Artificial Intelligence

Taro Sekiyama

National Institute of Informatics (NII)
sekiyama@nii.ac.jp

Types of ML algorithms

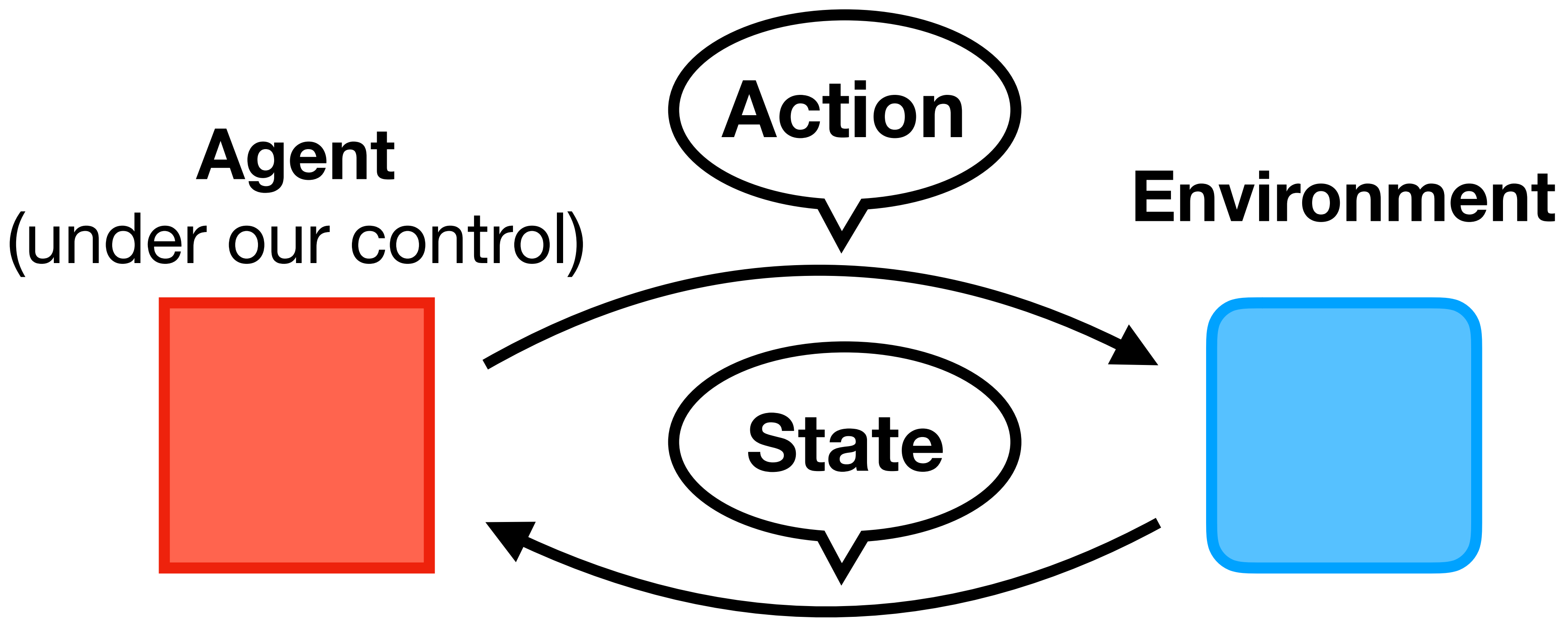
- Supervised learning
  - Learning *functions* from input—output pairs
  - **Ex:** classification and regression
- Unsupervised learning
  - Learning *structures/patterns* of datasets
  - **Ex:** cluster analysis and
    feature transformation/extraction

Types of ML algorithms

- Supervised learning
  - Learning *functions* from input—output pairs
  - **Ex:** classification and regression
- Unsupervised learning
  - Learning *structures/patterns* of datasets
  - **Ex:** cluster analysis and
    feature transformation/extraction
- ***Reinforcement learning***

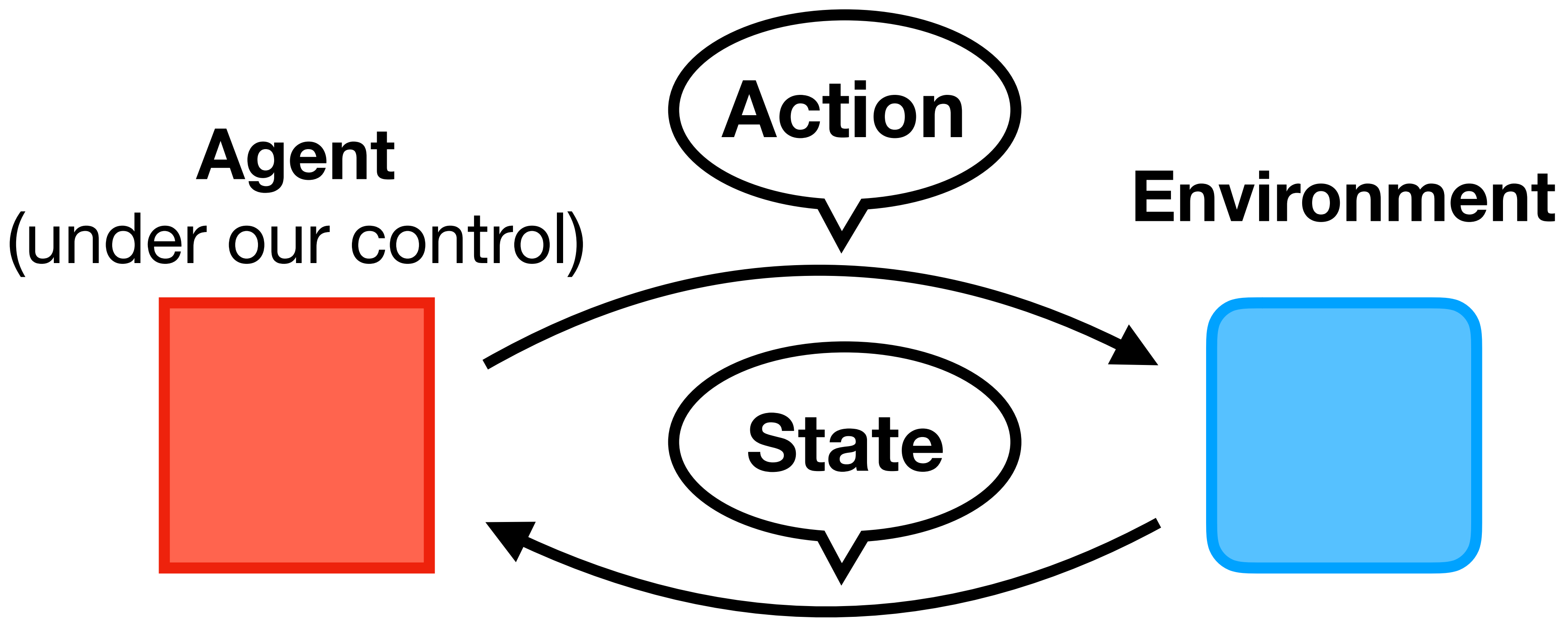
Reinforcement learning (RL)

- Learning ***strategies*** to interact with environments



Reinforcement learning (RL)

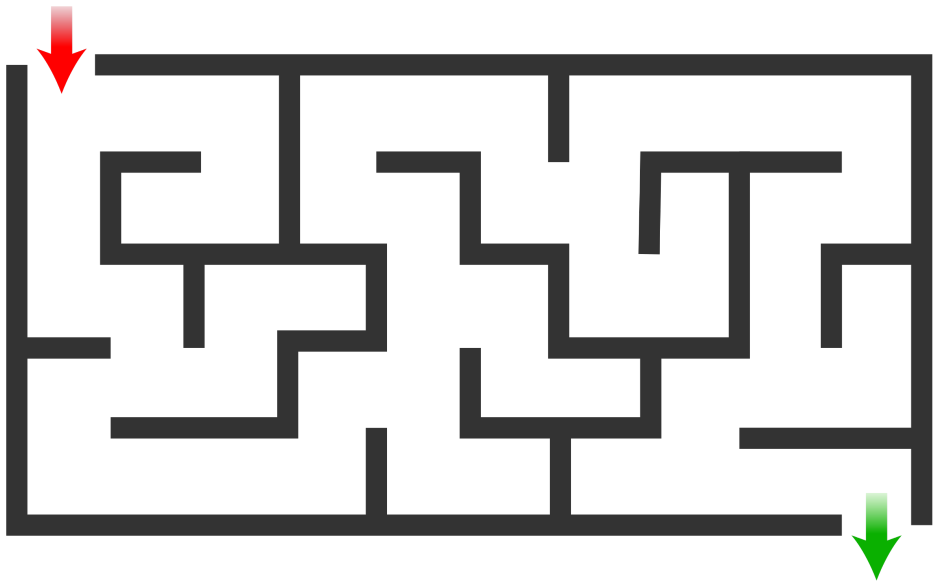
- **Goal:** *learning* ***series of actions*** *that lead to desirable states or do not to undesirable ones*



RL example

Maze

- Objective:
  reaching the goal ASAP
- Action: moving direction (up, down, left, or right)
- State: position in maze

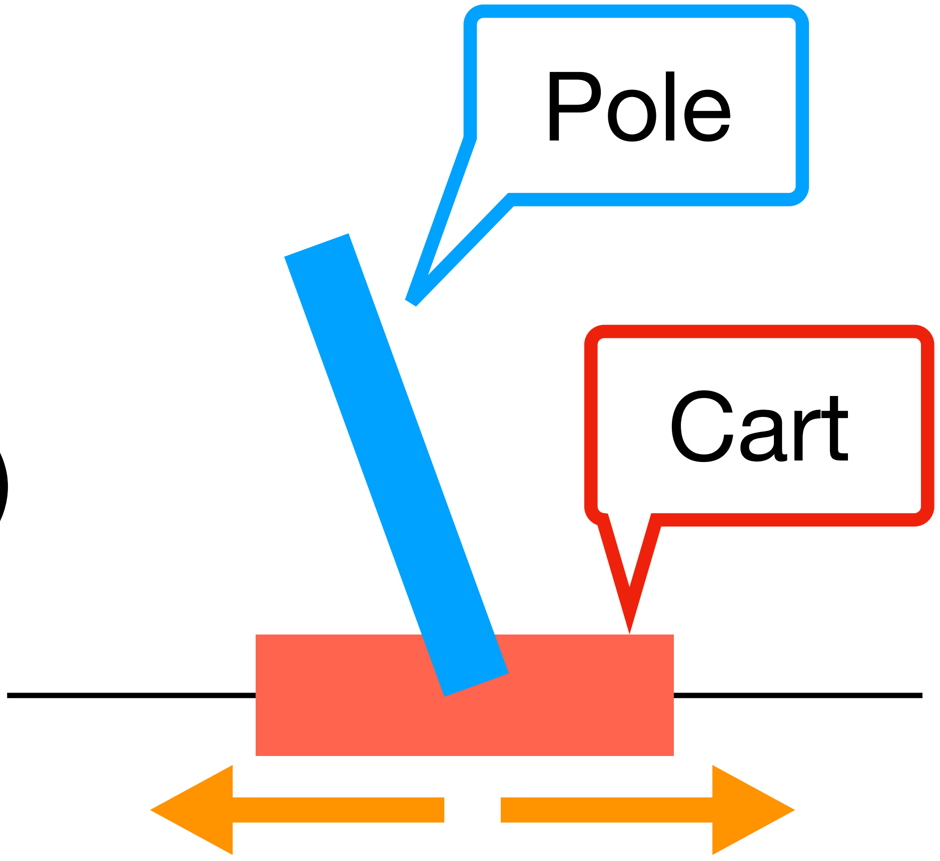


https://en.wikipedia.org/wiki/Maze#/media/File:Maze_simple.svg

RL example

Cart-pole problem

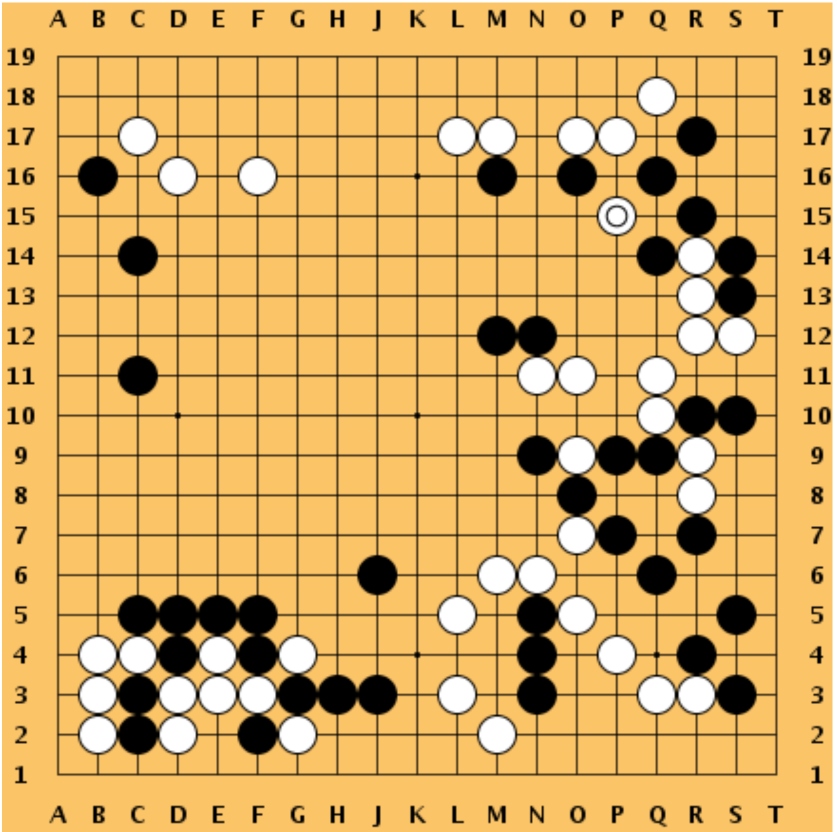
- Objective: preventing the pole from failing over
(i.e., $0 <$ the pole angle < 180)
- Action: moving direction of the cart (left or right)
- State: pole angle,
pole velocity, cart velocity, …
- C.f. https://towardsdatascience.com/cartpole-introduction-to-reinforcement-learning-ed0eb5b58288



RL example

Go (board game)

