

# **Markov Decision Processes**

Chen-Yu Wei

# Sequence of Actions

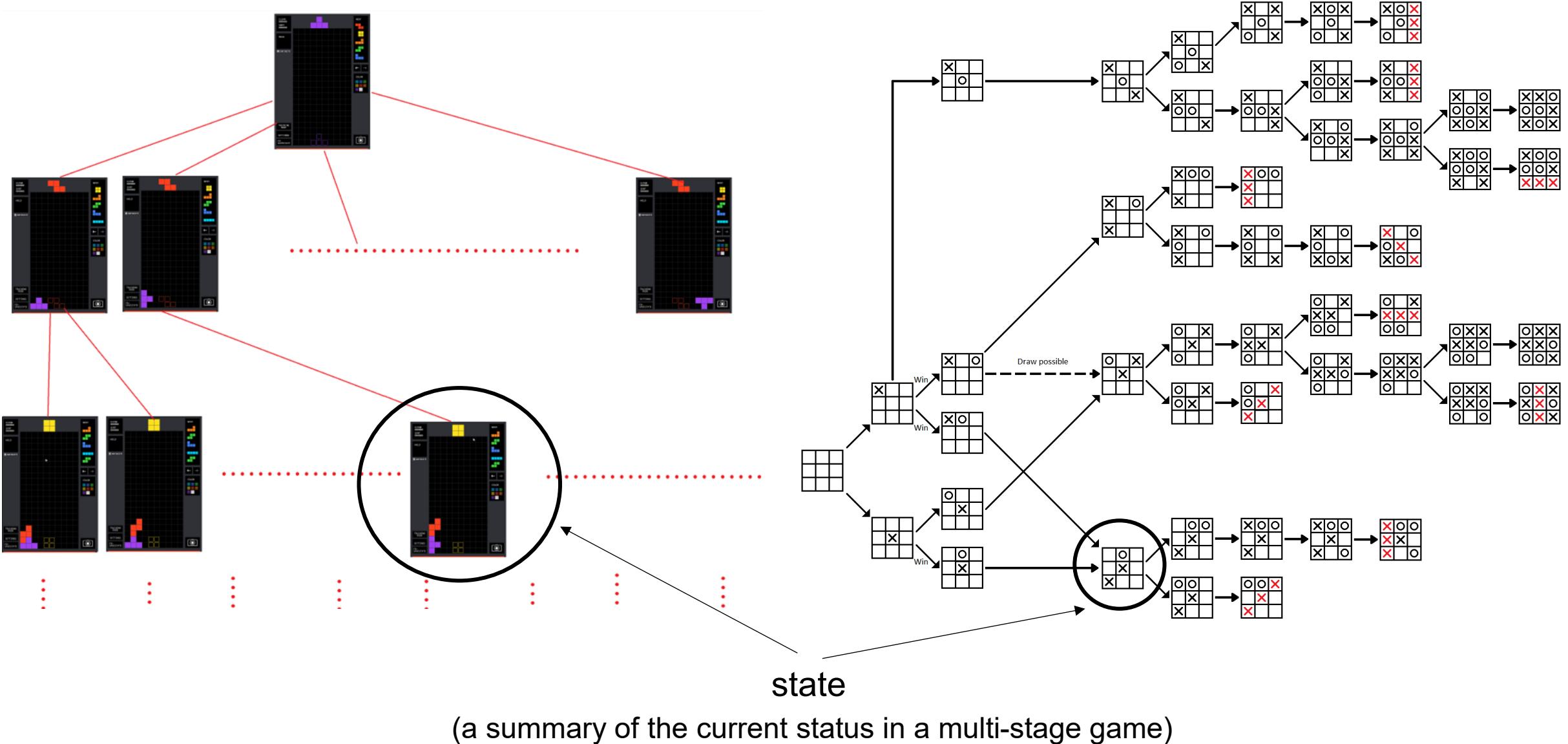


To win the game, the learner has to take a sequence of actions  $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_H$ .

The effect of a particular action may not be revealed instantaneously.

- Some effect may be revealed instantaneously
- Some may be revealed later

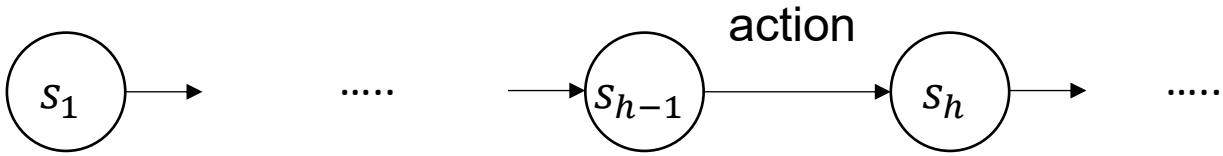
# Sequence of Actions



# Sequence of Actions

- The number of possible combinations of actions grows **exponentially** with the length of the sequence.
- We would like to decompose the problem so that every single decision in the sequence is easy to make.
- **State:** a **summary** of the **status of the world** and the **progress of the learner**, so that all future decisions can only depend on the state and not on everything else.
  - Games (Go, Chess): To decide future moves, the player only need the current board configuration.
  - Robot navigation to a goal: only need the current position and not the exact path reaching the current position.
  - Inventory management: only need the current inventory level, and not the sequence of past sales.

# Sequence of Actions



Like a sequential contextual bandit problem – except that future contexts depends on the learner's past decisions.

# Interaction Protocol (Episodic Setting)

For episode  $t = 1, 2, \dots, T$ :

$$h \leftarrow 1$$

Environment generates initial state  $s_{t,1}$

While episode  $t$  has not ended:

Learner chooses an action  $a_{t,h}$

Learner observes instantaneous reward  $r_{t,h}$  with  $\mathbb{E}[r_{t,h}] = R(s_{t,h}, a_{t,h})$

Environment generates next state  $s_{t,h+1} \sim P(\cdot | s_{t,h}, a_{t,h})$

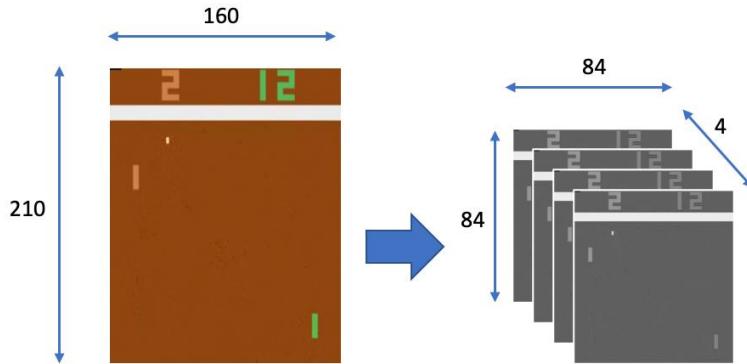
$$h \leftarrow h + 1$$

**Markov assumption:**

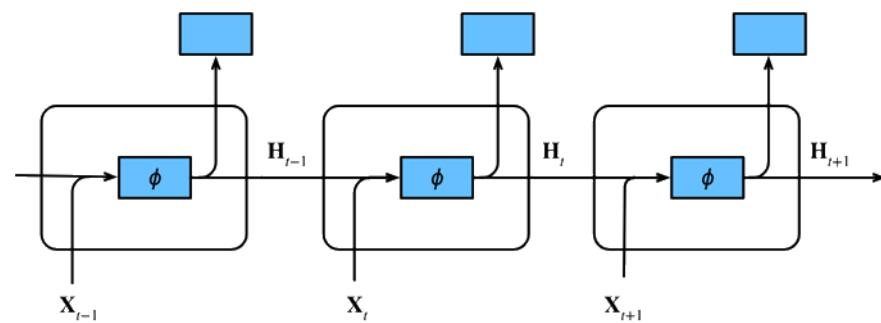
$r_{t,h}$  and  $s_{t,h+1}$  are conditionally independent of  $(s_{t,1}, a_{t,1}, \dots, s_{t,h-1}, a_{t,h-1})$  given  $s_{t,h}$

