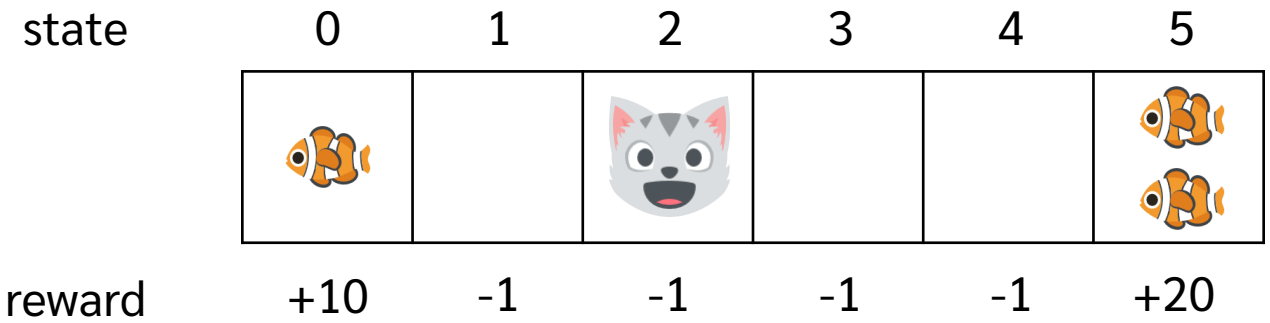


S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

Q table

	←	→
0	0	0
1	11	12
2	7	14
3	12	16
4	10	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 14

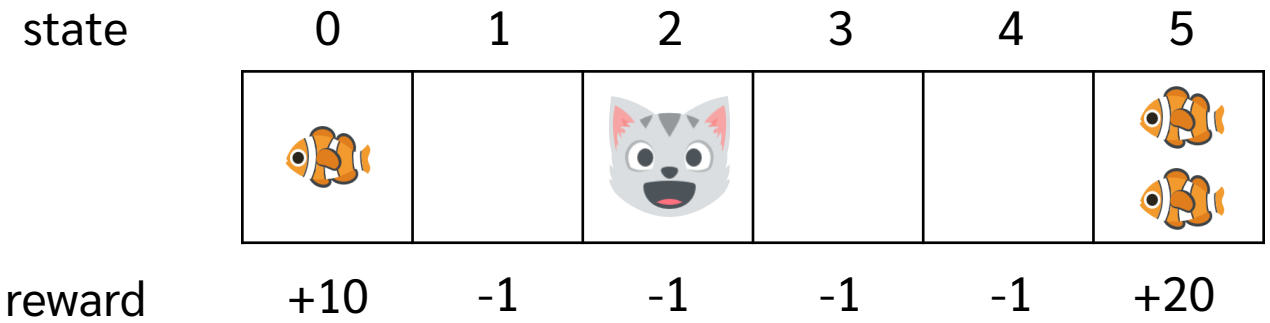
target value = -1 + 16 = 15

update value = 0.1 * (15 – 14) = 0.1

Q table

	←	→
0	0	0
1	11	12
2	7	14
3	12	16
4	10	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 14