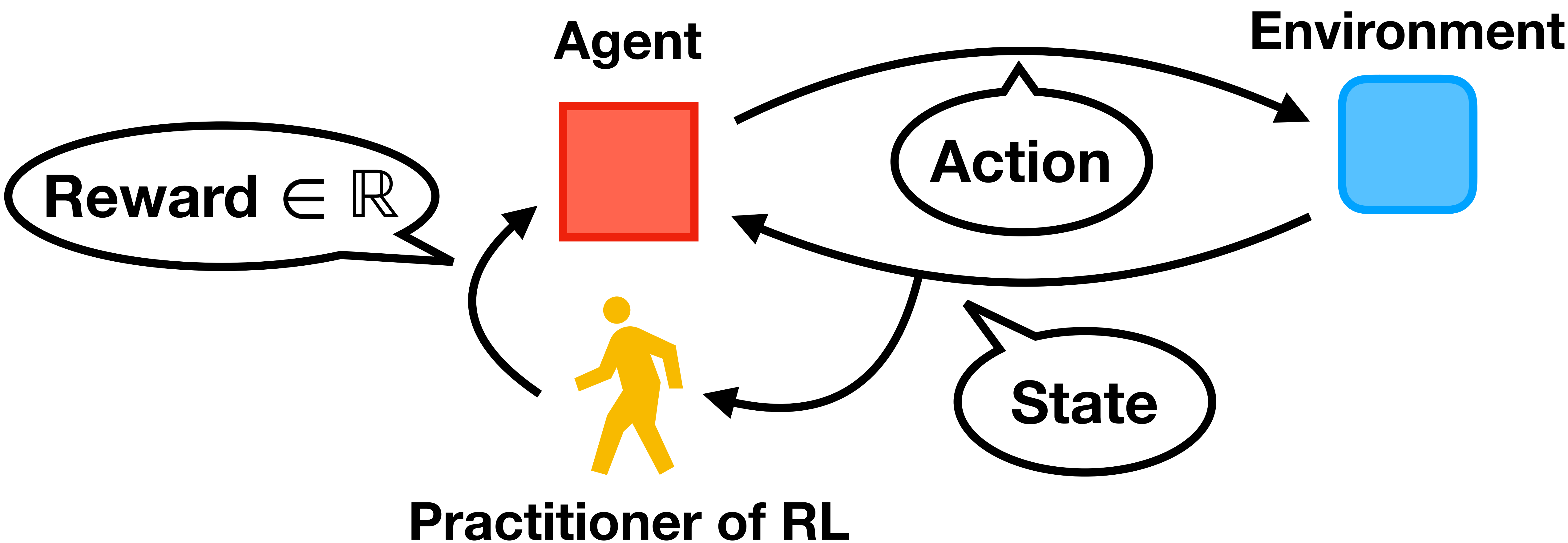
- Objective: winning the game
- Action: positions where the next piece is put
- State: the current board (after the opponent player play a move)



https://upload.wikimedia.org/wikipedia/commons/a/ab/Go_game_Kobayashi-Kato.png

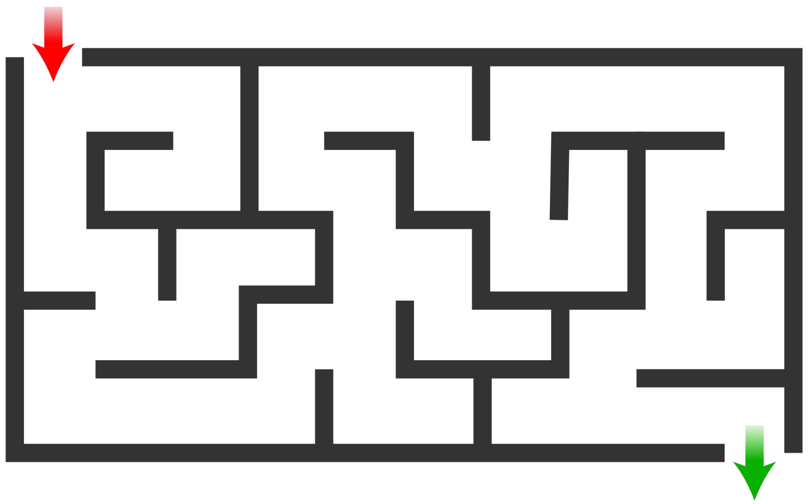
Reward

- Metric to evaluate how good strategies are
- The goal of RL is to maximize the sum of the rewards obtained by performing the actions



Reward example

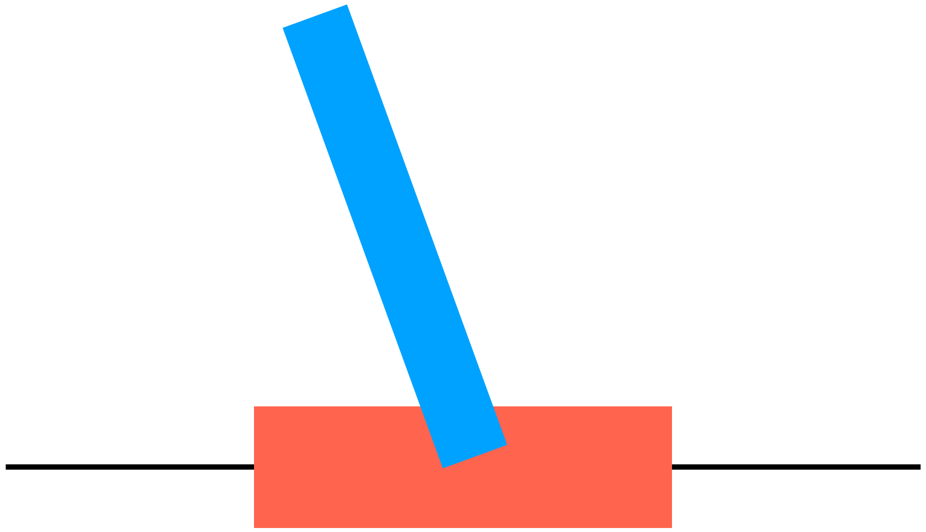
Maze



- Reward -1 is assigned
  for any position except for
  the goal position
- Maximizing the sum of rewards
  is the same as reaching the goal ASAP
  - The sum of rewards is larger
    as we reach the goal sooner

Reward example

Cart-pole problem

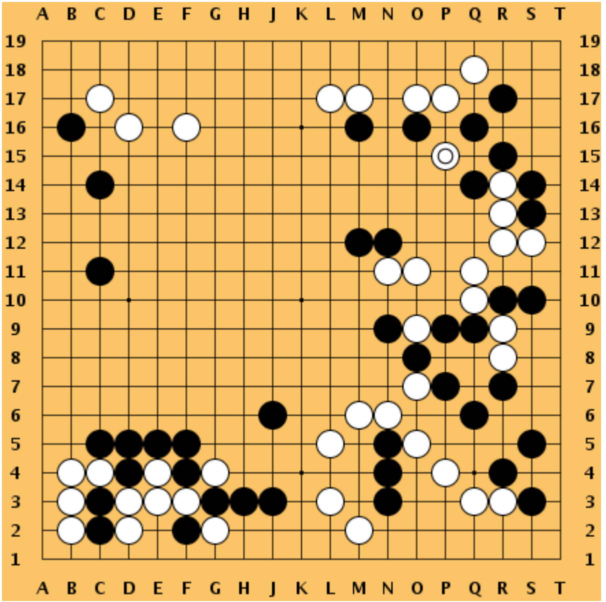


- Reward 1 is assigned for any state
- Maximizing the sum of rewards is the same as preventing the pole from failing
  - If the pole never falls over, the sum of rewards is the infinity
  - If the pole falls over, it is a some finite number

Reward example

Go

- Reward
  - 1 if the player wins
  - -1 if the player loses
  - 0 otherwise
- Maximizing the sum of rewards is the same as winning the game



Formulation by math

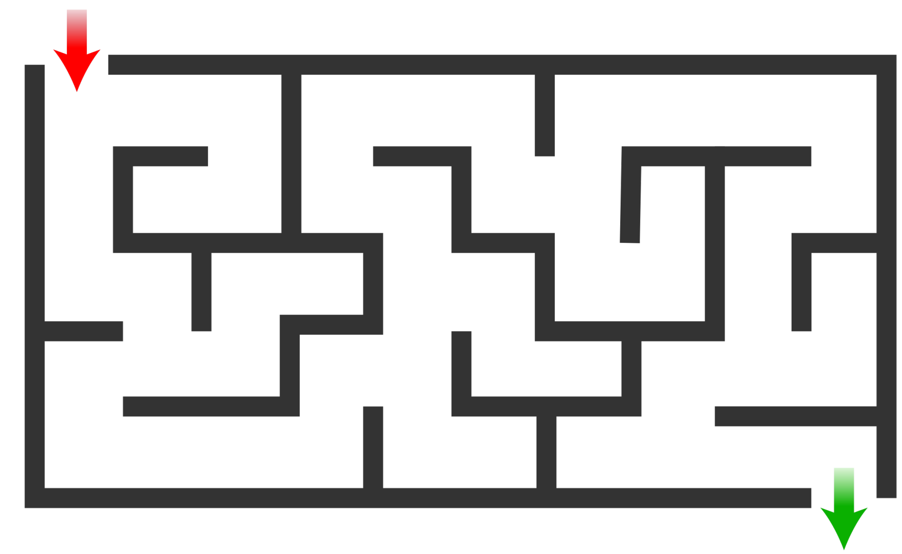
- Will formulate the class of RL problems for:
  - Giving general approaches to various RL problems
  - Clarifying the limitation of each approach

Mathematical formulation

- A *simple* RL problem is (A, S, p, r) where:
  - A is a set of actions
  - S is a set of states
  - p is a state transition function $\in S \times A \rightarrow S$
    - $p(s, a)$ is the state after doing action a in state s
  - r is a reward function $\in S \times A \rightarrow \mathbb{R}$
    - $r(s, a)$ is the reward for action a in state s

Formulation example

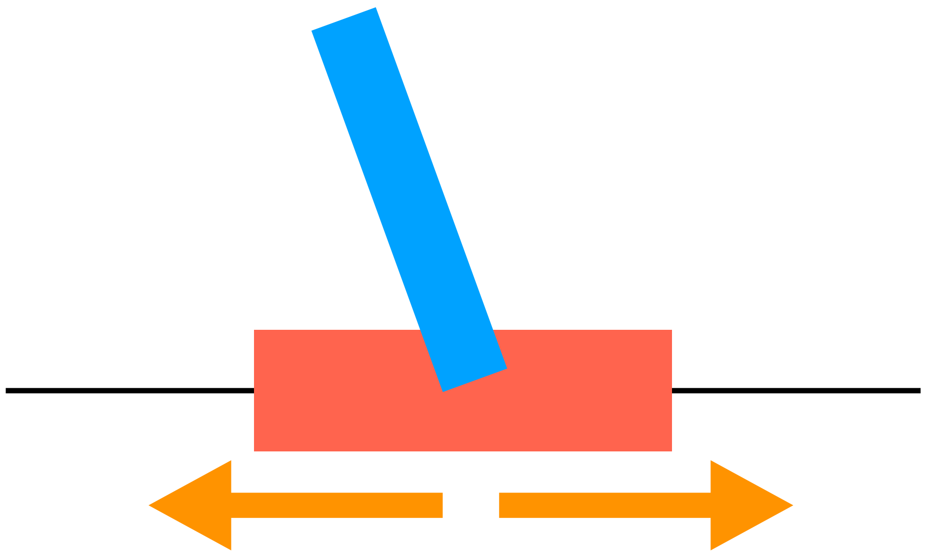
Maze



- $A = \{(0,1), (0,-1), (-1,0), (1,0)\}$
- $S \subseteq \mathbb{N} \times \mathbb{N}$
- $p(s,a) = \begin{cases} s & \text{(if } s \in G\text{)} \\ s + a & \text{(if } s \notin G \text{ and } (s,a) \in B\text{)} \end{cases}$ where:
  - $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
  - $(s,a) \in B \subseteq S \times A$ indicates the maze allows us to proceed along direction a at position s
- $r(s,a) = \begin{cases} 0 & \text{(if } s \in G\text{)} \\ -1 & \text{(if } s \notin G\text{)} \end{cases}$

Formulation example

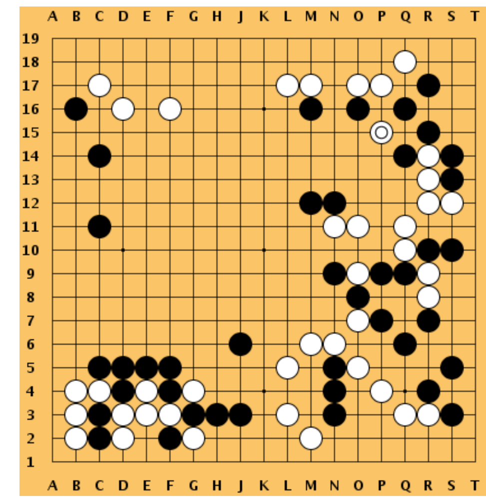
Cart-pole problem



- $A = \{\text{left}, \text{right}\}$
- $S = [0,180] \times \mathbb{R} \times \mathbb{R} \times \cdots$
  - pole angle, pole velocity, cart velocity, …
- $p(s, a)$ is determined by response from sensors or simulation
- $r(s, a) = \begin{cases} 1 & \text{(if the pole doesn't fall over)} \\ 0 & \text{(if the pole has falled over)} \end{cases}$

Formulation example

Go



- $A \subseteq \mathbb{N} \times \mathbb{N}$
- $S \subseteq \{\mathrm{B}, \mathrm{W}, \mathrm{N}\}^{[1,19]\times[1,19]}$
- $p(s, a)$: the board state after playing the player's move a and the opponent's move in s
- $r(s, a) = \begin{cases} 1 & \text{(if the player wins in } p(s, a)\text{)} \\ -1 & \text{(if the player loses in } p(s, a)\text{)} \\ 0 & \text{(otherwise)} \end{cases}$

Objective

Finding a ***policy*** that maximizes ***the cumulative reward***

- A ***policy*** π is a function $\in S \to A$
- ***The cumulative reward*** (the sum of rewards) ***under*** π is

$$r(s_0, \pi(s_0)) + r(s_1, \pi(s_1)) + \cdots$$

where:

- s_0 is an initial state
- $s_{i+1} = p(s_i, \pi(s_i))$ for $i \geq 0$

Optimal policy

$\pi^\star$

Objective

Finding a ***policy*** that maximizes ***the cumulative reward***

- A ***policy*** π is a function $\in S \to A$
- ***The cumulative reward*** (the sum of rewards) ***under*** π is

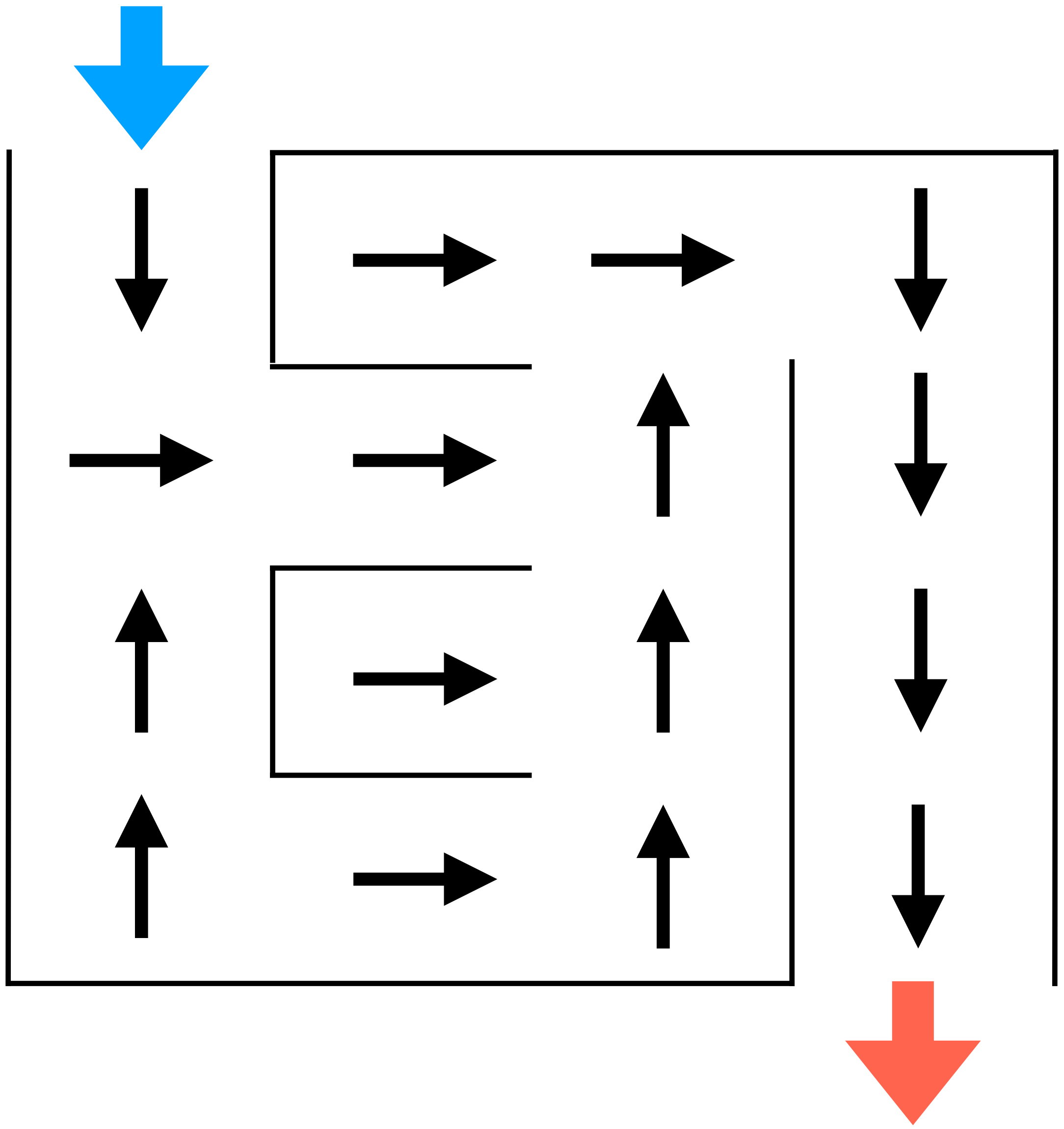
$$\sum_{i=0} r(s_i, \pi(s_i))$$

where:

- s_0 is an initial state
- $s_{i+1} = p(s_i, \pi(s_i))$ for $i \geq 0$

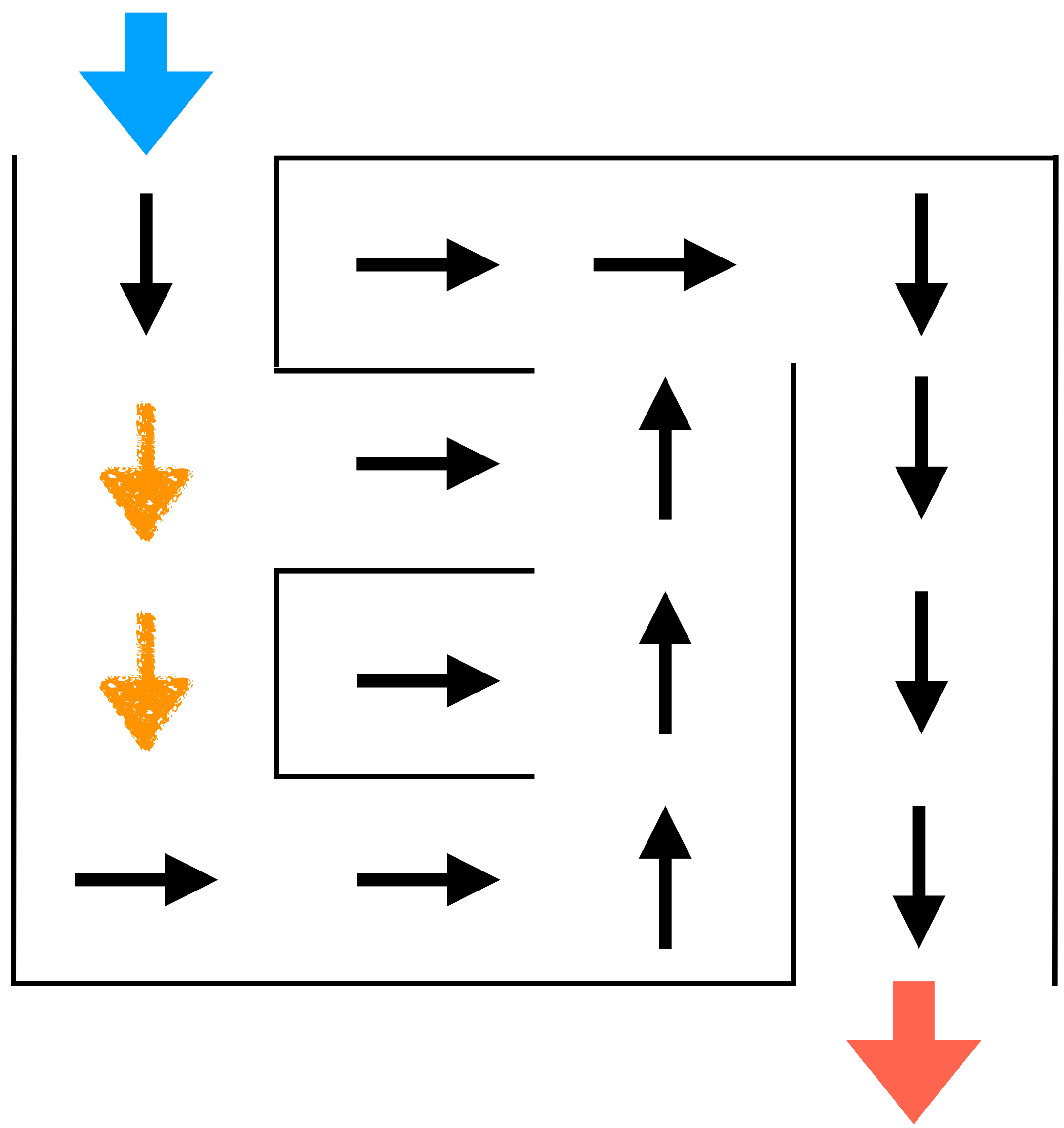
Policy example

- An optimal policy $\pi^{\star}$ for the maze is given by:



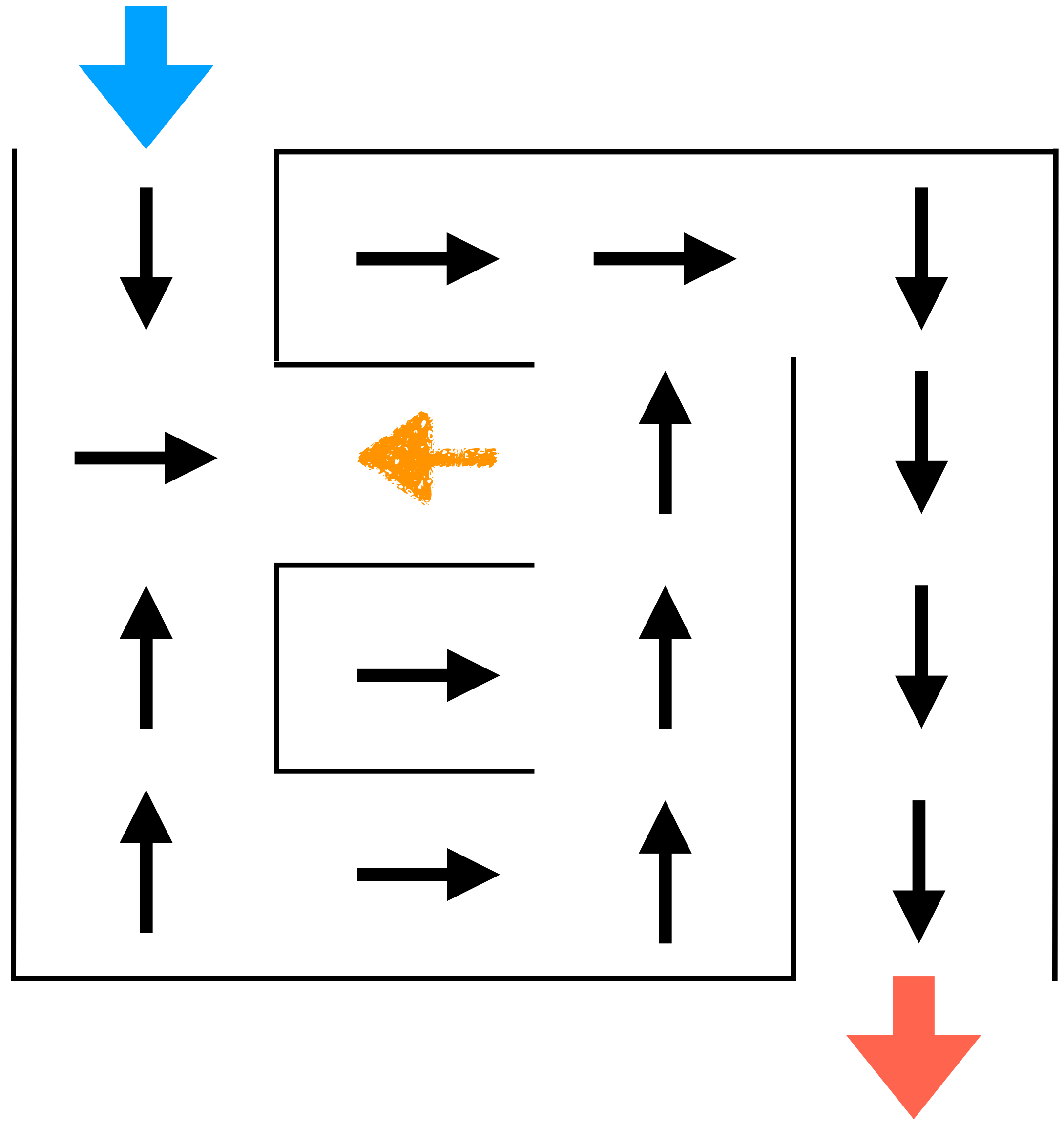
Policy example

- A *non*-optimal policy π is given by:



Policy example

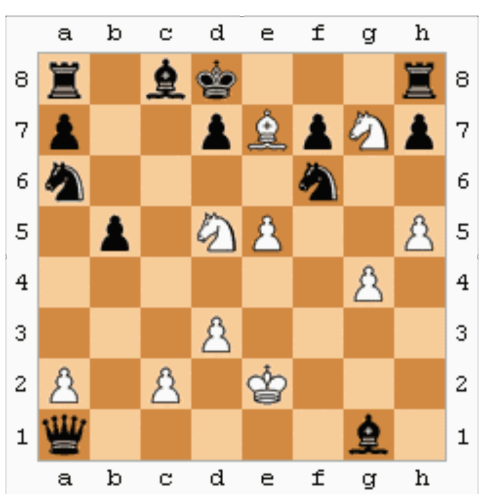
- A *non*-optimal policy π is given by:



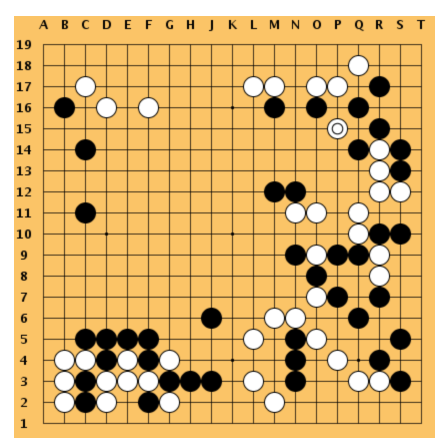
Problems with simple formulation

1. State transition may be nondeterministic

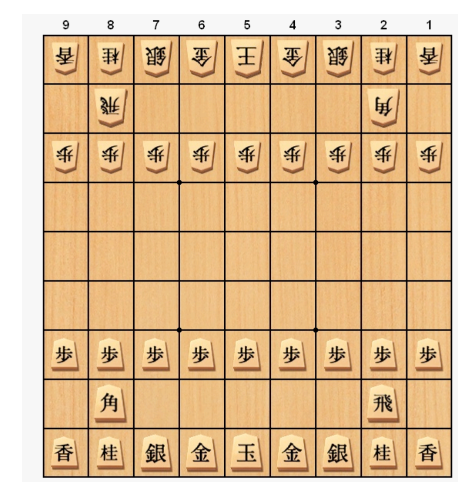
- **Ex:** Board game (Chess, Go, Shogi, etc.)
  - Opponent's moves may be different even for the same board state



https://upload.wikimedia.org/wikipedia/commons/c/cc/Immortal_game_animation.gif



https://upload.wikimedia.org/wikipedia/commons/a/ab/Go_game_Kobayashi-Kato.png



https://upload.wikimedia.org/wikipedia/commons/9/9f/Shogiban.png

Problems with simple formulation

2. The cumulative reward may diverge

- For reward function r and policy π s.t.

  (1) $r(s_{2i}, \pi(s_{2i})) = 1$

  (2) $r(s_{2i+1}, \pi(s_{2i+1})) = -1$,

  the reward $\sum_i r(s_i, \pi(s_i))$ never converges

Extensions of formulation

Problem 1.
State transition may be nondeterministic

Solution. Extending with ***probability***

Problem 2.
The cumulative reward may diverge

Solution. Introducing ***discount factors***

General formulation (1/2)

- A RL problem is a ***Markov decision process (MDP)*** (A, S, p, r, γ) where:
  - A is a set of actions
  - S is a set of states
  - p is ***a probability distribution*** of states given state-action pairs
    - $p(s' \mid s, a)$ is the probability that the state s' is reached when action a is performed in the state s

General formulation (2/2)

- A RL problem is a ***Markov decision process (MDP)*** (A, S, p, r, γ) where:
  - r is ***a probability distribution*** of rewards state-action pairs
    - $r(t \mid s, a)$ is the probability that t is the reward for action a in state s
  - $\gamma \in [0,1]$ is ***a discount factor***
    - Taken into account in the cumulative reward

Review: cumulative reward

$$\sum_{i=0} r(s_i, \pi(s_i))$$

where:

- s_0 is an initial state
- $s_{i+1} = p(s_i, \pi(s_i))$ for $i \geq 0$

Cumulative discounted reward

$$\boxed{\sum_{i=0} \boxed{\gamma^i \cdot} r(s_i, \pi(s_i))}$$

where:

- s_0 is an initial state
- $s_{i+1} = p(s_i, \pi(s_i))$ for ≥ 0

$r(s_0, \pi(s_0)) + \gamma \cdot r(s_1, \pi(s_1)) + \gamma^2 \cdot r(s_2, \pi(s_2)) + \cdots$ converges
(if $r(s_i, \pi(s_i))$ doesn't change drastically as i increases)

Cumulative discounted reward

$$\sum_{i=0} \gamma^i \cdot r(s_i, \pi(s_i))$$

where:

In MDPs, these are ***probabilistic distributions***

- s_0 is an initial state
- $s_{i+1} = p(s_i, \pi(s_i))$ for $i \geq 0$

Cumulative discounted reward in MDPs

- Formulated by ***expected values***

$$\mathbb{E}[\sum_{i=0} \gamma^i \cdot \mathscr{R}_i]$$

where:

- $\mathscr{R}_i \sim r(\cdot \mid \mathscr{S}_i, \pi(\mathscr{S}_i))$ for any $i \geq 0$
  (i.e., $\mathscr{R}_i$ is a random variable following $r(\cdot \mid \mathscr{S}_i, \pi(\mathscr{S}_i))$)
- $\mathscr{S}_{i+1} \sim p(\cdot \mid \mathscr{S}_i, \pi(\mathscr{S}_i))$ for any $i \geq 0$
- $\mathscr{S}_0$ is a random variable standing for initial states