Goal: maximize  $\sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$

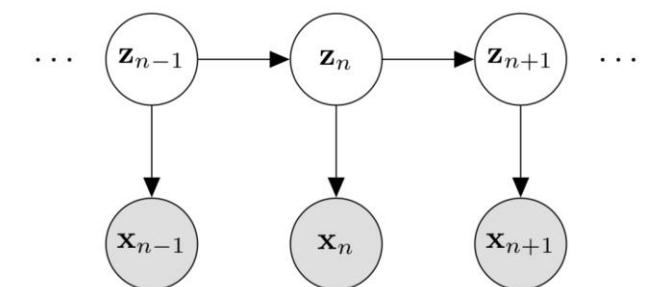
# From Observations to States



Stacking recent observations



Recurrent neural network

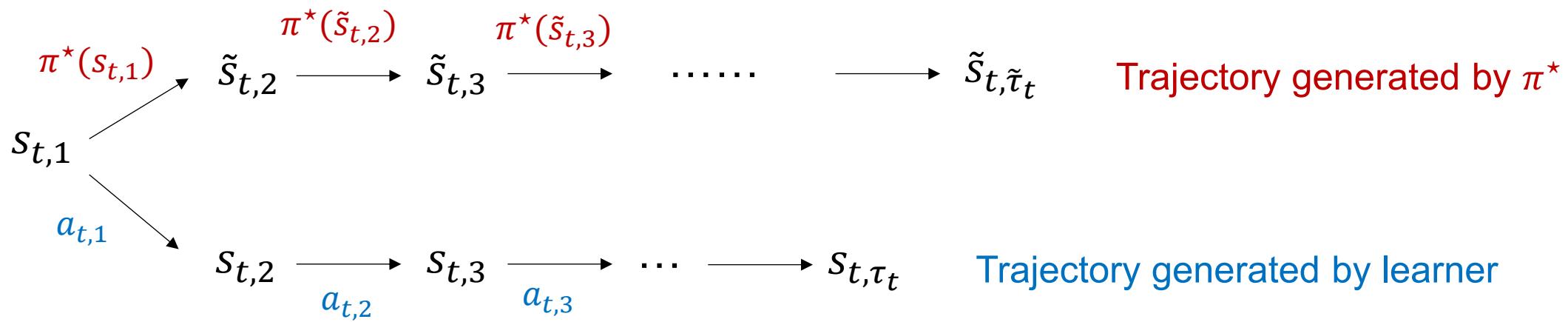


Hidden Markov model

# Regret (Episodic Setting)

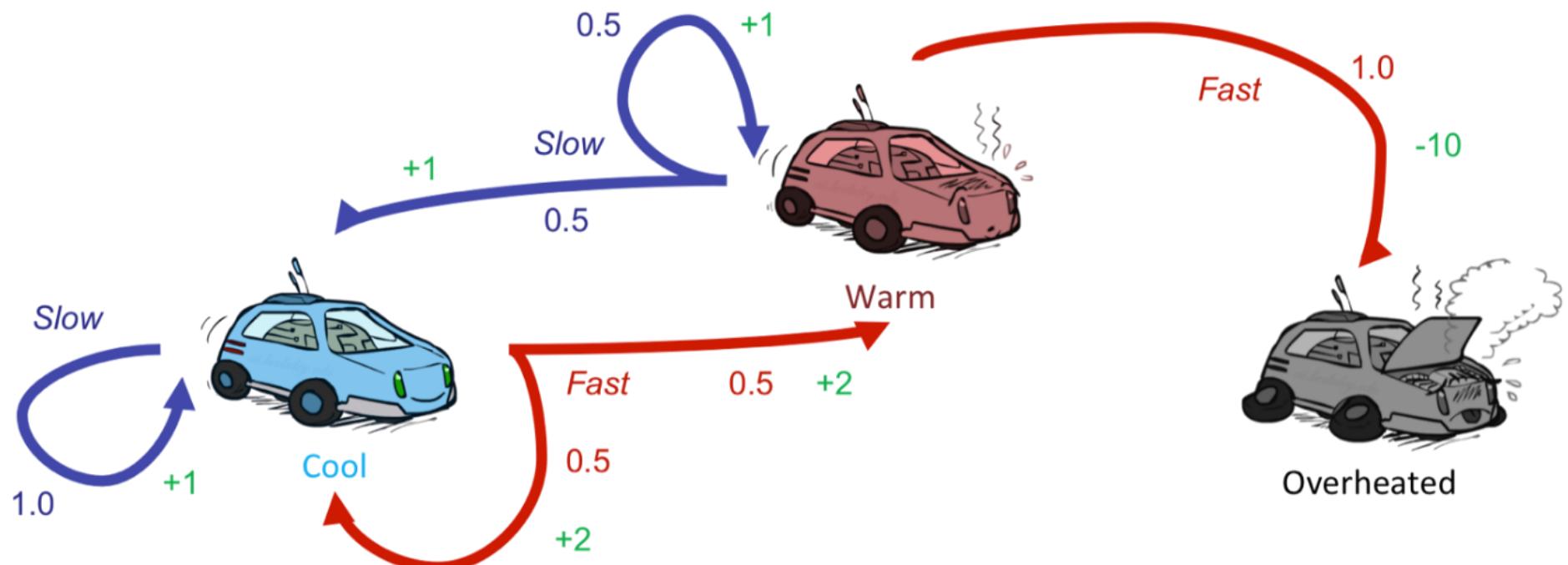
$$\text{Regret} = \max_{\pi^*} \mathbb{E}^{\pi^*} \left[ \sum_{t=1}^T \sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^*(\tilde{s}_{t,h})) \right] - \sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$

Benchmark



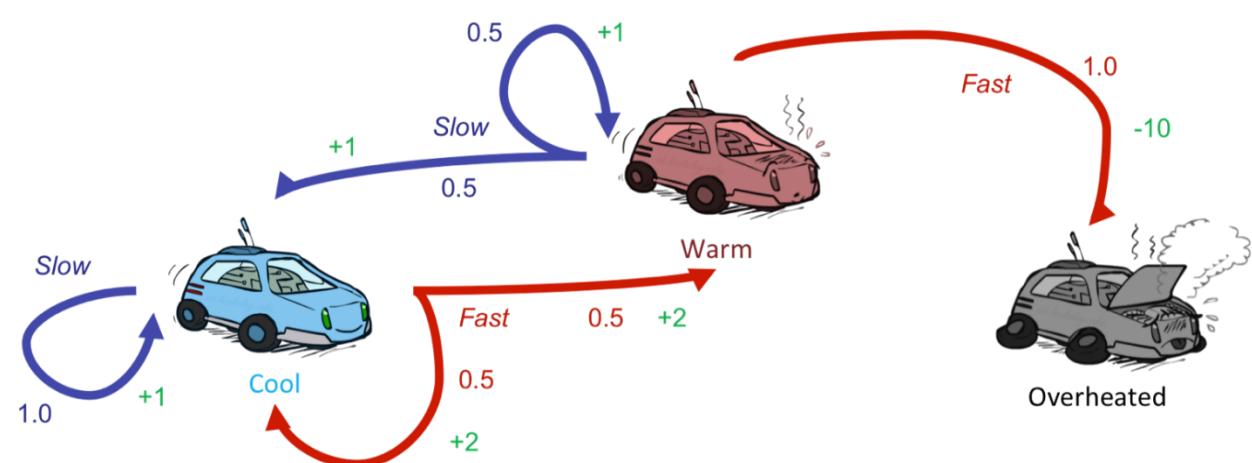
# Example: Racing

- A robot car wants to travel far, quickly
- Three states: Cool, Warm, Overheated
- Two actions: Slow, Fast
- Going faster gets double reward



# Example: Racing

$s$	$a$	$s'$	$P(s' s, a)$	$R(s, a)$
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	-10
	(end)		1.0	0



# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon
- Performance Metric
  - Total Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

Horizon = Length of an episode

# Interaction Protocols (1/2): Fixed-Horizon

Horizon length is a fixed number  $H$

$h \leftarrow 1$

Observe initial state  $s_1 \sim \rho$

**While  $h \leq H$ :**

    Choose action  $a_h$

    Observe reward  $r_h$  with  $\mathbb{E}[r_h] = R(s_h, a_h)$

    Observe next state  $s_{h+1} \sim P(\cdot | s_h, a_h)$

**Examples:** games with a fixed number of time

# Interaction Protocols (2/2): Variable-Horizon

The learner interacts with the environment until reaching **terminal states**  $\mathcal{T} \subset \mathcal{S}$

$h \leftarrow 1$

Observe initial state  $s_1 \sim \rho$

**While**  $s_h \notin \mathcal{T}$ :

    Choose action  $a_h$

    Observe reward  $r_h$  with  $\mathbb{E}[r_h] = R(s_h, a_h)$

    Observe next state  $s_{h+1} \sim P(\cdot | s_h, a_h)$

$h \leftarrow h + 1$

**Examples:** video games, robotics tasks, personalized recommendations, etc.

# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon
- Performance Metric
  - Total Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

Horizon = Length of an episode

# Performance Metric

$\tau$ : the step where the episode ends

**Total Reward:**

$$\sum_{h=1}^{\tau} r_h$$

**Discounted Total Reward:**

$$\sum_{h=1}^{\tau} \gamma^{h-1} r_h$$

$\gamma \in [0,1)$ : discount factor

There is a potential mismatch between our ultimate goal and what we optimized.

# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon
- Performance Metric
  - Total Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

# Policy for MDPs

## Markov Policy

$$\begin{aligned} a_h &\sim \pi_{\textcolor{red}{h}}(\cdot | s_h) \\ a_h &= \pi_{\textcolor{red}{h}}(s_h) \end{aligned} \quad \leftarrow$$

For **fixed-horizon** setting, there exists an optimal policy in this class

## Stationary Policy

$$\begin{aligned} a_h &\sim \pi(\cdot | s_h) \\ a_h &= \pi(s_h) \end{aligned} \quad \leftarrow$$

For **infinite-horizon/goal-oriented** settings, there exists an optimal policy in this class

Markov Policy = Stationary Policy where the state is augmented with **the timestep**.

A stationary policy specifies

$$\pi(\text{Slow} \mid \text{Cool})$$

$$\pi(\text{Fast} \mid \text{Cool})$$

$$\pi(\text{Slow} \mid \text{Warm})$$

$$\pi(\text{Fast} \mid \text{Warm})$$

A Markov policy specifies

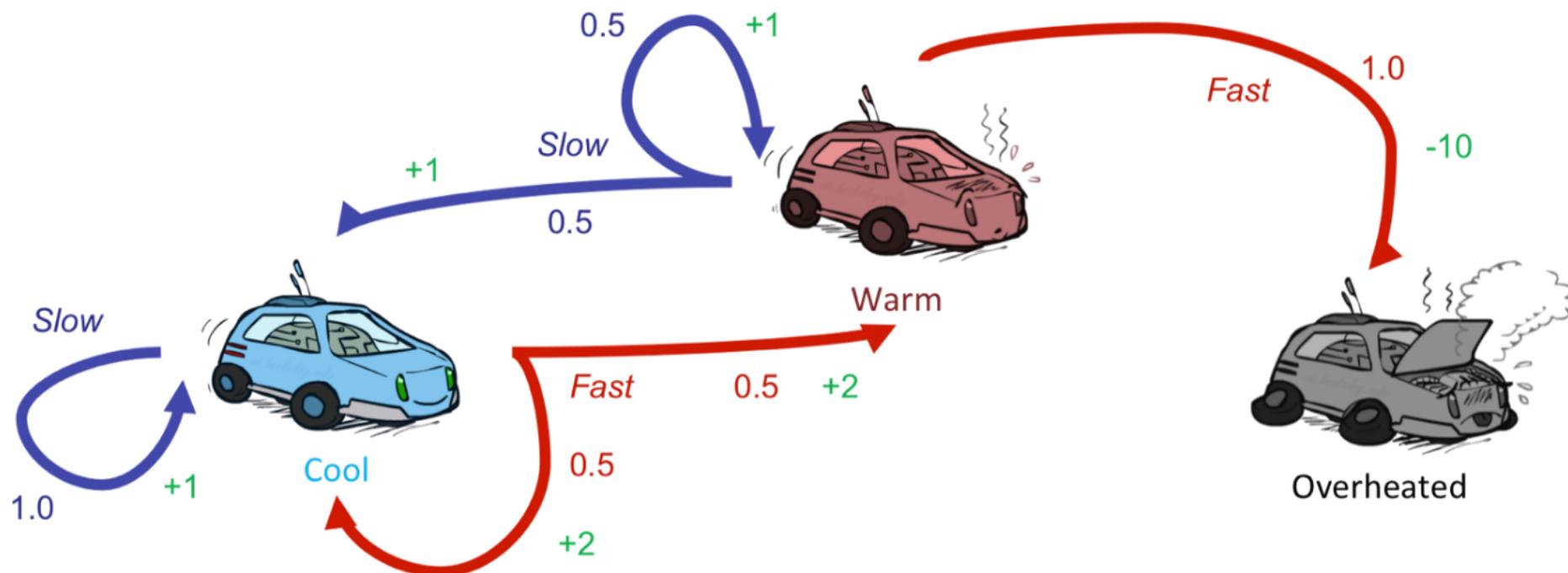
$$\pi_h(\text{Slow} \mid \text{Cool})$$

$$\pi_h(\text{Fast} \mid \text{Cool})$$

$$\pi_h(\text{Slow} \mid \text{Warm})$$

$$\pi_h(\text{Fast} \mid \text{Warm})$$

$$\forall h$$

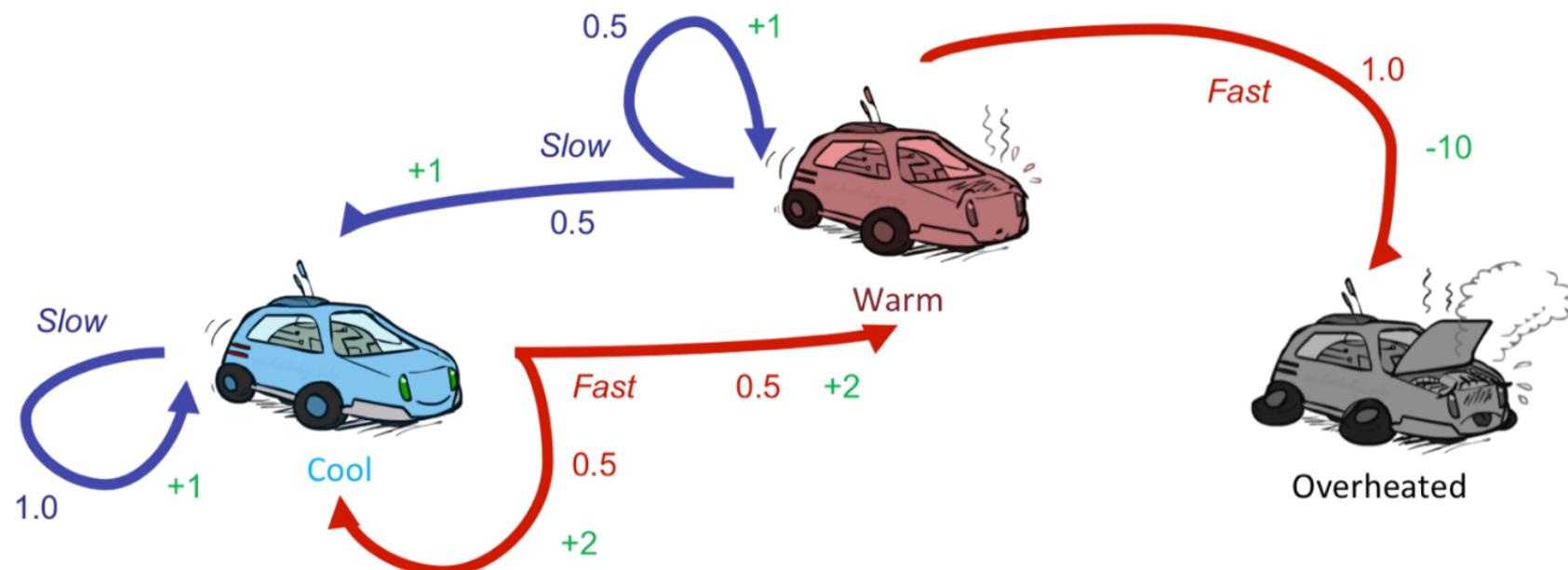


# **Value Iteration**

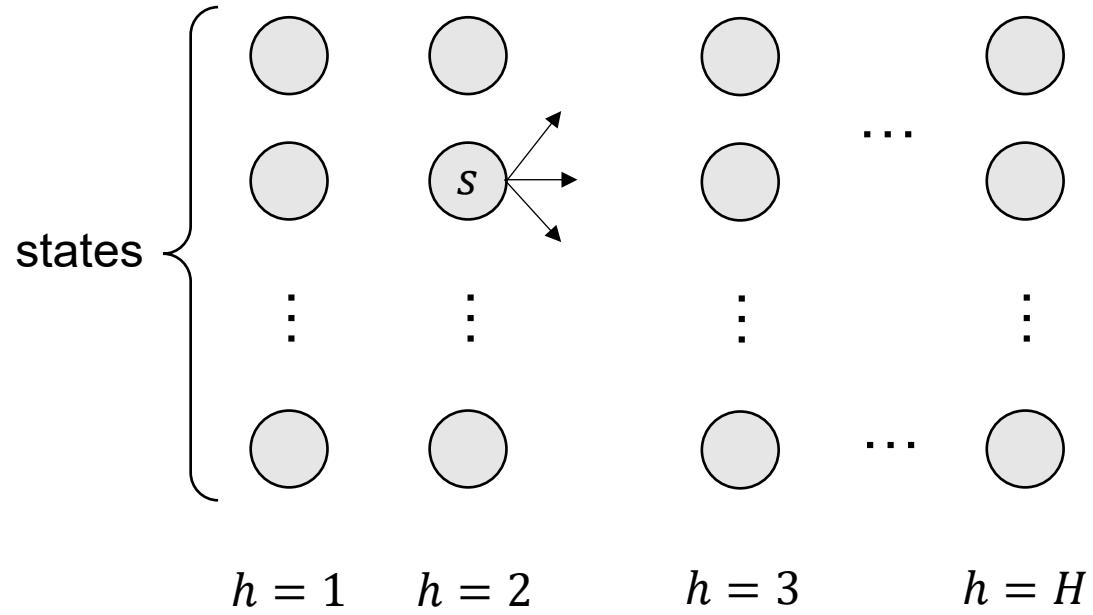
(Fixed-Horizon + Total-Reward)

# Two Tasks

- Policy Evaluation:** Calculate the expected total reward of a given policy  
What is the expected total reward for the policy  $\pi(\text{cool}) = \text{fast}$ ,  $\pi(\text{warm}) = \text{slow}$ ?
- Policy Optimization:** Find the best policy  
What is the policy that achieves the highest expected total reward?



# Value Iteration for Policy Evaluation



State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$Q_h^\pi(s, a) = \mathbb{E}^{\textcolor{red}{\pi}} \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^\pi(s) = \mathbb{E}^{\pi} \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

**Backward induction:**

$$V_{H+1}^\pi(s) = 0 \quad \forall s$$

For  $h = H, \dots, 1$ : for all  $s, a$

$$Q_h^\pi(s, a) = R(s, a) + \underbrace{\sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')}_{\text{Expected total reward of } \pi \text{ from step } h+1}$$

$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

# Bellman Equation

$Q_h^\pi$  is called “the state-action value functions of policy  $\pi$ ”