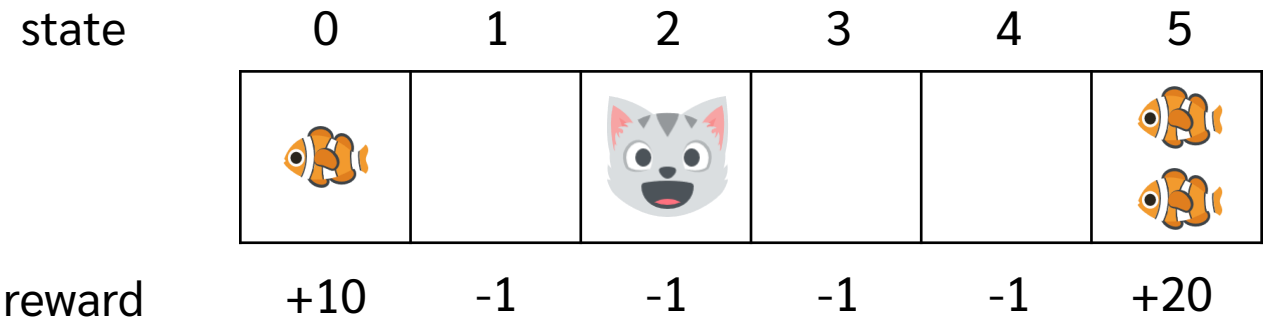
target value = -1 + 16 = 15

update value = 0.1 * (15 – 14) = 0.1

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16
4	10	18
5	0	0

Grid World



S_0 A_0 R_1 S_1 A_1 R_2 S_2 A_2 R_3 S_3 A_3 R_4 S_4 A_4 R_5 S_5

2 → -1 3 → -1 4 ← -1 3 → -1 4 → 20 5

old value = 16

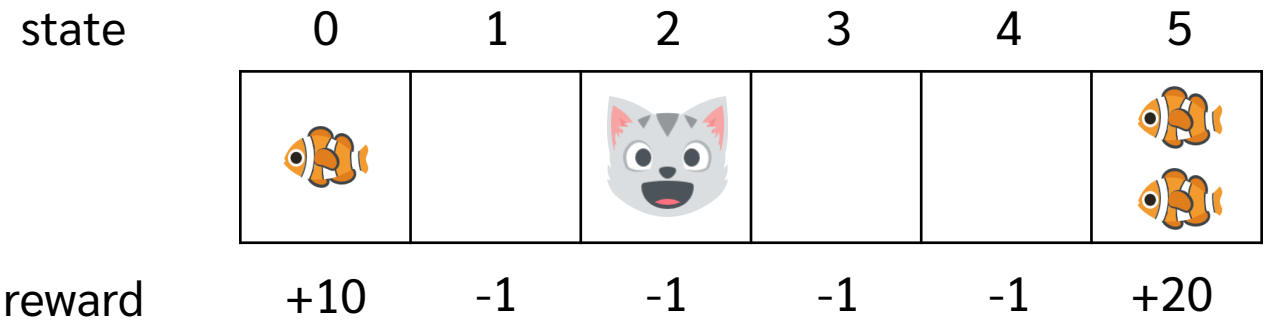
target value = -1 + 18 = 17

update value = 0.1 * (17 – 16) = 0.1

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16
4	10	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 16

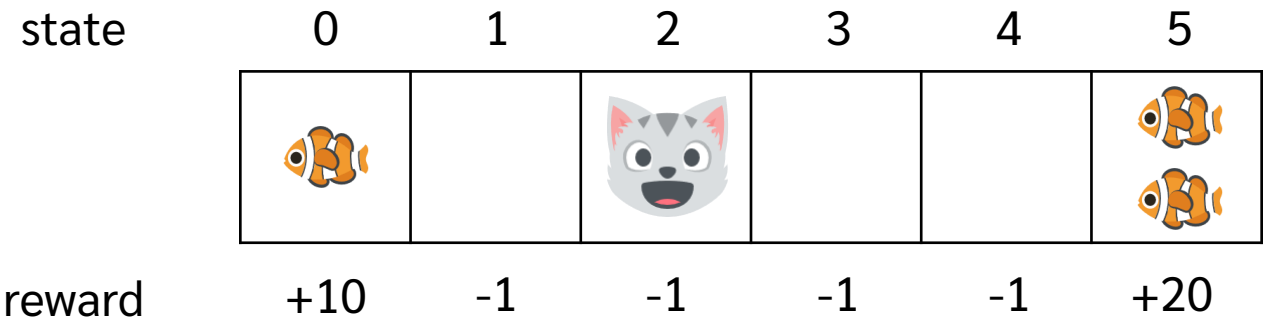
target value = -1 + 18 = 17

update value = 0.1 * (17 – 16) = 0.1

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.1
4	10	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 10

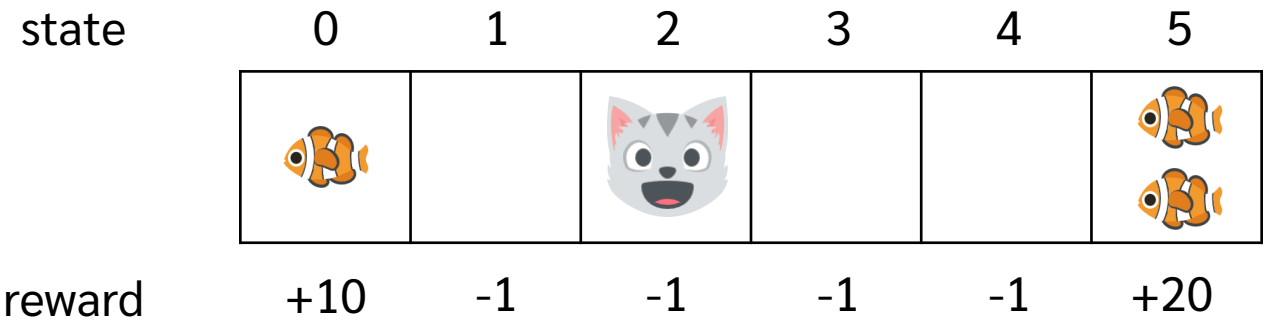
target value = -1 + 16.1 = 15.1

update value = 0.1 * (15.1 – 10) = 0.51

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.1
4	10	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 10