Objective

- Finding an optimal policy $\pi^\star$ maximizing the cumulative discounted reward $\mathbb{E}[\sum_{i=0} \gamma^i \cdot \mathcal{R}_i]$

Approaches to RL problems

1. **Value-based**

2. **Policy-based**

Approaches to RL problems

1. **Value-based**
   - **A. Model-based**
   - **B. Model-free**
2. **Policy-based**

Approaches to RL problems

1. **Value-based**
   - **A. Model-based**
     - Dynamic programming
   - **B. Model-free**
     - Monte Carlo
     - Q-learning
2. **Policy-based**

Approaches to RL problems

1. **Value-based**
   - **A. Model-based**
     - Dynamic programming
   - **B. Model-free**
     - Monte Carlo
     - Q-learning
2. **Policy-based**
   - Policy gradient

Approaches to RL problems

1. ***Value-based***
   A. **Model-based**
      - Dynamic programming

   B. **Model-free**
      - Monte Carlo
      - Q-learning
2. **Policy-based**
   - Policy gradient

Value-based approach

1. Estimating reward metrics

2. Deriving a policy from the metrics estimated
   - The derived policy is optimal
     if the metrics satisfy some desirable property

- Note: usually assume MDPs are *finite*
  - Action set A and state set S are finite

Value-based approach

1. Estimating ***reward metrics***
2. Deriving a policy from the metrics estimated
   - The derived policy is optimal if the metrics satisfy some desirable property

- Note: usually assume MDPs are *finite*
  - Action set A and state set S are finite

Reward metrics

- ***Value function*** $V^\pi \in S \to \mathbb{R}$

  $\mathscr{R}_i \sim r(\,\cdot \mid \mathscr{S}_i, \pi(\mathscr{S}_i))$

  - Measuring how good state s is under π

$$V^\pi(s) = \mathbb{E}[\sum_{i=0} \gamma^i \cdot \mathscr{R}_i \mid \mathscr{S}_0 = s]$$

- ***Q-value function*** $Q^\pi \in S \times A \to \mathbb{R}$
  - Measuring how good action a is at state s under π

$$Q^\pi(s, a) = \mathbb{E}[\sum_{i=0} \gamma^i \cdot \mathscr{R}_i \mid \mathscr{S}_0 = s, \mathscr{R}_0 \sim p(\,\cdot \mid s, a)]$$

Value-based approach

1. Estimating ***reward metrics***
2. ***Deriving a policy from the metrics estimated***
   - The derived policy is optimal
     if the metrics satisfy some desirable property

- Note: usually assume MDPs are *finite*
  - Action set A and state set S are finite

Greedy derivation

Deriving a ***greedy*** policy by preferring actions that maximize the metric values

- From value function V

$$\pi_V(s) = \arg\max_{a \in A} \mathbb{E}[V(\mathcal{S}) \mid \mathcal{S} \sim p(\cdot \mid s, a)]$$

- From Q-value function Q

$$\pi_Q(s) = \arg\max_{a \in A} Q(s, a)$$

Greed is optimal

Fact

Greedy policies from ***optimal*** metrics are optimal

- Optimal value function $V^\star$

$$V^\star(s) = \max_\pi V^\pi(s)$$

- Optimal Q-value function

$$Q^\star(s, a) = \max_\pi Q^\pi(s, a)$$

Value-based approach

1. ***Estimating*** ***reward metrics***

2. ***Deriving a policy from the metrics estimated***
   - The derived policy is optimal
     if the metrics satisfy some desirable property

- Note: usually assume MDPs are *finite*
  - Action set A and state set S are finite

Value-based approach

1. ***Estimating*** ***reward metrics***

   Key idea: Bellman optimality equation

2. ***Deriving a policy from the metrics estimated***
   - The derived policy is optimal
     if the metrics satisfy some desirable property

- Note: usually assume MDPs are *finite*
  - Action set A and state set S are finite

Bellman op

The action that maximizes
the cumulative discounted reward

For value functions

$$V(s) = \max_{a \in A} \mathbb{E}[\mathscr{R} + \gamma \cdot V(\mathscr{S}) \mid \mathscr{R} \sim r(\cdot \mid s, a), \mathscr{S} \sim p(\cdot \mid s, a)]$$

- V satisfies the equation iff V is optima
- The value of s is maximized by the "best" action a

- ***Goal of estimation:***
  finding V satisfying the above equation

Bellman optimality equation

For Q-value functions

$$Q(s,a) = \mathbb{E}[\mathscr{R} + \gamma \max\nolimits_{a' \in A} Q(\mathcal{S}, a') \mid \mathscr{R} \sim r(\cdot \mid s,a), \mathcal{S} \sim p(\cdot \mid s,a)]$$

- Q satisfies the equation iff Q is optimal
- The value of (s,a) is maximized when the "best" action a' is performed in the next state $\mathcal{S}$

- ***Goal of estimation:***
  finding Q satisfying the above equation

Approaches to RL problems

1. **Value-based:** estimating value/Q-value functions
   A. **Model-based**
      - Dynamic programming
   B. **Model-free**
      - Monte Carlo
      - Q-learning
2. **Policy-based:** estimating policies directly

Approaches to RL problems

1. **Value-based:** estimating value/Q-value functions
   A. ***Model-based***
      - Dynamic programming

   B. **Model-free**
      - Monte Carlo
      - Q-learning
2. **Policy-based:** estimating policies directly

$p(s' \mid s, a)$ is the probability of
how likely the state s' is reached from state s by action a

- Mathematical representation of knowledge about environments
- In MDPs, models correspond to probabilistic distribution p
- ***Model-based*** approaches assume probabilities assigned by p are ***known***
- ***Model-free*** approaches assume they are ***unknown***

Approaches to RL problems

1. **Value-based:** estimating value/Q-value functions
   A. **Model-based**
      - *Dynamic programming*

   B. **Model-free**
      - Monte Carlo
      - Q-learning
2. **Policy-based:** estimating policies directly

Dynamic programming

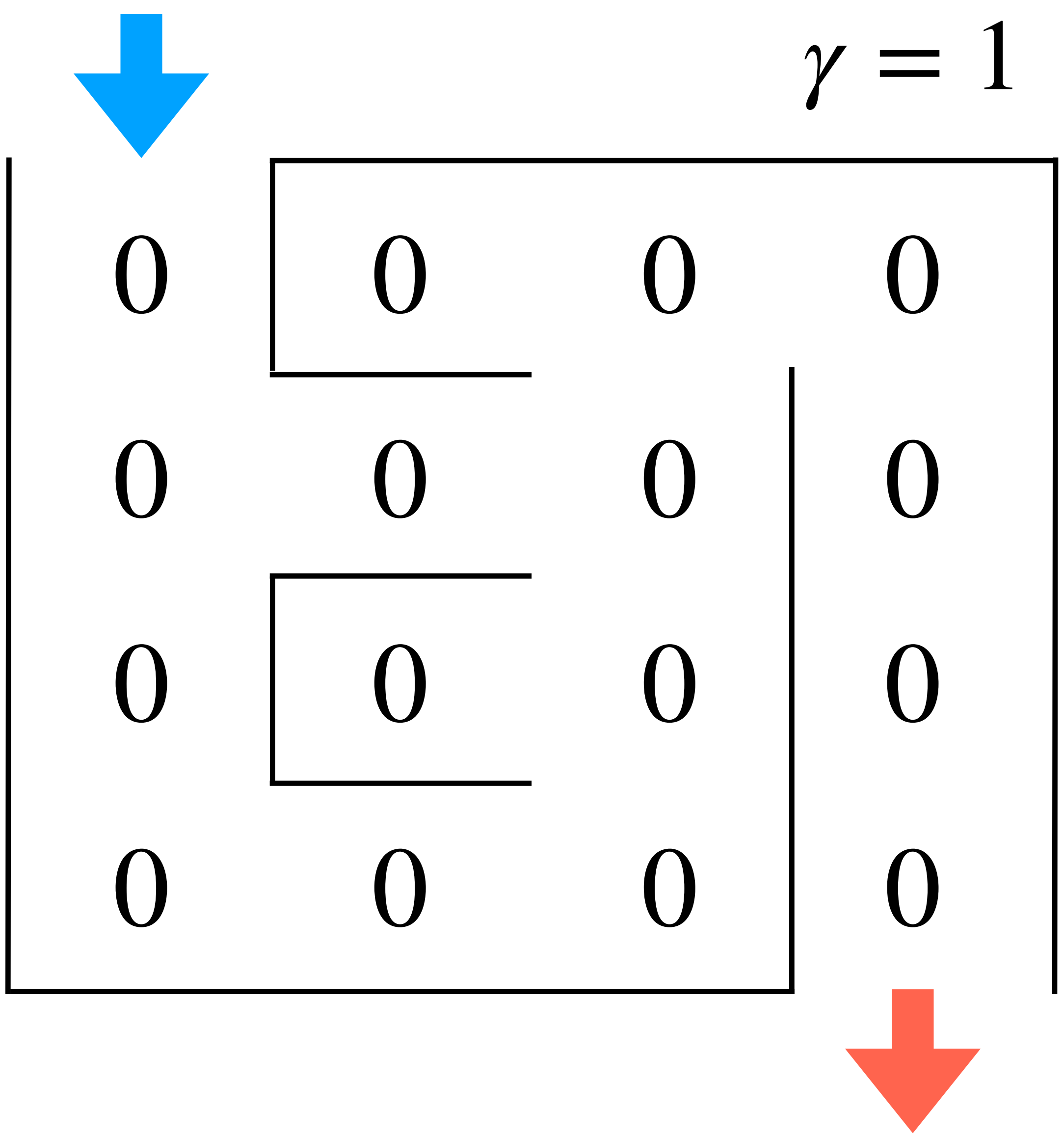
- Updating a value function by following the Bellman optimality equation
- Algorithm
  1. Initialize value function V (e.g., $V(s) := 0$ for any s)
  2. Repeat, for any $s \in S$,

$$V(s) := \max_{a \in A} \mathbb{E}[\mathscr{R} + \gamma \cdot V(\mathscr{S}) \mid \mathscr{R} \sim r(\cdot \mid s, a), \mathscr{S} = s(\cdot \mid s, a)]$$

  until all the update changes become small

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

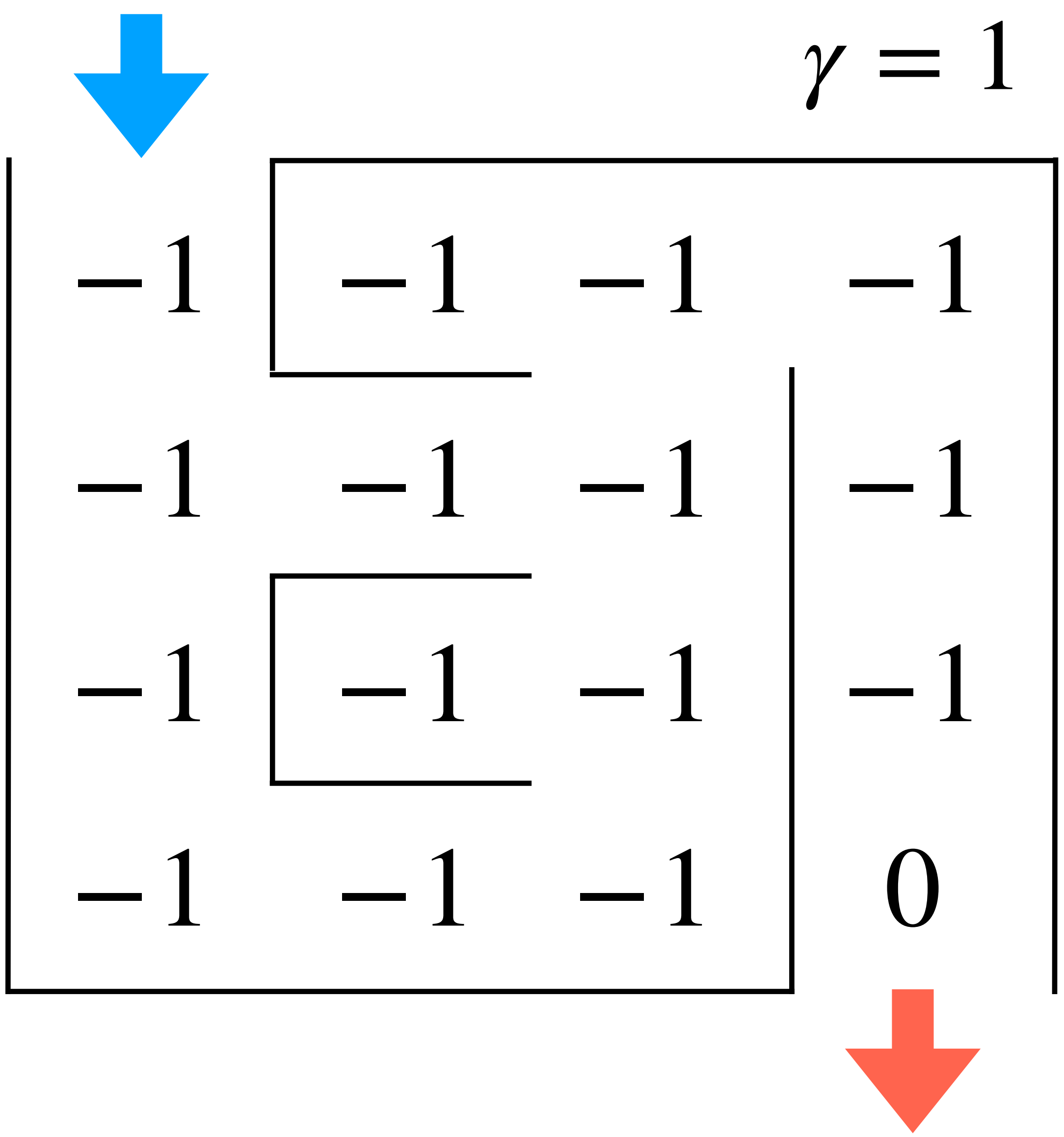
- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze

$\gamma = 1$



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

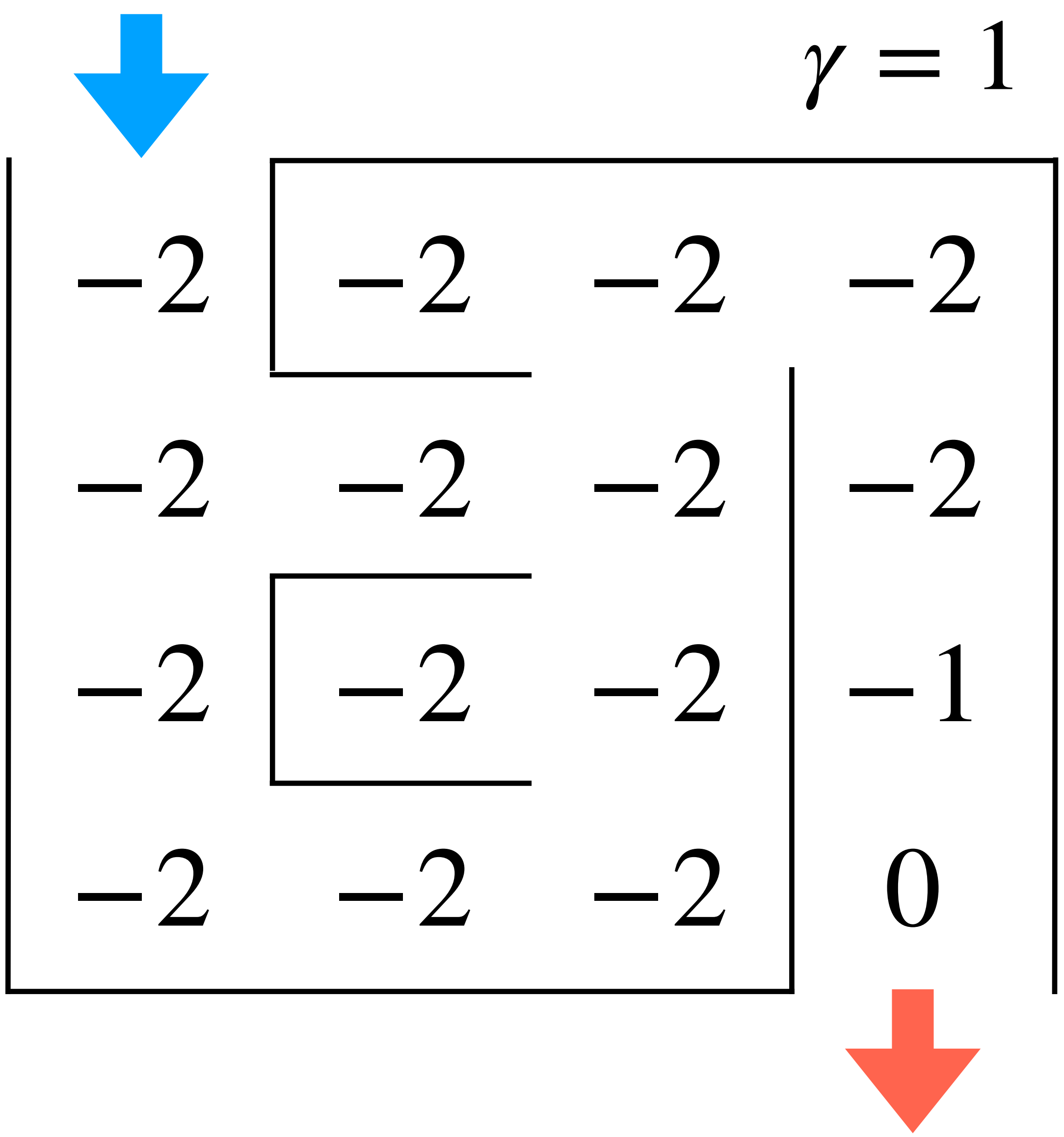
- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze

$\gamma = 1$



$$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$$

$$S \subseteq \mathbb{N} \times \mathbb{N}$$

$$p(s' \mid s, a) = \begin{cases} 1 & \text{(if } s \in G, s' = s\text{)} \\ 1 & \text{(if } s \notin G, (s, a) \in B, \\ & \quad s' = s + a\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$

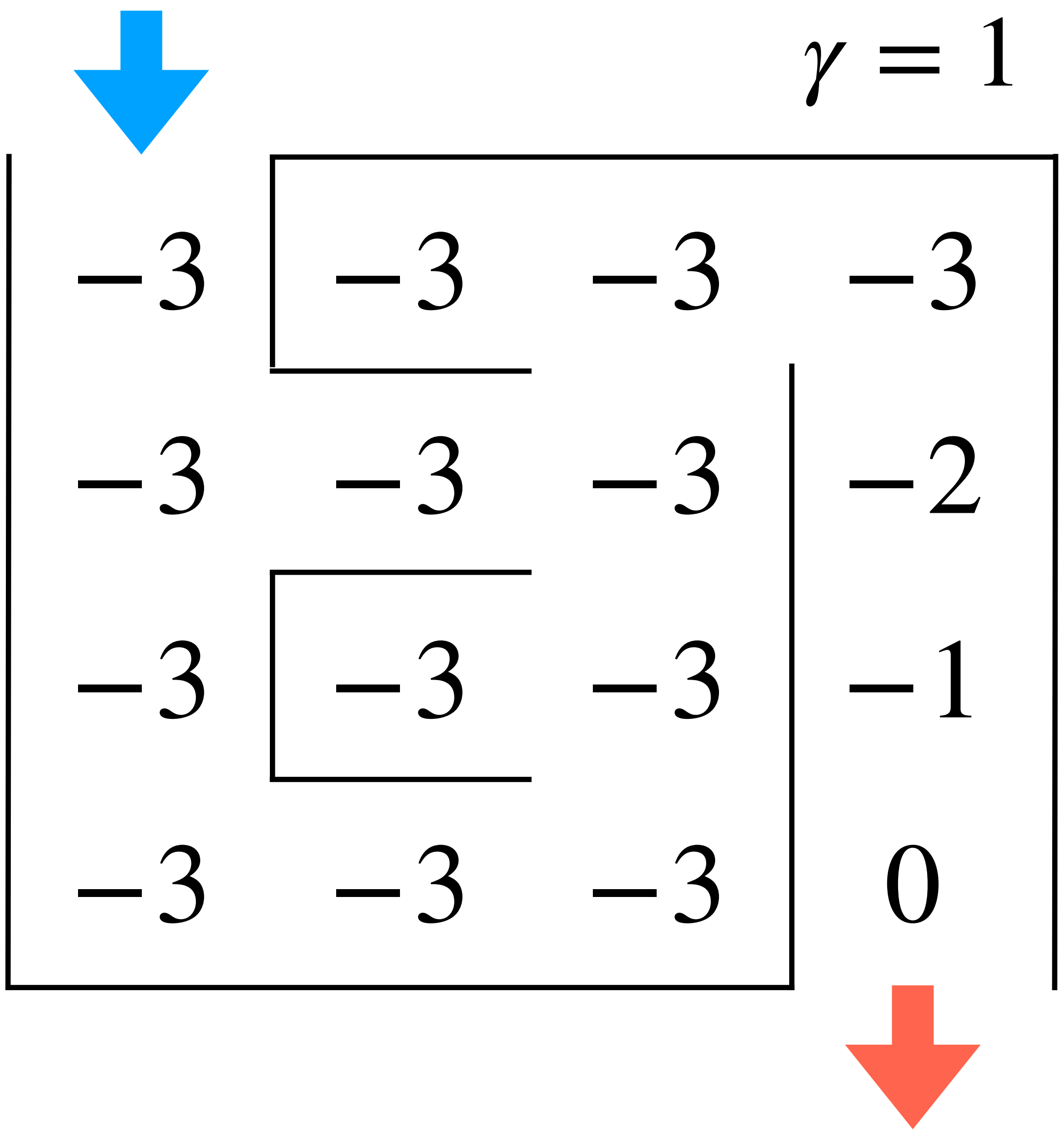
- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & \text{(if } s \in G, t = 0\text{)} \\ 1 & \text{(if } s \notin G, t = -1\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$

Running example

Maze

$\gamma = 1$



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

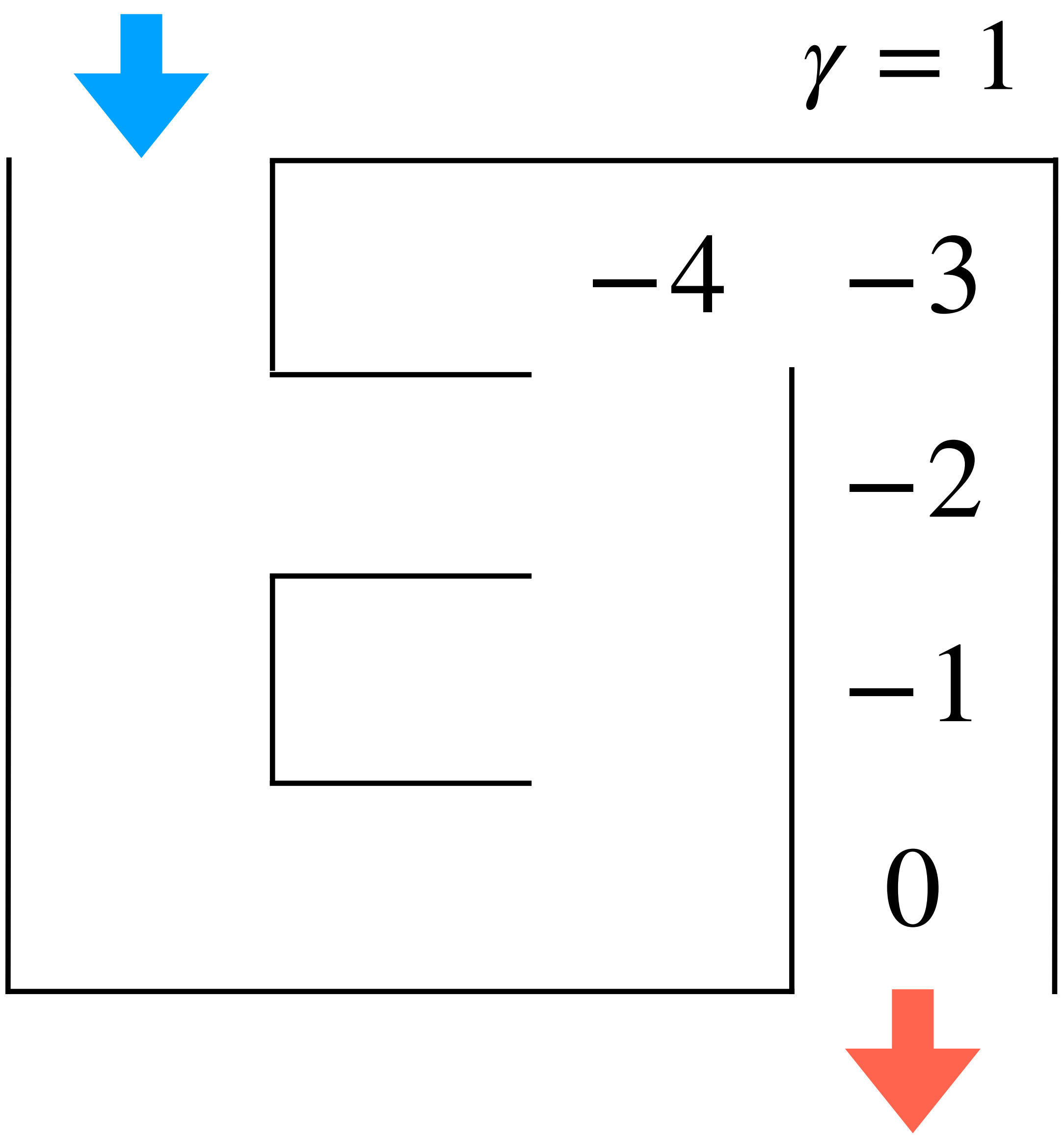
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$$

$$S \subseteq \mathbb{N} \times \mathbb{N}$$

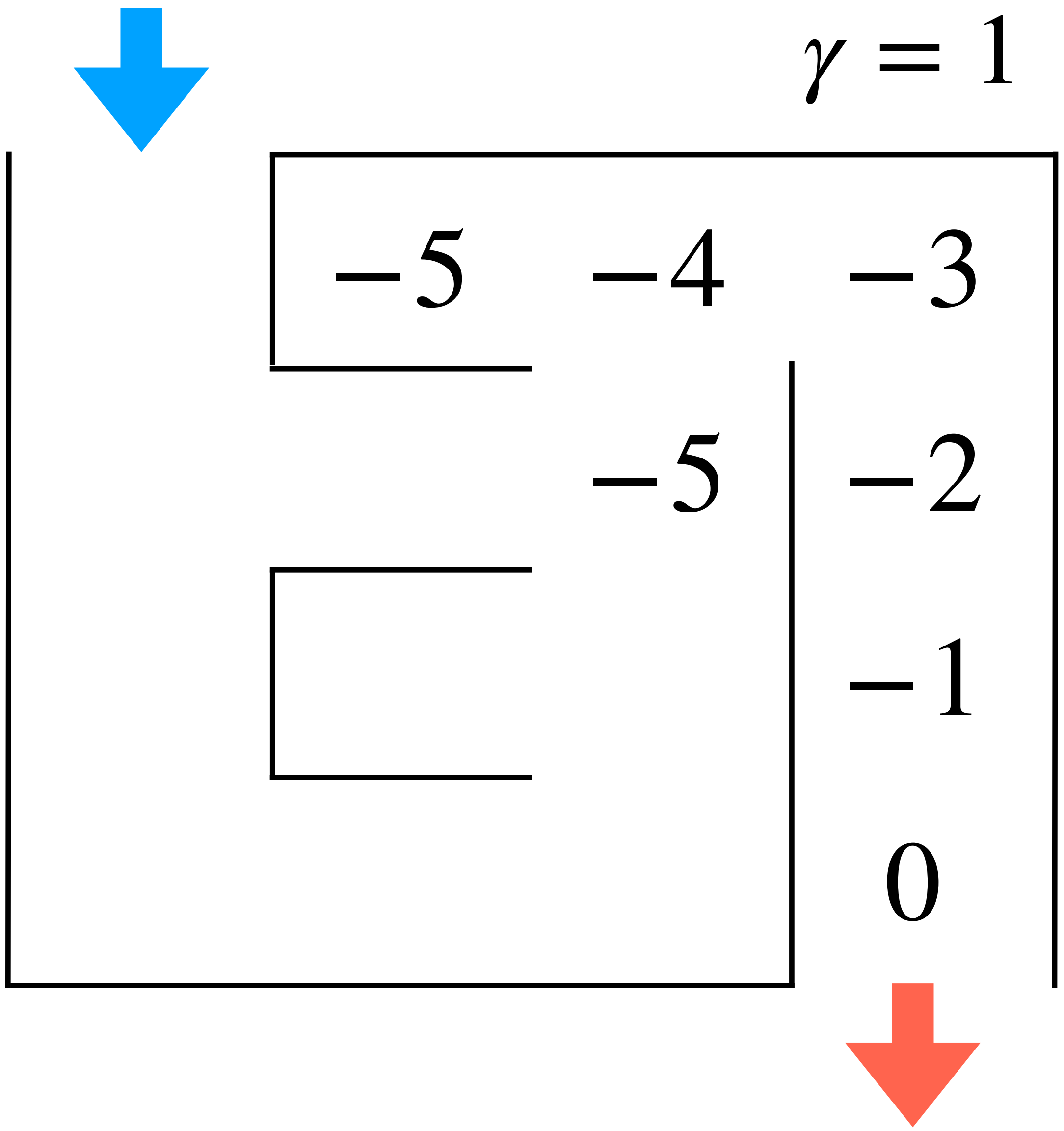
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