$V_h^\pi$  is called “the state value function of policy  $\pi$ ”

Both can be just called “**value functions**”

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')$$

$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

or

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s', a'} P(s'|s, a) \pi_{h+1}(a'|s') Q_{h+1}^\pi(s', a')$$

or

$$V_h^\pi(s) = \sum_a \pi_h(a|s) \left( R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s') \right)$$

# The Meaning of Bellman Equations

## Definitions

$$Q_h^\pi(s, a) \triangleq \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^\pi(s) \triangleq \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

## Relations (Bellman Equations)

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')$$

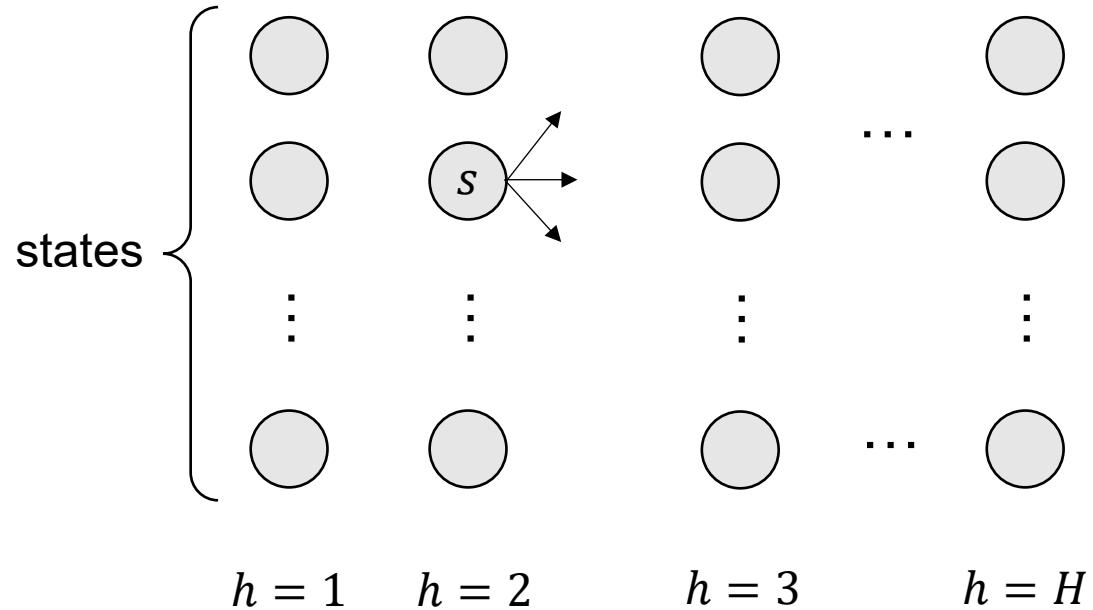
$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

## Calculation (VI)

Calculate

$Q_h^\pi(s, a), V_h^\pi(s) \forall s, a$   
from  $h = H$  to  $h = 1$

# Value Iteration for Policy Optimization



State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$Q_h^*(s, a) = \max_{\pi \in \Pi_M} \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^*(s) = \max_{\pi \in \Pi_M} \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

**Backward induction:**

$$V_{H+1}^*(s) = 0 \quad \forall s$$

For  $h = H, \dots, 1$ : for all  $s, a$

$$Q_h^*(s, a) = R(s, a) + \underbrace{\sum_{s'} P(s'|s, a) V_{h+1}^*(s')}_{\text{Expected optimal total reward from step } h+1}$$

$$V_h^*(s) = \max_a Q_h^*(s, a) \quad \pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$$

# Exercise

$s$	$a$	$s'$	$P(s' s, a)$	$R(s, a)$
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	-10
	(end)		1.0	0

$$Q_3^*(s, a)$$

$$Q_3^*(\text{cool}, \text{slow})$$

$$Q_3^*(\text{cool}, \text{fast})$$

$$Q_3^*(\text{warm}, \text{slow})$$

$$Q_3^*(\text{warm}, \text{fast})$$


---

$$V_3^*(s)$$

$$V_3^*(\text{cool})$$

$$V_3^*(\text{warm})$$


---

$$Q_2^*(s, a)$$

$$Q_2^*(\text{cool}, \text{slow})$$

$$Q_2^*(\text{cool}, \text{fast})$$

$$Q_2^*(\text{warm}, \text{slow})$$

$$Q_2^*(\text{warm}, \text{fast})$$


---

$$V_2^*(s)$$

$$V_2^*(\text{cool})$$

$$V_2^*(\text{warm})$$

Assume  $H = 3$

# Bellman Optimality Equation

$Q_h^*$  : optimal state-action value functions  
 $V_h^*$  : optimal state value functions  
or “optimal value functions”