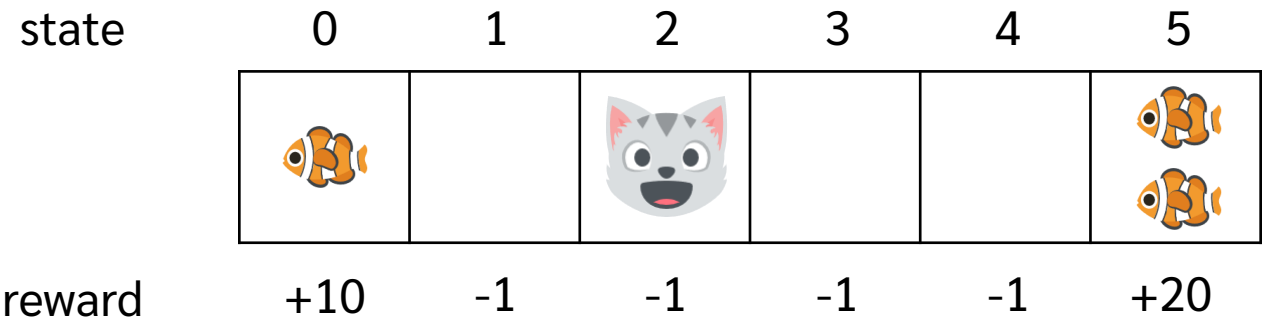
target value = -1 + 16.1 = 15.1

update value = 0.1 * (15.1 – 10) = 0.51

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.1
4	10.51	18
5	0	0

Grid World



S_0 A_0 R_1 S_1 A_1 R_2 S_2 A_2 R_3 S_3 A_3 R_4 S_4 A_4 R_5 S_5

2 → -1 3 → -1 4 ← -1 3 → -1 4 → 20 5

old value = 16.1

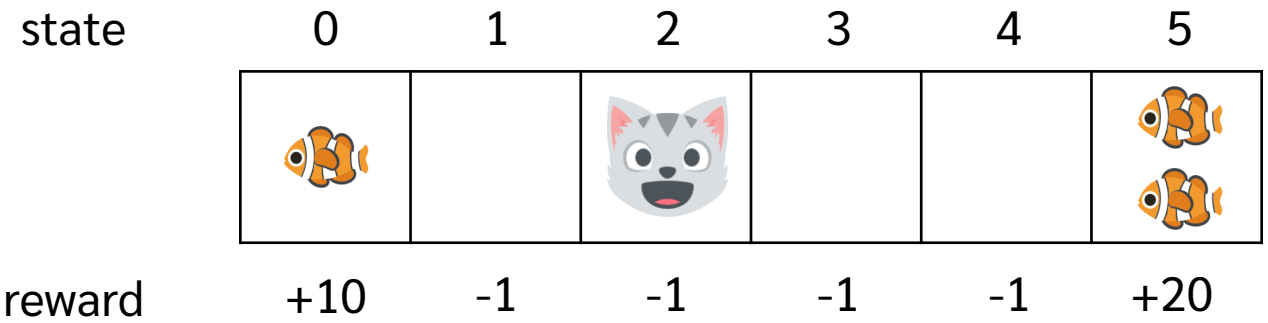
target value = -1 + 18 = 17

update value = 0.1 * (17 – 16.1) = 0.09

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.1
4	10.51	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	➡	-1	3	➡	-1	4	⬅	-1	3	➡	-1	4	➡	20	5