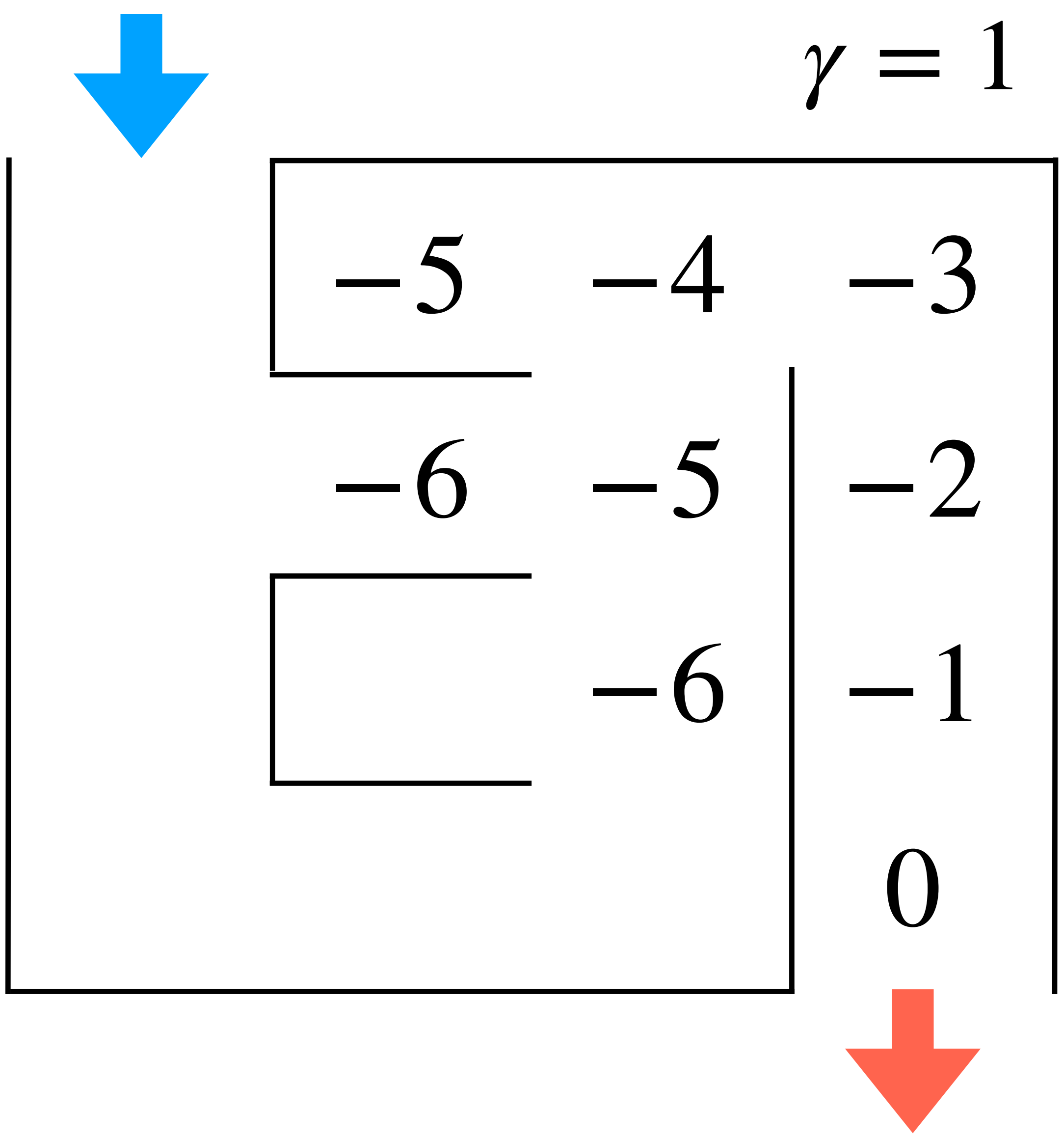
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s, a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

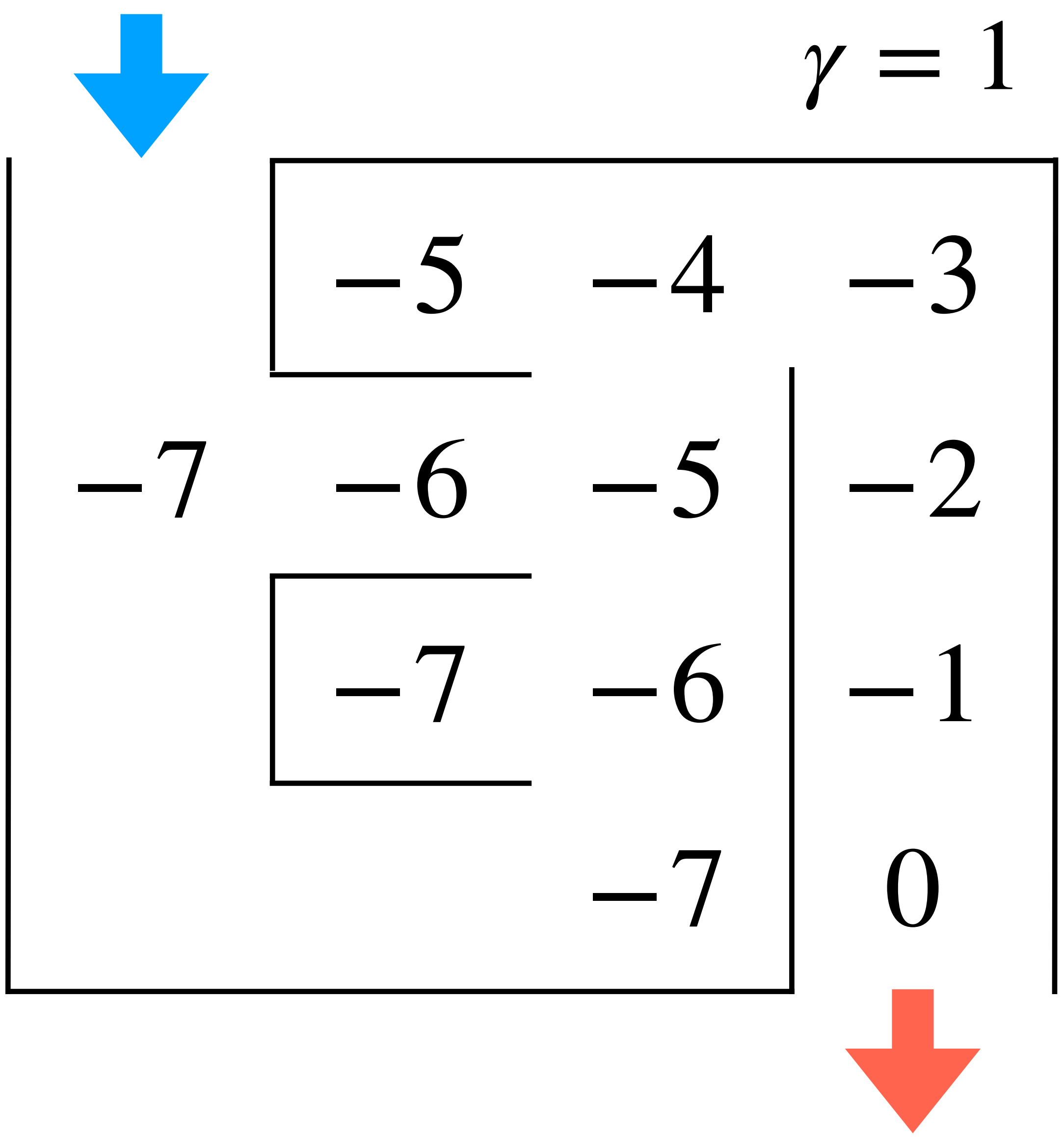
$$p(s' \mid s, a) = \begin{cases} 1 & \text{(if } s \in G, s' = s\text{)} \\ 1 & \text{(if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & \text{(if } s \in G, t = 0\text{)} \\ 1 & \text{(if } s \notin G, t = -1\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$

Running example

Maze



$A = \{(0,1), (0, -1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

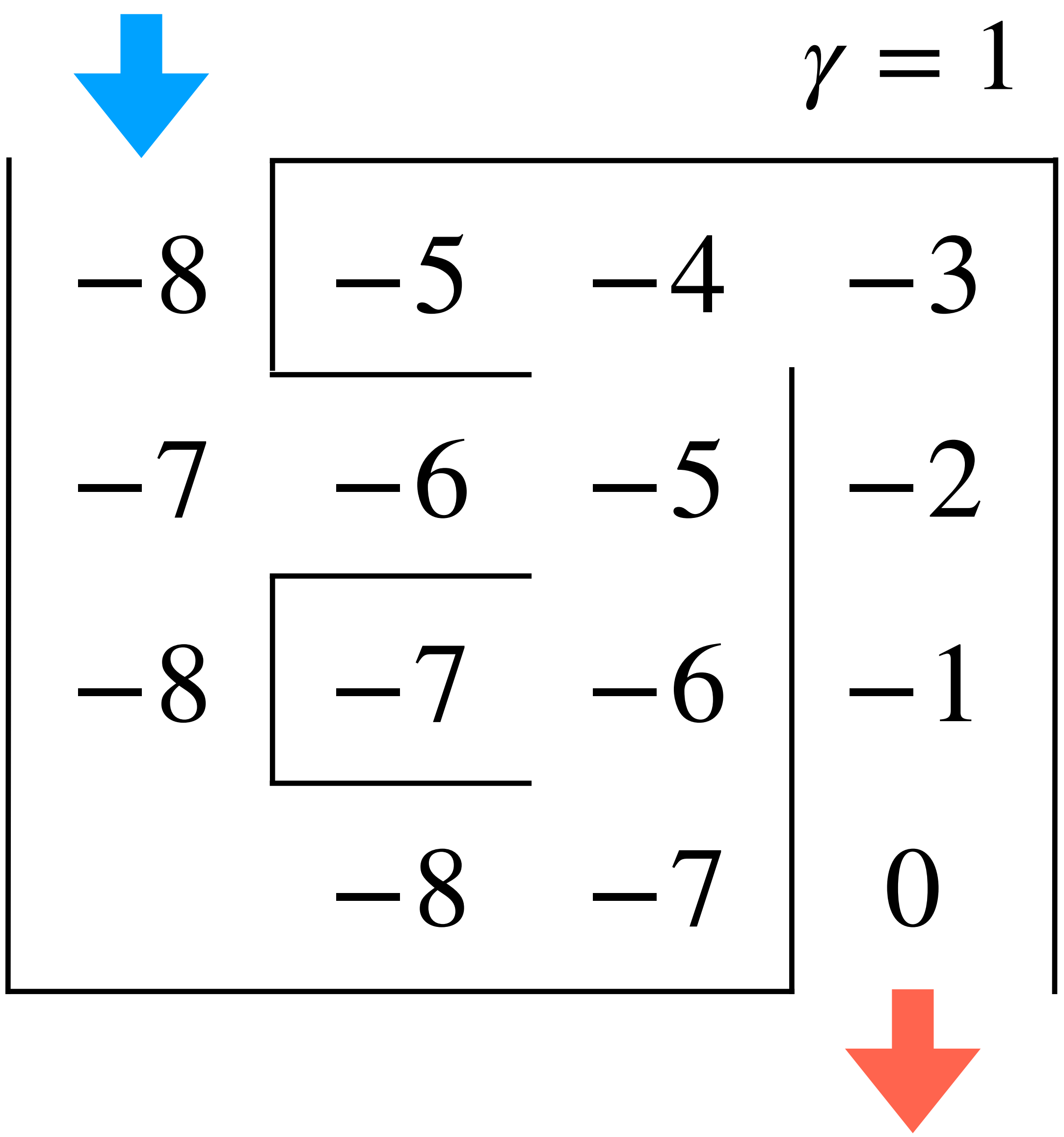
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

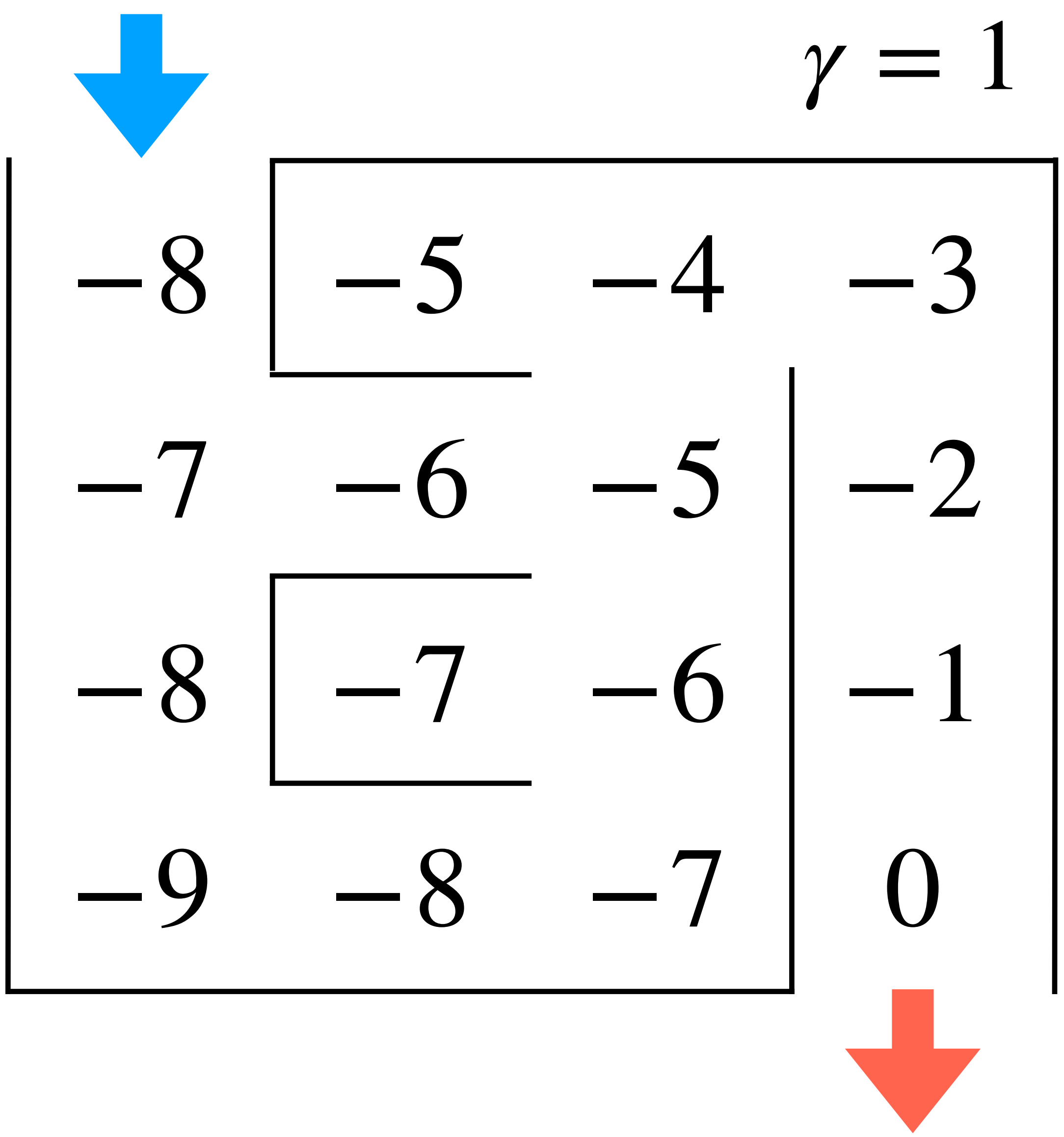
- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze

$\gamma = 1$



$$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$$

$$S \subseteq \mathbb{N} \times \mathbb{N}$$

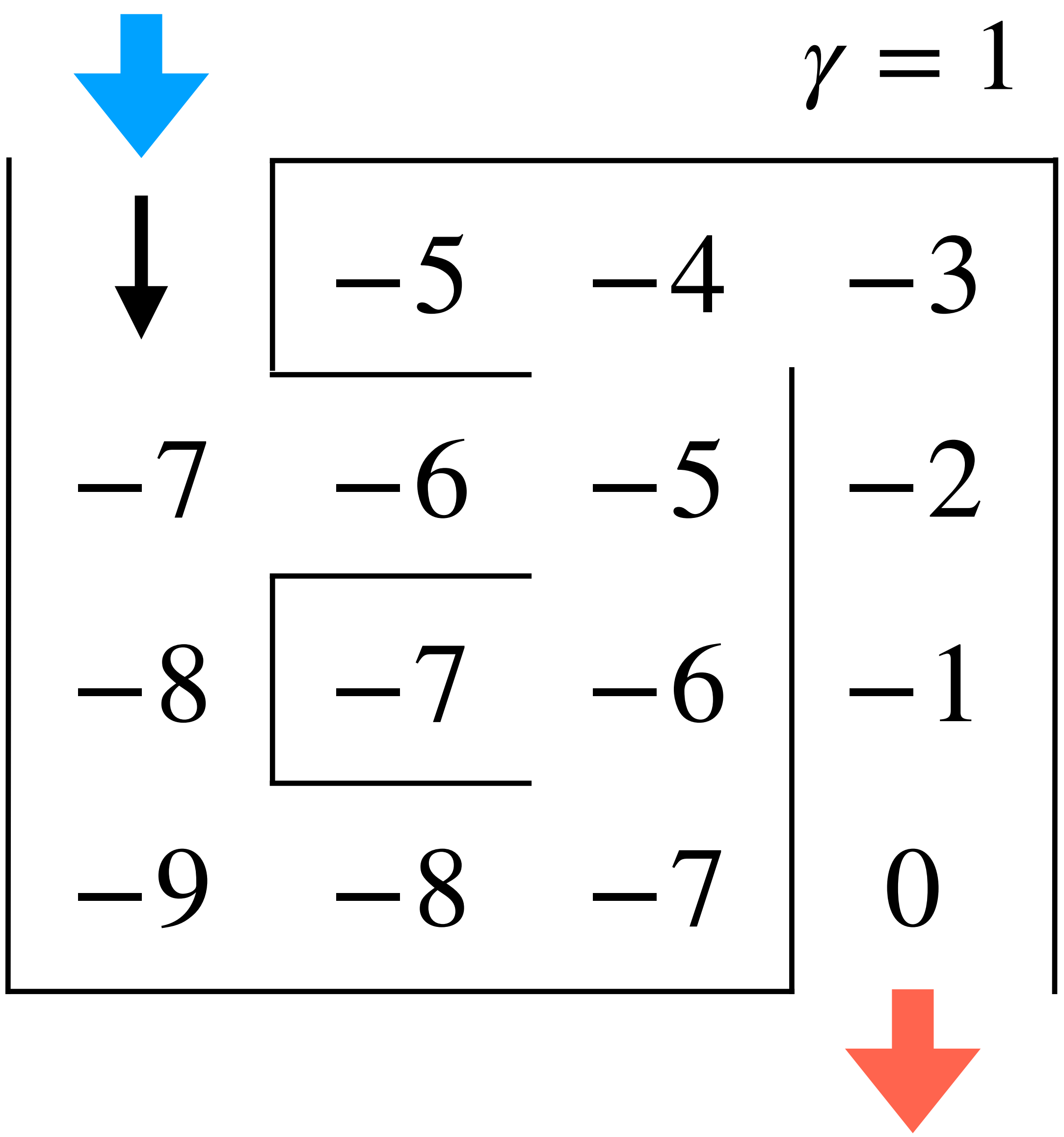
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0, -1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s, a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

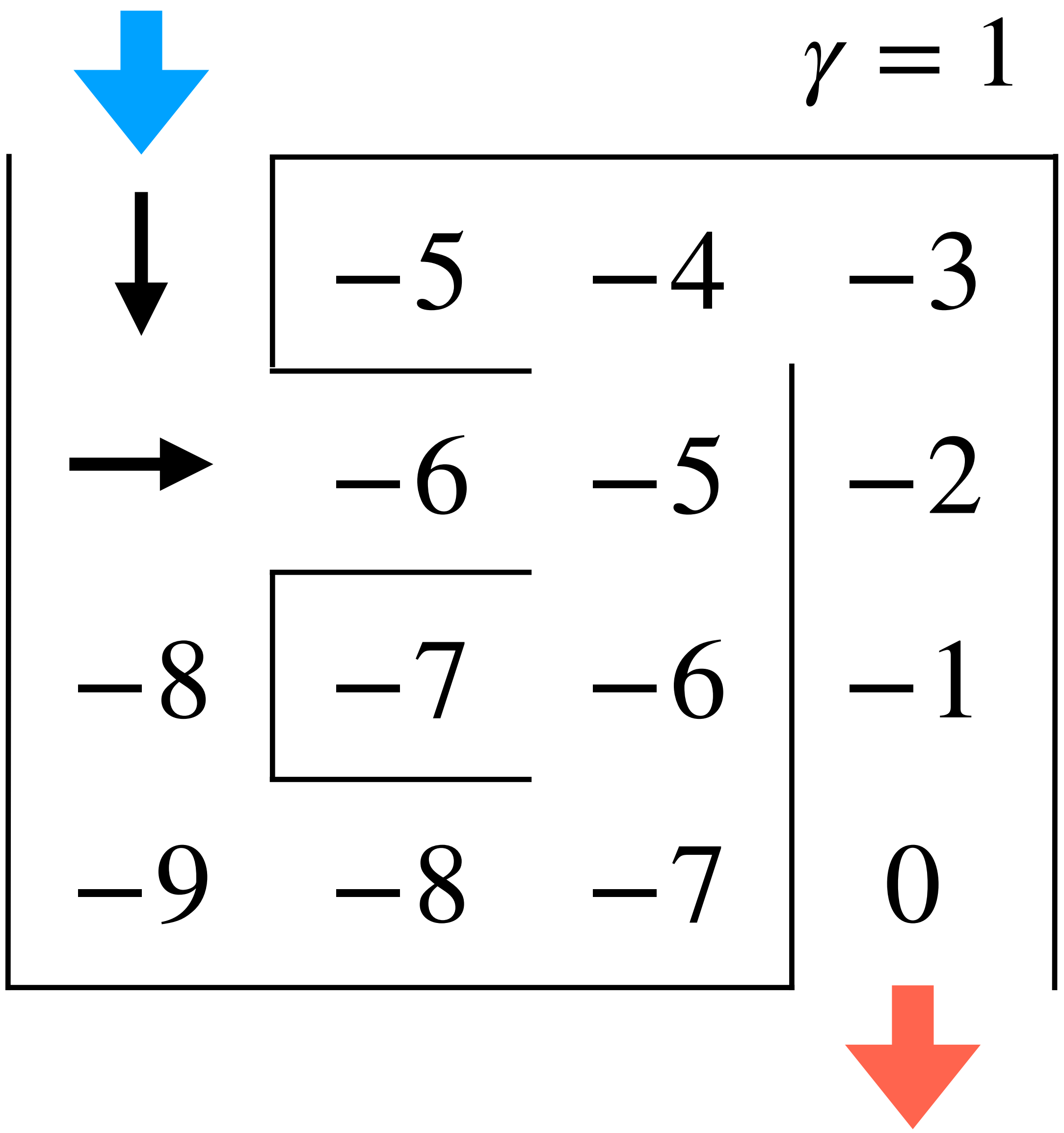
- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze

$\gamma = 1$



$$A = \{(0,1), (0, -1), (-1,0), (1,0)\}$$

$$S \subseteq \mathbb{N} \times \mathbb{N}$$

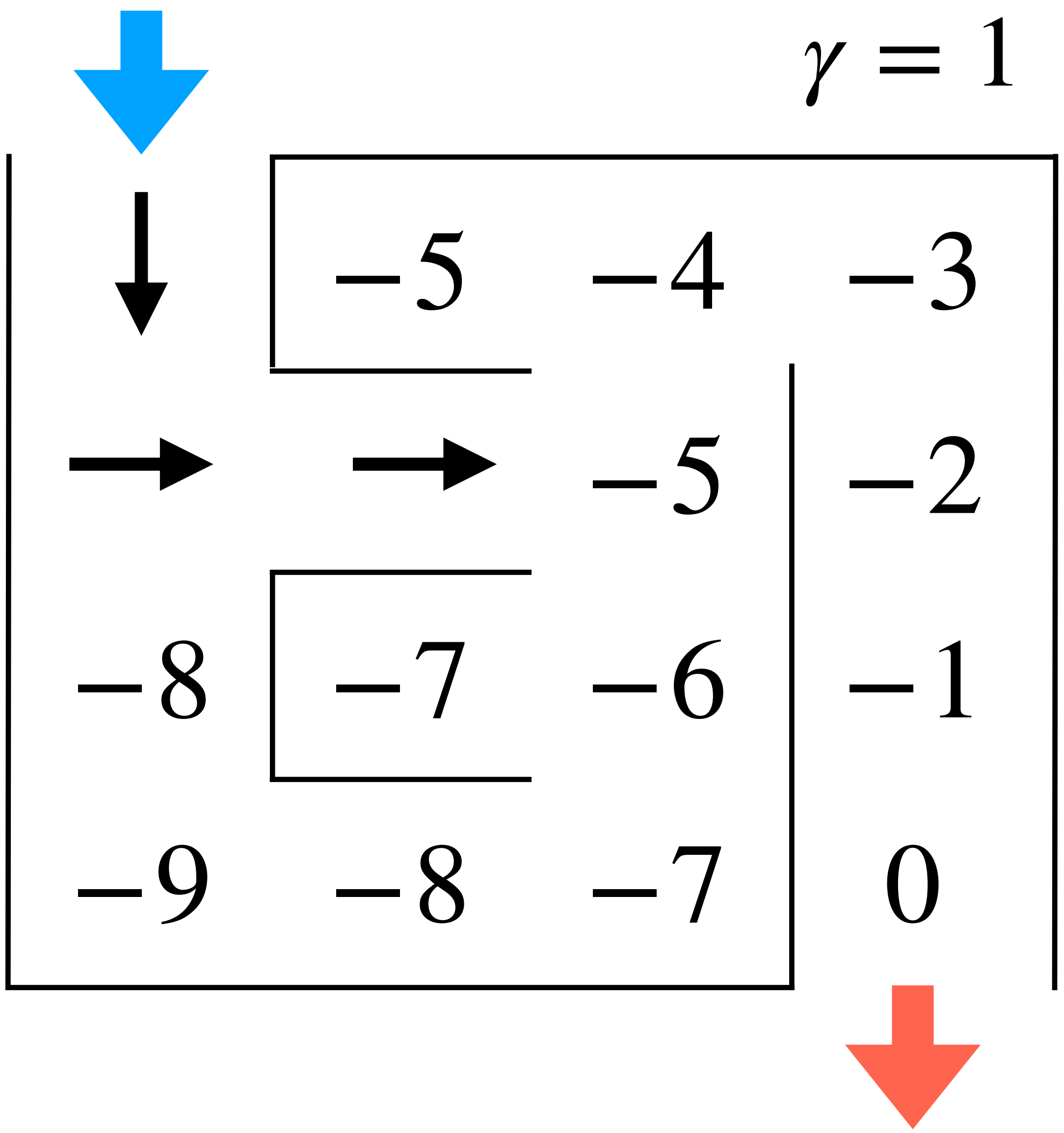
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0, -1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

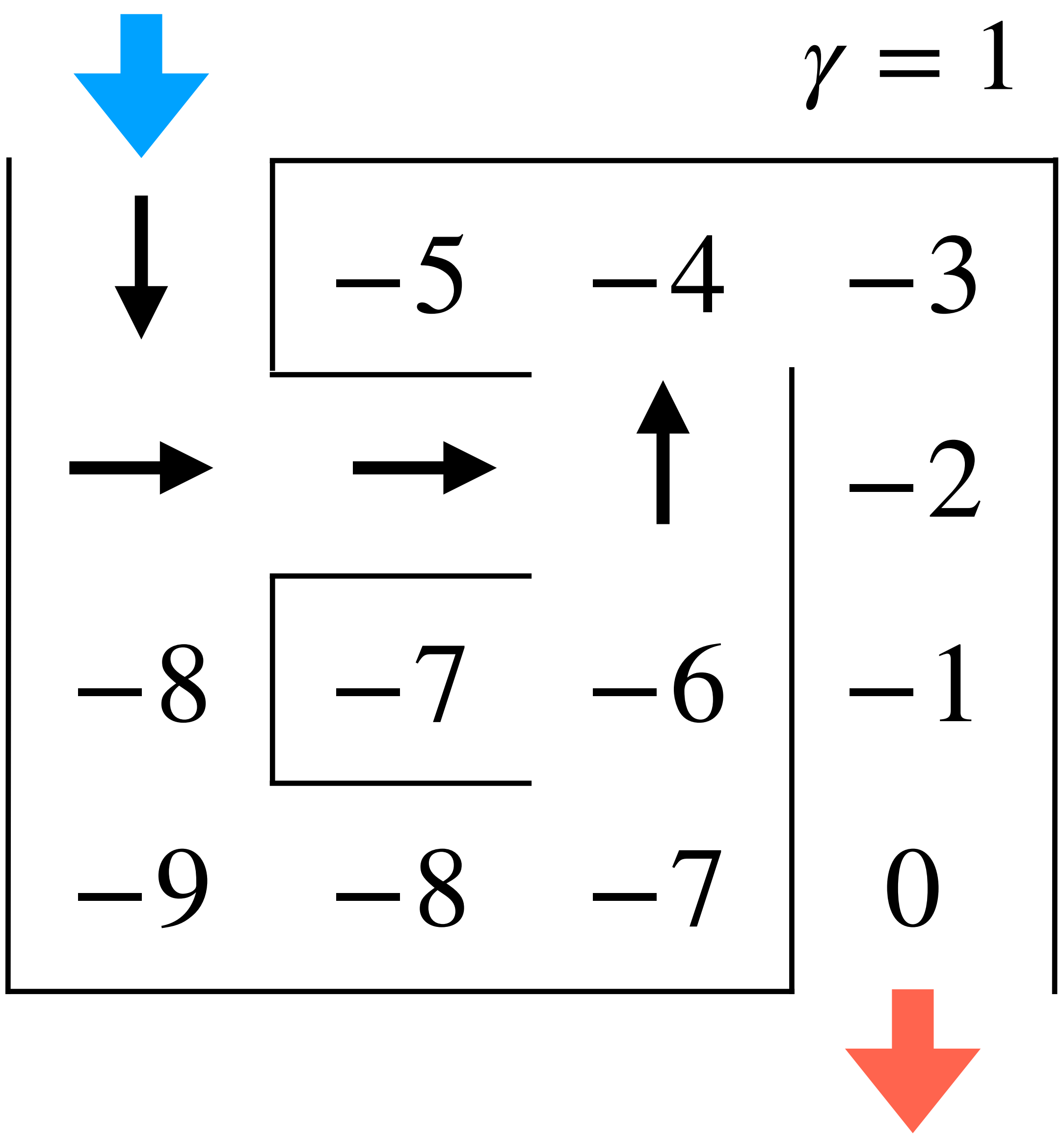
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