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s')$$

$$V_h^*(s) = \max_a Q_h^*(s, a)$$

or

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) \left( \max_{a'} Q_{h+1}^*(s', a') \right)$$

or

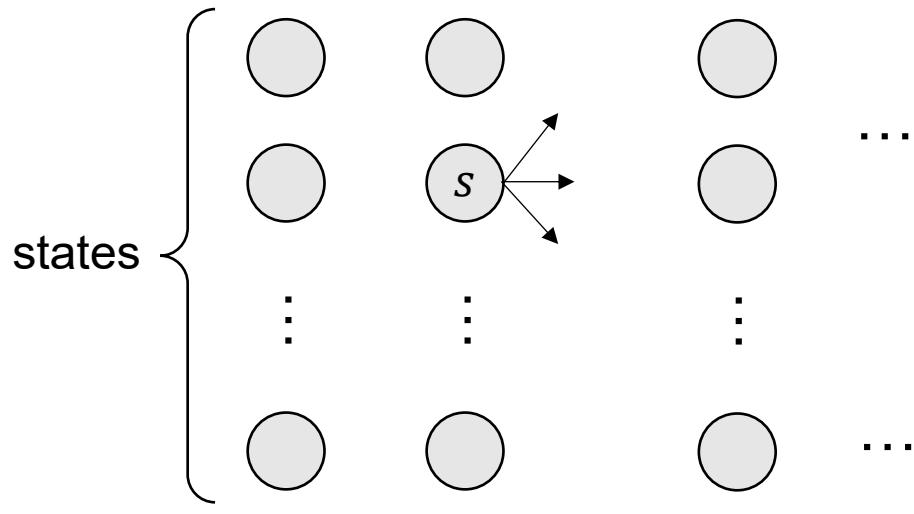
$$V_h^*(s) = \max_a \left( R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s') \right)$$

$$\pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$$

# **Value Iteration**

(Variable-Horizon + Discounted Reward)

# Value Iteration for Policy Evaluation



$h = 1$	$h = 2$	$h = 3$
weight	1	$\gamma$
		$\gamma^2$

State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$Q_i^\pi(s, a) = \mathbb{E}^\pi \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid (s_1, a_1) = (s, a) \right]$$

$$V_i^\pi(s) = \mathbb{E}^\pi \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid s_1 = s \right]$$

$$Q^\pi(s, a) = Q_\infty^\pi(s, a) \quad V^\pi(s) = V_\infty^\pi(s)$$

$$V_0^\pi(s) = 0 \quad \forall s$$

For  $i = 1, 2, 3, \dots$ : for all  $s, a$

$$Q_i^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_{i-1}^\pi(s')$$

$$V_i^\pi(s) = \sum_a \pi(a|s) Q_i^\pi(s, a)$$

If  $|Q_i^\pi(s, a) - Q_{i-1}^\pi(s, a)| \leq \epsilon$  for all  $s, a$ : **terminate**

# Bellman Equation

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')$$

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s, a)$$

or

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi(a'|s') Q^\pi(s', a')$$

or

$$V^\pi(s) = \sum_a \pi(a|s) \left( R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s') \right)$$

# The Meaning of Bellman Equations

## Definitions

$$Q^\pi(s, a) = \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid (s_1, a_1) = (s, a) \right]$$

$$V^\pi(s) = \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_1 = s \right]$$

## Relations (Bellman Equations)

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')$$

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s, a)$$

## Calculation (VI)

Calculate  
 $Q_i^\pi(s, a), V_i^\pi(s) \forall s, a$   
for  $i = 1, 2, \dots$   
until terminated

# What's the Quality of $Q_i^\pi(s, a)$ when VI Terminates?

1. Value Iteration for policy evaluation will terminate.
2. When it terminates, it holds that

$$|Q_i^\pi(s, a) - Q^\pi(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

# Convergence

1. Value Iteration for policy evaluation will terminate.
2. When it terminates, it holds that

$$|Q_i^\pi(s, a) - Q^\pi(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

Proof strategy:

- 1) Prove that VI will terminate (i.e.,  $\max_{s,a} |Q_i^\pi(s, a) - Q_{i-1}^\pi(s, a)| \leq \epsilon$  will eventually hold)
- 2) At termination,

$$\text{BellmanError}(Q_i^\pi) = \max_{s,a} \left| Q_i^\pi(s, a) - \left( R(s, a) + \gamma \sum_{s',a'} P(s'|s, a) \pi(a'|s') Q_i^\pi(s', a') \right) \right| \leq \epsilon$$

- 3) Use the **Value error  $\leq (1 - \gamma)^{-1}$  Bellmen Error** lemma to claim

$$|Q_i^\pi(s, a) - Q^\pi(s, a)| \leq \frac{\epsilon}{1 - \gamma}.$$

# Convergence (A More General Statement of 2.)

**Value error  $\leq (1 - \gamma)^{-1}$  Bellmen Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function (not necessarily generated by Value Iteration)

If

$$\left| f(s, a) - \left( R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi(a'|s') f(s', a') \right) \right| \leq \epsilon \quad \forall s, a$$

then

$$|f(s, a) - Q^\pi(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

Given  $\pi$ , Assume we have

$$\underline{f(s,a)} \stackrel{\geq}{\leq} R(s,a) + \gamma \sum_{s',a'} p(s'|s,a) \pi(a'|s') f(s',a') - \varepsilon$$

$\forall s,a$

$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{s',a'} p(s'|s,a) \pi(a'|s') Q^\pi(s',a')$$

$\forall s,a$

$$\underline{f(s,a) - Q^\pi(s,a)} \leq \gamma \sum_{s',a'} p(s'|s,a) \pi(a'|s') (f(s',a') - Q^\pi(s',a')) + \varepsilon$$

$\forall s,a$

$$\leq \gamma \max_{s',a'} (f(s',a') - Q^\pi(s',a')) + \varepsilon$$

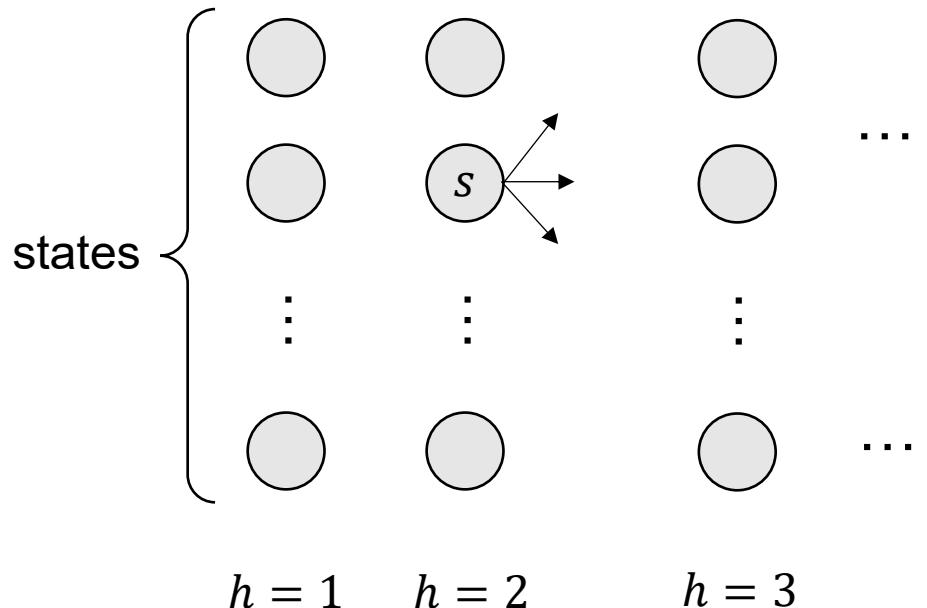
$$\Rightarrow \max_{s,a} (f(s,a) - Q^\pi(s,a)) \leq \gamma \max_{s',a'} (f(s',a') - Q^\pi(s',a')) + \varepsilon$$

Similarly:

$$\Rightarrow (\cancel{\max_{s,a}}) \max_{s,a} (f(s,a) - Q^\pi(s,a)) \leq \frac{\varepsilon}{1-\gamma}$$

$$\min_{s,a} (f(s,a) - Q^\pi(s,a)) \geq -\frac{\varepsilon}{1-\gamma}$$

# Value Iteration for Policy Optimization



State transition:  $P(s'|s, a)$

Reward:  $R(s, a)$

$$Q_i^*(s, a) = \max_{\pi} \mathbb{E}^{\pi} \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid (s_0, a_0) = (s, a) \right]$$

$$V_i^*(s) = \max_{\pi} \mathbb{E}^{\pi} \left[ \sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid s_0 = s \right]$$

$$Q^*(s, a) = Q_{\infty}^*(s, a) \quad V^*(s) = V_{\infty}^*(s)$$

$$V_0^*(s) = 0 \quad \forall s$$

For  $i = 1, 2, 3, \dots$ : for all  $s, a$

$$Q_i^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_{i-1}^*(s')$$

$$V_i^*(s) = \max_a Q_i^*(s, a)$$

If  $|Q_i^*(s, a) - Q_{i-1}^*(s, a)| \leq \epsilon$  for all  $s, a$ : **terminate**