old value = 16.1

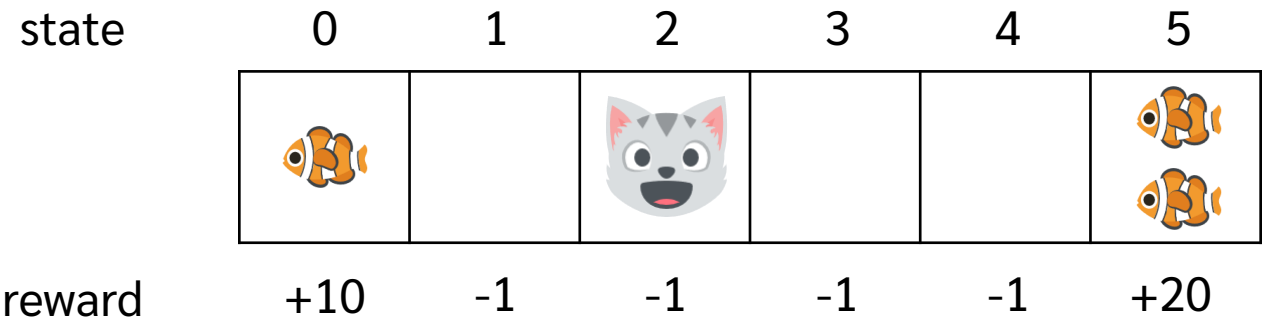
target value = -1 + 18 = 17

update value = 0.1 * (17 – 16.1) = 0.09

Q table

	⬅	➡
0	0	0
1	11	12
2	7	14.1
3	12	16.19
4	10.51	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 18

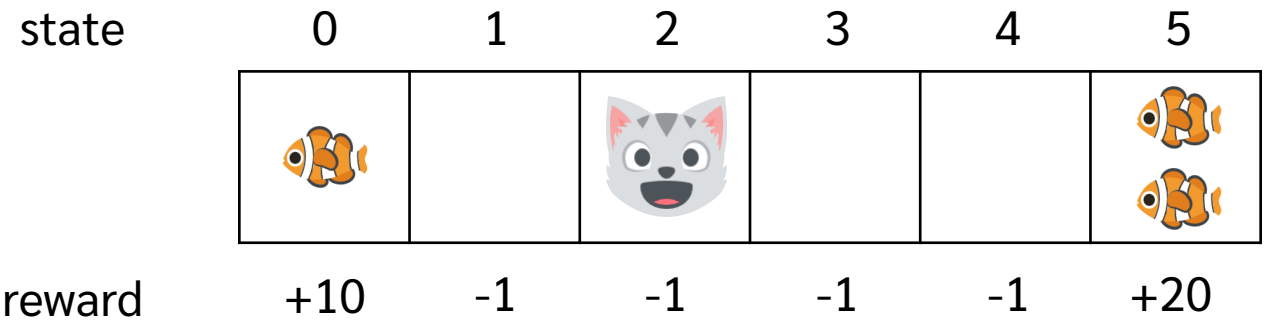
target value = 20 + 0 = 20

update value = 0.1 * (20 – 18) = 0.2

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.19
4	10.51	18
5	0	0

Grid World



S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
2	→	-1	3	→	-1	4	←	-1	3	→	-1	4	→	20	5

old value = 18

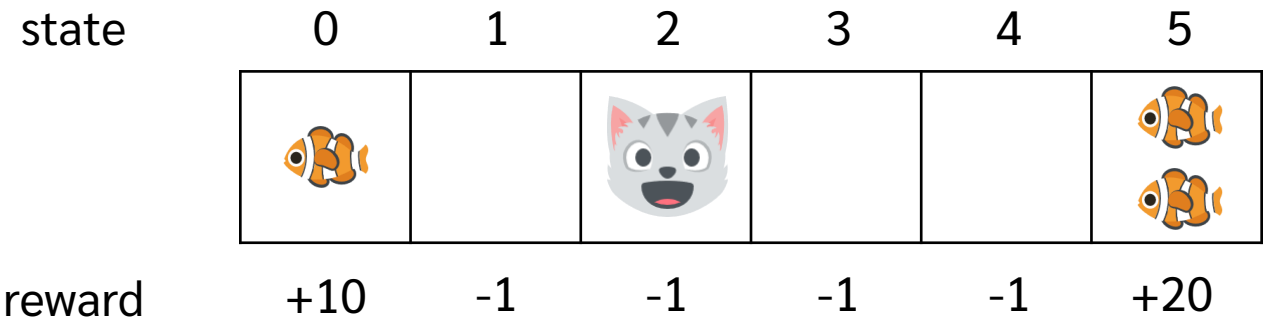
target value = 20 + 0 = 20

update value = 0.1 * (20 – 18) = 0.2

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.19
4	10.51	18.2
5	0	0

Grid World



S_0 A_0 R_1 S_1 A_1 R_2 S_2 A_2 R_3 S_3 A_3 R_4 S_4 A_4 R_5 S_5

2 → -1 3 → -1 4 ← -1 3 → -1 4 → 20 5

จบ 1 episode

Q table

	←	→
0	0	0
1	11	12
2	7	14.1
3	12	16.19
4	10.51	18.2
5	0	0