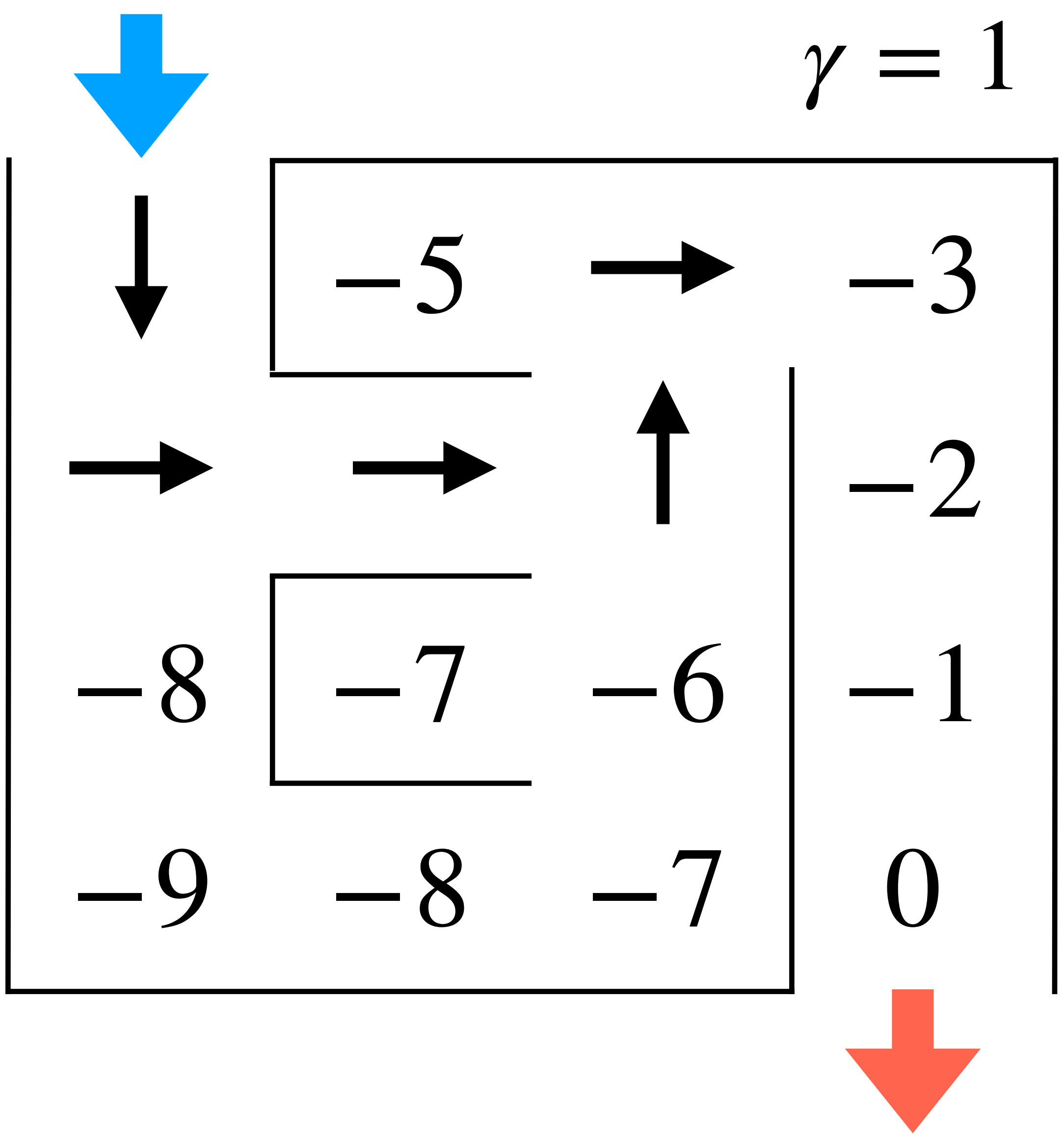
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0, -1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

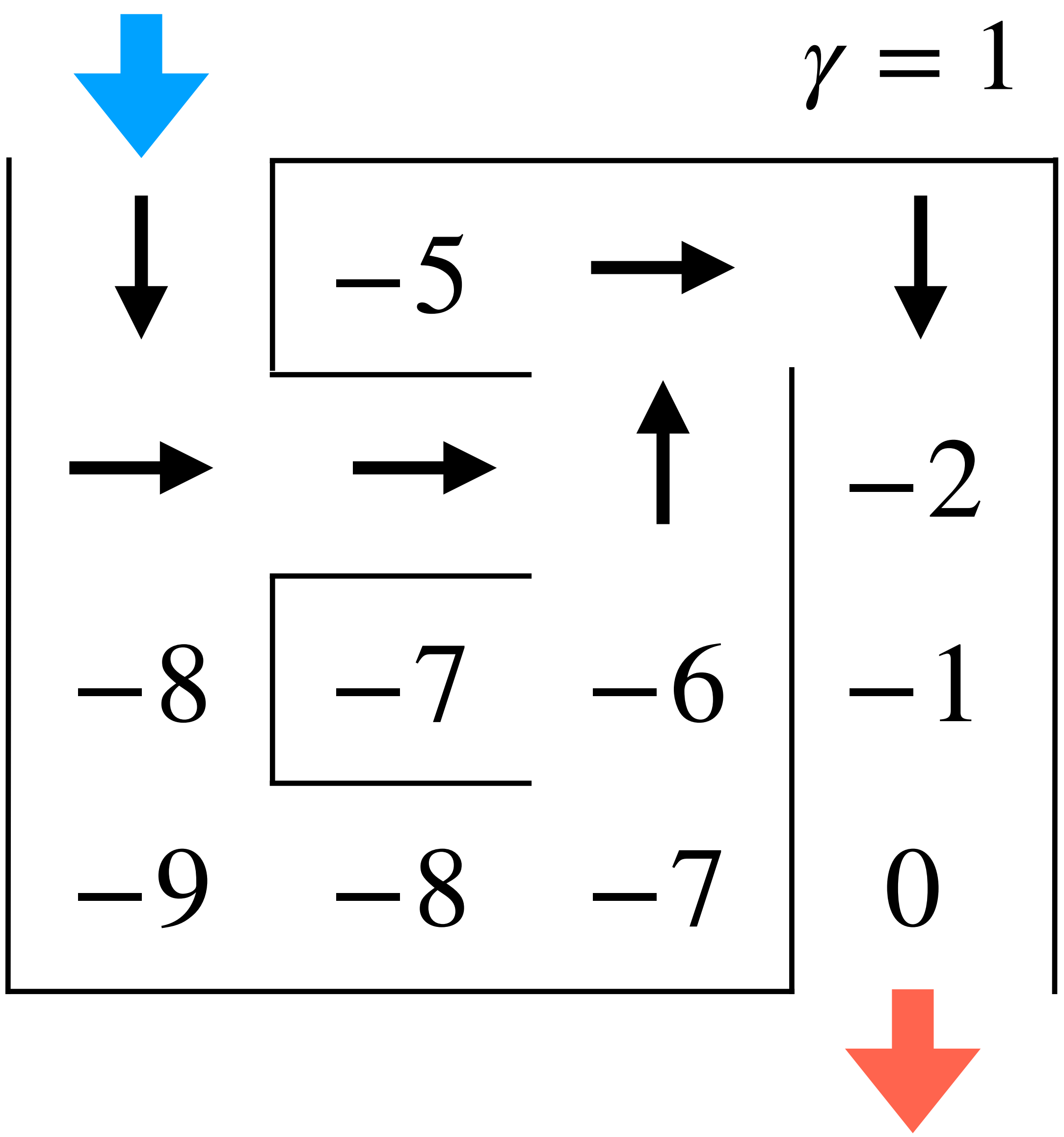
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s, a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

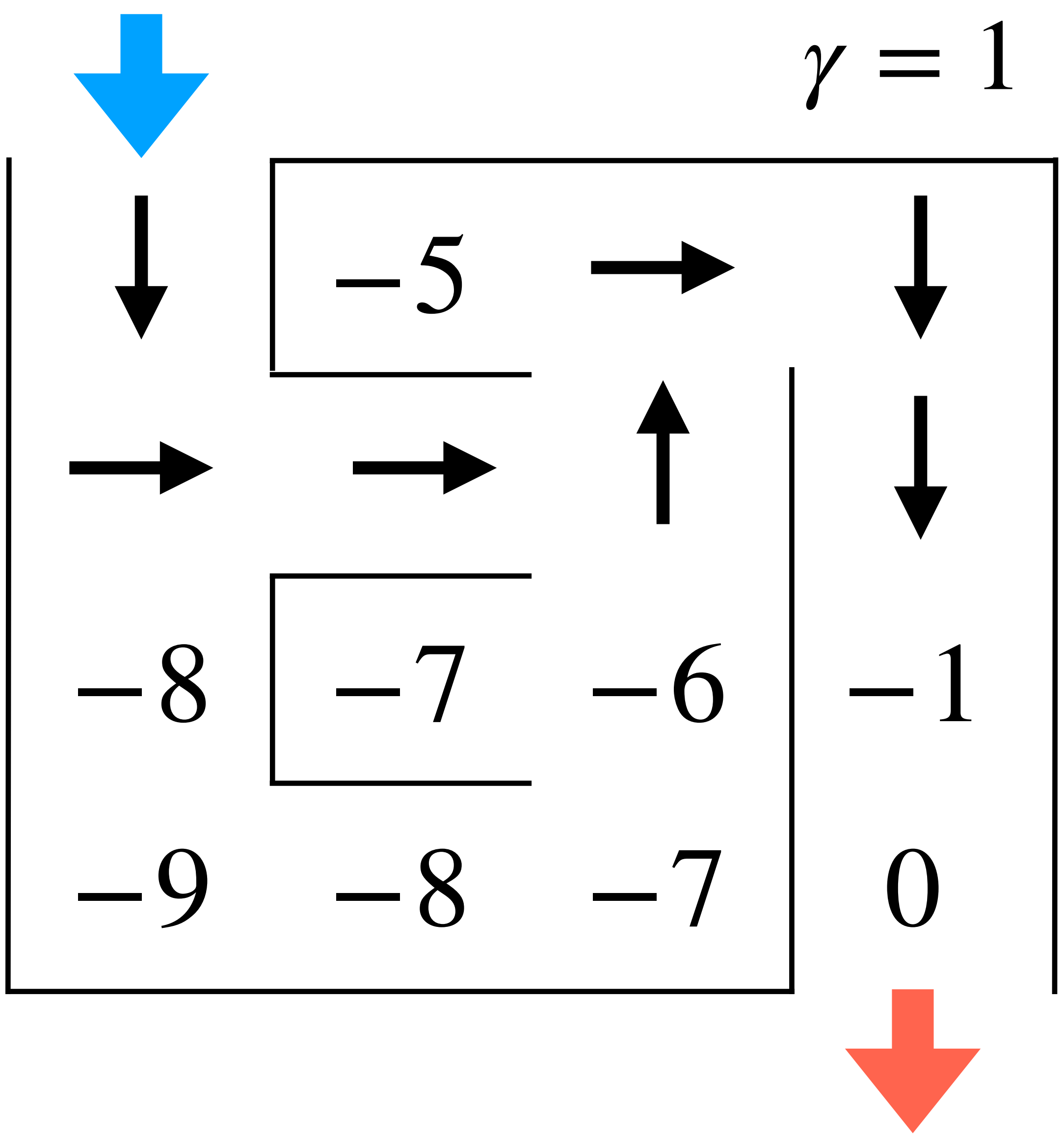
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s,a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$

$S \subseteq \mathbb{N} \times \mathbb{N}$

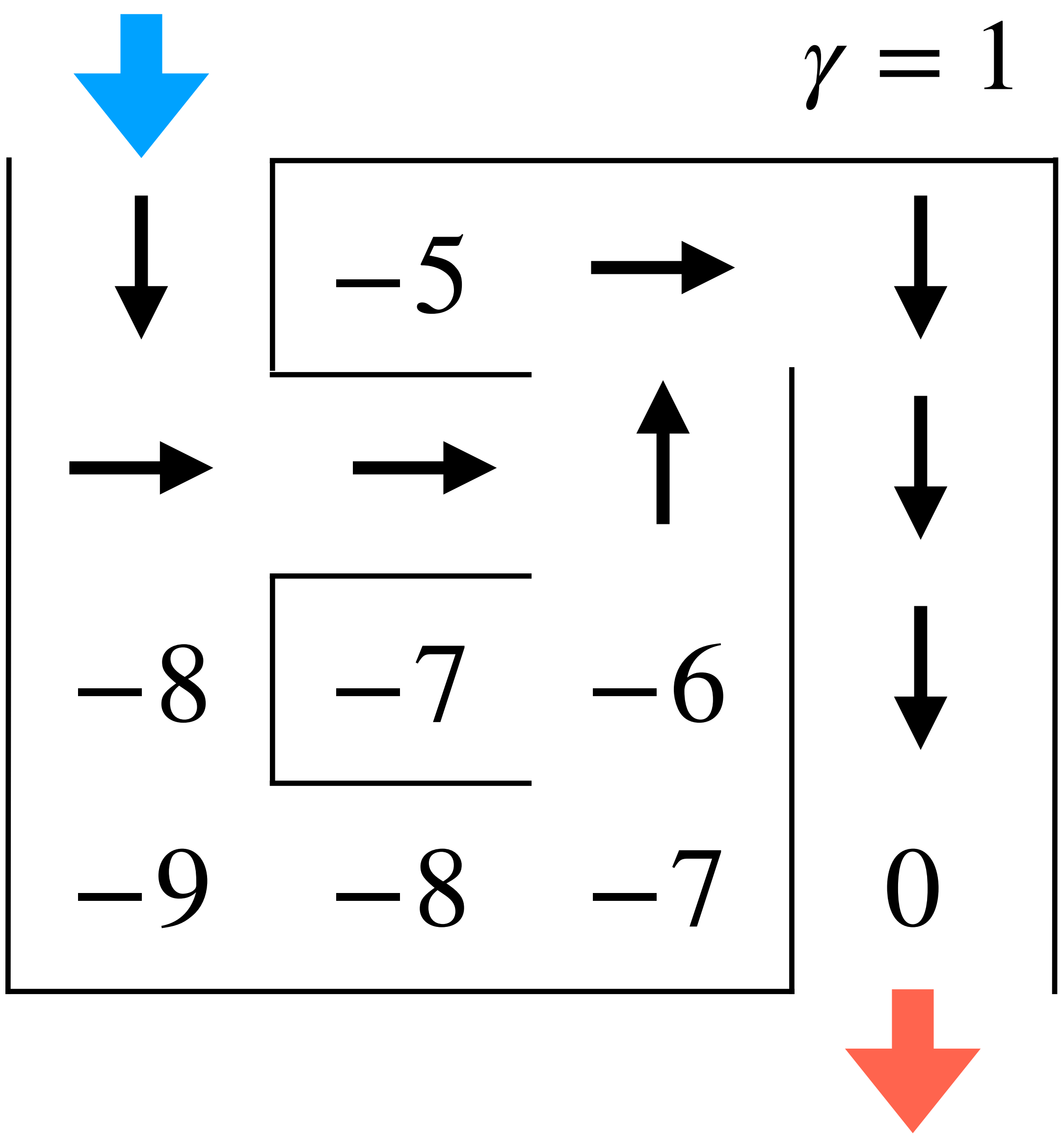
$$p(s' \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Running example

Maze



$$A = \{(0,1), (0,-1), (-1,0), (1,0)\}$$

$$S \subseteq \mathbb{N} \times \mathbb{N}$$

$$p(s' \,|\, s, a) = \begin{cases} 1 & (\text{if } s \in G, s' = s) \\ 1 & (\text{if } s \notin G, (s,a) \in B, \\ & \quad s' = s + a) \\ 0 & (\text{otherwise}) \end{cases}$$

- $G \subseteq \mathbb{N} \times \mathbb{N}$ is a set of goal positions
- $(s, a) \in B \subseteq S \times A$ indicates we can proceed along direction a at position s

$$r(t \mid s, a) = \begin{cases} 1 & (\text{if } s \in G, t = 0) \\ 1 & (\text{if } s \notin G, t = -1) \\ 0 & (\text{otherwise}) \end{cases}$$

Property

The dynamic programming approach can find an optimal policy by iterating the update infinitely