# Bellman Optimality Equation

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

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s')$$

$$V^*(s) = \max_a Q^*(s, a)$$

or

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a')$$

or

$$V^*(s) = \max_a \left( R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s') \right)$$

# Convergence

1. Value Iteration for policy optimization will terminate.
2. When it terminates, it holds that

$$|Q_i^*(s, a) - Q^*(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

3. When it terminates, it holds that

$$V^*(s) - V^{\hat{\pi}}(s) \leq \frac{2\epsilon}{(1 - \gamma)^2} \quad \forall s$$

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

# Convergence (A More General Statement of 2.)

**Value error  $\leq (1 - \gamma)^{-1}$  Bellmen Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function (not necessarily generated by Value Iteration)

If

$$\left| f(s, a) - \left( R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} f(s', a') \right) \right| \leq \epsilon \quad \forall s, a$$

then

$$|f(s, a) - Q^*(s, a)| \leq \frac{\epsilon}{1 - \gamma} \quad \forall s, a$$

# Convergence (A More General Statement of 3.)

**Suboptimality  $\leq (1 - \gamma)^{-1}$  Value Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function (not necessarily generated by Value Iteration)

If

$$|f(s, a) - Q^*(s, a)| \leq \epsilon \quad \forall s, a$$

then

$$V^*(s) - V^{\pi_f}(s) \leq \frac{2\epsilon}{1 - \gamma} \quad \forall s$$

where  $\pi_f(s) = \operatorname{argmax}_a f(s, a)$

Review:



pure exploration

$$\hat{R}(a)$$

estimated value function

pure exploitation

$$a_t = \underset{a}{\operatorname{argmax}} \hat{R}(a)$$

$$\hat{a}^* = \underset{a}{\operatorname{argmax}} \hat{R}(a)$$

$$a^* = \underset{a}{\operatorname{argmax}} R(a)$$

$$\forall a \quad |R(a) - \hat{R}(a)| \leq \varepsilon$$

$$R(\hat{a}) - R(\hat{a}) = \underbrace{\hat{R}(a^*) - \hat{R}(\hat{a})}_{\leq 0} + \underbrace{R(a^*) - \hat{R}(a^*)}_{\leq \varepsilon} + \underbrace{\hat{R}(\hat{a}) - R(\hat{a})}_{\leq \varepsilon}$$

$$\leq 0$$

$$\leq \varepsilon$$

$$\leq \varepsilon$$

$$\leq 2\varepsilon$$

# Summary (Fixed Horizon)

## Definitions

$$Q_h^\pi(s, a) \triangleq \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^\pi(s) \triangleq \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

## Relations (Bellman Equations)

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')$$

$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

## Calculation (VI)

Calculate  
 $Q_h^\pi(s, a), V_h^\pi(s) \forall s, a$   
from  $h = H$  to  $h = 1$

$$Q_h^*(s, a) \triangleq \max_\pi \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^*(s) \triangleq \max_\pi \mathbb{E}^\pi \left[ \sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s')$$

$$V_h^*(s) = \max_a Q_h^*(s, a)$$

Calculate  
 $Q_h^*(s, a), V_h^*(s) \forall s, a$   
from  $h = H$  to  $h = 1$

# Summary (Discounted Variable Horizon)

## Definitions

$$Q^\pi(s, a) = \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid (s_1, a_1) = (s, a) \right]$$

$$V^\pi(s) = \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_1 = s \right]$$

## Relations (Bellman Equations)

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')$$

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s, a)$$

## Calculation (VI)

Calculate  
 $Q_i^\pi(s, a), V_i^\pi(s) \forall s, a$   
for  $i = 1, 2, \dots$   
until convergence

$$Q^*(s, a) = \max_{\pi} \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid (s_1, a_1) = (s, a) \right]$$

$$V^*(s) = \max_{\pi} \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_1 = s \right]$$

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s')$$

$$V^*(s) = \max_a Q^*(s, a)$$

Calculate  
 $Q_i^*(s, a), V_i^*(s) \forall s, a$   
for  $i = 1, 2, \dots$   
until convergence

# **Policy Iteration**

---

*Policy Optimization*

# Policy Iteration

$$\pi_i : S \rightarrow A$$

## Policy Iteration

For  $i = 1, 2, \dots$

$$\forall s, \quad \pi_i(s) \leftarrow \underset{a}{\operatorname{argmax}} Q^{\pi_{i-1}}(s, a)$$

$\pi_i(s) \neq \underset{a}{\operatorname{argmax}} Q^{\pi_i}(s, a)$

Requires an inner  
VI for policy evaluation algo

**Theorem (monotonic improvement).** Policy Iteration ensures

$$\forall s, a, \quad Q^{\pi_i}(s, a) \geq Q^{\pi_{i-1}}(s, a)$$

When converged (i.e.,  $\pi_i = \pi_{i-1}$ ), we have  $\pi_i = \pi^*$ .

(We will prove this later.)

# Generalized Policy Iteration

$N = \infty \Rightarrow$  Policy Iteration

(Sub-routine:  
VI for policy evaluation)

$N = 1 \Rightarrow$  Value Iteration for policy optimization

For  $i = 1, 2, \dots$

$$\pi_i(s) = \max_a Q_i(s, a)$$

$$Q_i(s, a) = \underbrace{Q}_{\text{Policy update}}^{\pi_{i-1}}(s, a) \quad (\text{inductive prove this})$$

$$Q \leftarrow Q_i$$

Repeat for  $N$  times:

$$Q_{i+1}(s, a) \leftarrow R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi_i(a'|s') Q_i(s', a')$$

$$Q_{i+1} \leftarrow Q$$

perform Value iteration to evaluate  $\pi_i$

**Value update**

$(s, a) \rightarrow R$

**Notice:** in value iteration for PO, there may not exist a policy  $\pi$  such that  $Q_i = Q^\pi$

In contrast, in policy iteration we have  $Q_i = Q^{\pi_{i-1}}$

VI for PO can be viewed as PI with **incomplete policy evaluation**

$$Q_i = Q^{\pi_{i-1}}$$

# Summary

- Value Iteration for Policy Optimization (VI for PO)
  - Is essentially a **dynamic programming** algorithm
  - Finds the value functions of the optimal policy
- Value Iteration for Policy Evaluation (VI for PE)
  - Also a **dynamic programming** algorithm
  - Finds the value functions of the given policy
- Policy Iteration (PI)
  - An iterative policy improvement algorithm (for PO)
  - Each iteration involves a policy evaluation subtask
- VI for PO and PI can be viewed as special cases of Generalized PI



# **Performance Difference Lemma**

# Recall: Regret

$$\text{Regret} = \max_{\pi^*} \mathbb{E}^{\pi^*} \left[ \sum_{t=1}^T \sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^*(\tilde{s}_{t,h})) \right] - \sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$

$$\mathbb{E}[\text{Regret}] = \mathbb{E} \left[ \sum_{t=1}^T (V_1^*(s_{t,1}) - V_1^{\pi_t}(s_{t,1})) \right]$$

$$= \mathbb{E} \left[ \sum_{t=1}^T (V_1^*(\rho) - V_1^{\pi_t}(\rho)) \right] \quad V_1^\pi(\rho) \triangleq \mathbb{E}_{s \sim \rho}[V_1^\pi(s)]$$

# Unanswered Questions

- For an estimation  $\hat{Q}(s, a) \approx Q^*(s, a)$  with error, how can we bound

$$V^*(\rho) - V^{\hat{\pi}}(\rho) \quad \text{where } \hat{\pi}(s) = \operatorname{argmax}_a \hat{Q}(s, a)?$$

- How to show that Policy Iteration leads to monotonic policy improvement?
- Also, how are these methods related to the third challenge of online RL: credit assignment?

## Performance Difference Lemma

$$\sum_{s,a} d_\rho^{\pi'}(s) \pi'(a|s) Q^{\pi}(s,a) = \sum_{s,a} d_\rho^{\pi'}(s,a) Q^{\pi}(s,a)$$

$$\sum_{s,a} d_\rho^{\pi'}(s) \pi(a|s) Q^{\pi}(s,a) = \sum_s d_\rho^{\pi'}(s) V^{\pi}(s)$$

For any two stationary policies  $\pi'$  and  $\pi$  in the discounted setting,

$$\mathbb{E}_{s \sim \rho} [V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho} [V^{\pi}(s)] = \sum_{s,a} d_\rho^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^\pi(s,a)$$

$$\sum_{k=1}^{\infty} \gamma^{k-1} = \frac{1}{1-\gamma}$$

$$= \sum_{s,a} d_\rho^{\pi'}(s,a) (Q^\pi(s,a) - V^\pi(s))$$

$$d_\rho^\pi(s) \triangleq \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} \mathbb{I}\{s_h = s\} \mid s_1 \sim \rho \right]$$

$$d_\rho^\pi(s,a) \triangleq \mathbb{E}^\pi \left[ \sum_{h=1}^{\infty} \gamma^{h-1} \mathbb{I}\{(s_h, a_h) = (s, a)\} \mid s_1 \sim \rho \right]$$

Discounted occupancy measure on state  $s$

$$\sum_a d_\rho^{\pi'}(s,a) = d_\rho^{\pi'}(s)$$

$$= d_\rho^{\pi}(s) \pi(a|s)$$

# Performance Difference Lemma

We can also swap the roles of  $\pi'$  and  $\pi$  and apply the same lemma

$$\mathbb{E}_{s \sim \rho}[V^\pi(s)] - \mathbb{E}_{s \sim \rho}[V^{\pi'}(s)] = \sum_{s,a} d_\rho^\pi(s) (\pi(a|s) - \pi'(a|s)) Q^{\pi'}(s, a)$$

$$\times (-1) \Rightarrow \mathbb{E}_{s \sim \rho}[V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho}[V^\pi(s)] = \sum_{s,a} d_\rho^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^\pi(s, a)$$

||

Original version:

$$\mathbb{E}_{s \sim \rho}[V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho}[V^\pi(s)] = \sum_{s,a} d_\rho^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^\pi(s, a)$$

# Performance Difference Lemma (Fixed-Horizon)

For any two Markov policies  $\pi'$  and  $\pi$  in the fixed-horizon setting,

$$\begin{aligned}\mathbb{E}_{s_1 \sim \rho} [V_1^{\pi'}(s_1)] - \mathbb{E}_{s_1 \sim \rho} [V_1^{\pi}(s_1)] &= \sum_{h=1}^H \sum_{s,a} d_{\rho,h}^{\pi'}(s) (\pi'_h(a|s) - \pi_h(a|s)) \underline{Q_h^{\pi}(s, a)} \\ &= \sum_{h=1}^H \sum_{s,a} d_{\rho,h}^{\pi'}(s, a) (Q_h^{\pi}(s, a) - V_h^{\pi}(s))\end{aligned}$$

$$\underline{d_{\rho,h}^{\pi}(s)} \triangleq \mathbb{E}^{\pi} [\mathbb{I}\{s_h = s\} \mid s_1 \sim \rho] = \mathbb{P}^{\pi}(s_h = s \mid s_1 \sim \rho)$$

$$d_{\rho,h}^{\pi}(s, a) \triangleq \mathbb{E}^{\pi} [\mathbb{I}\{(s_h, a_h) = (s, a)\} \mid s_1 \sim \rho] = \mathbb{P}^{\pi}((s_h, a_h) = (s, a) \mid s_1 \sim \rho)$$

# The Meaning of Performance Difference Lemma

It tells us how **credit** are assigned to each state/step

The sub-optimality of a policy  $\pi$ :

$$\mathbb{E}_{s \sim \rho}[V^*(s)] - \mathbb{E}_{s \sim \rho}[V^\pi(s)]$$

If  $\pi$  is highly sub-optimal, then we can always find

- 1) An  $(s, a)$ -pair on the path of  $\pi$  such that  $V^*(s) - Q^*(s, a)$  is positive and large
- 2) An  $(s, a)$ -pair on the path of  $\pi^*$  such that  $Q^\pi(s, a) - V^\pi(s)$  is positive and large

If  $\pi$  puts a lot of probability  $\pi(a|s)$  on an action with large  $V^*(s) - Q^*(s, a)$

$$= \sum_{s,a} d_\rho^\pi(s) (\pi^*(a|s) - \pi(a|s)) Q^*(s, a)$$

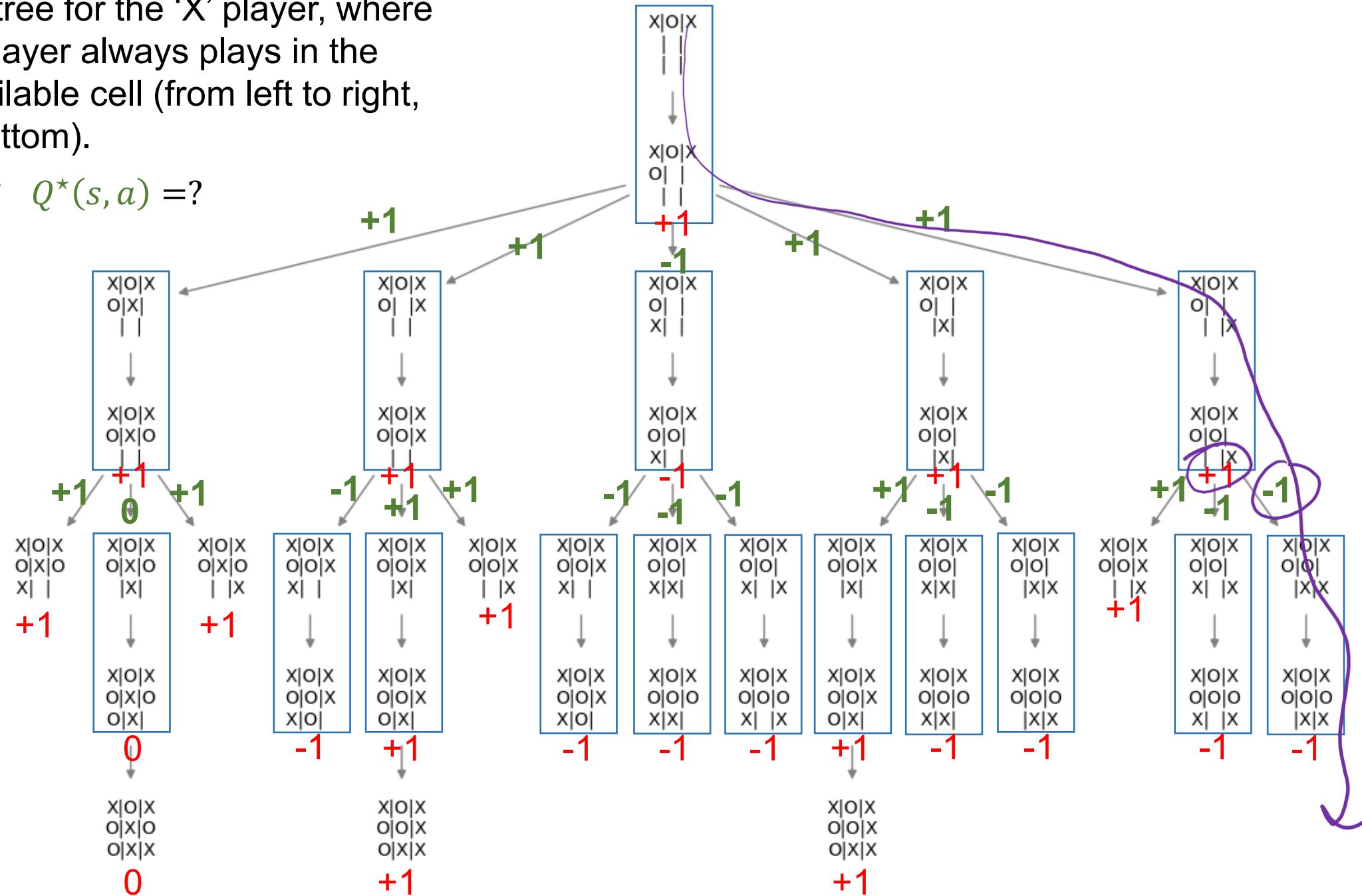
$$= \sum_{s,a} d_\rho^\pi(s, a) (V^*(s) - Q^*(s, a)) \geq 0 \quad \checkmark$$

$$= \sum_{s,a} d_\rho^{\pi^*}(s) (\pi^*(a|s) - \pi(a|s)) Q^\pi(s, a)$$

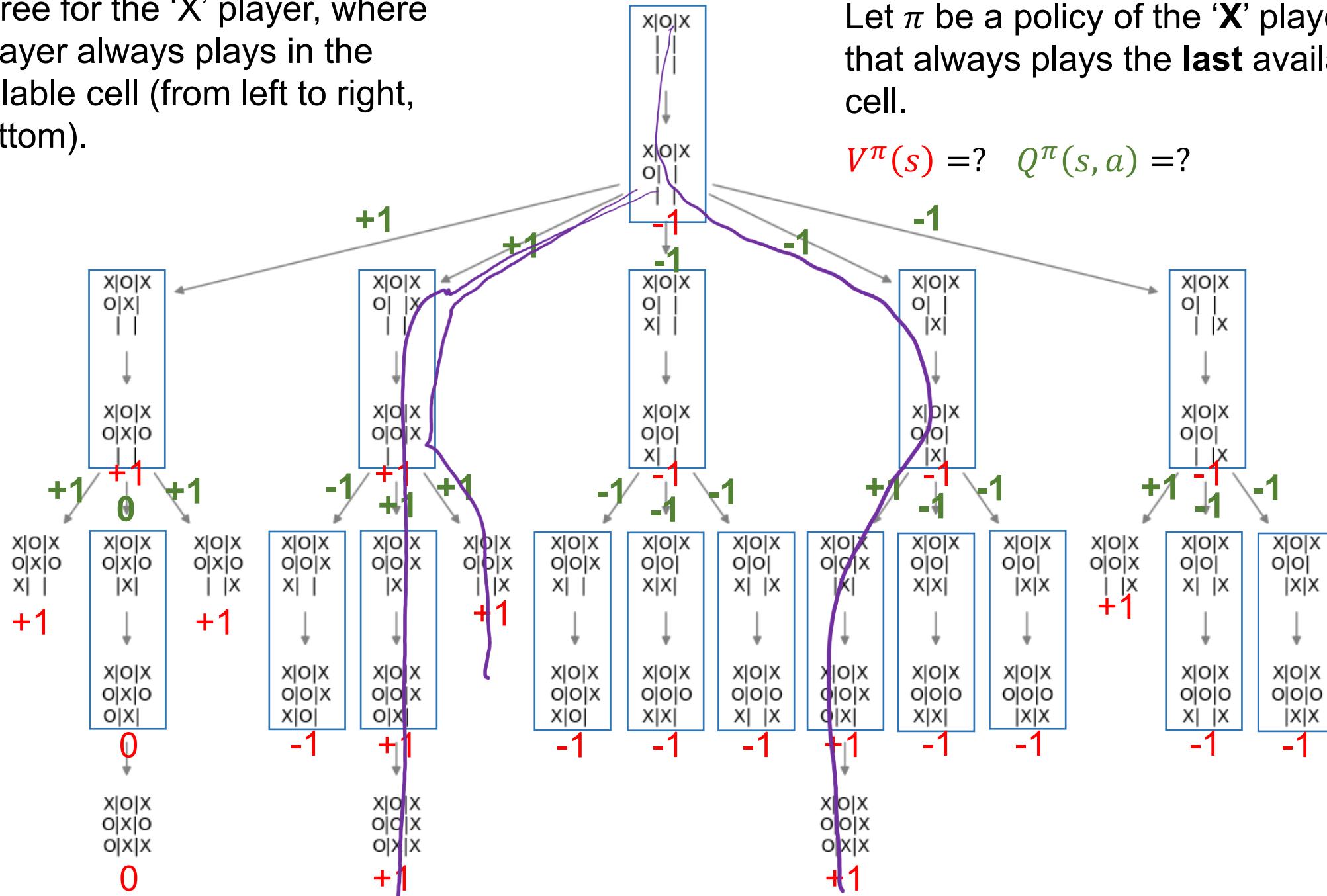
$$= \sum_{s,a} d_\rho^{\pi^*}(s, a) (Q^\pi(s, a) - V^\pi(s)) \quad \checkmark$$

A game tree for the 'X' player, where the 'O' player always plays in the **first** available cell (from left to right, top to bottom).

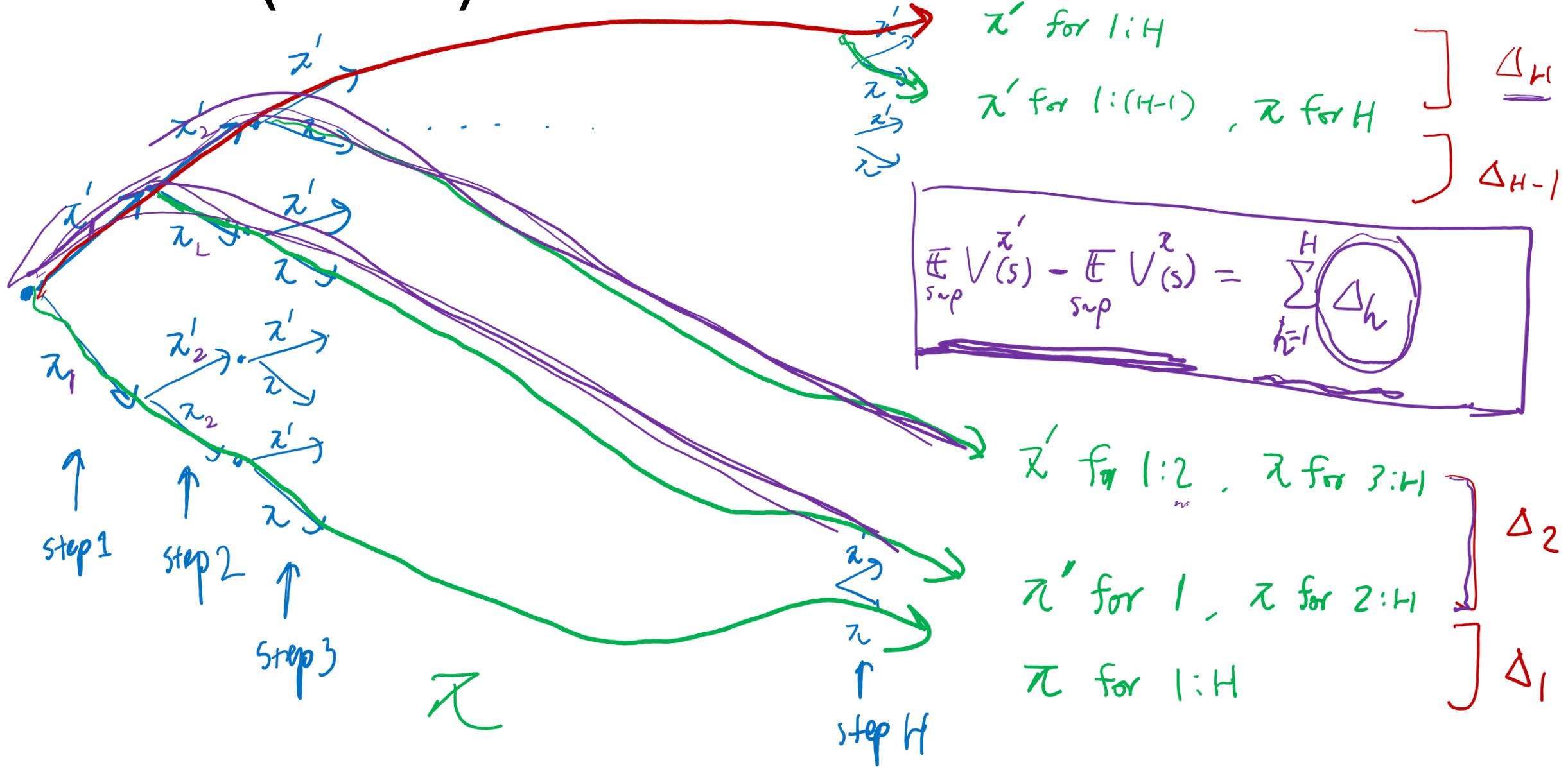
$$V^*(s) = ? \quad Q^*(s, a) = ?$$



A game tree for the 'X' player, where the 'O' player always plays in the **first** available cell (from left to right, top to bottom).



# Proof (Sketch) of Performance Difference Lemma



Now focus on calculating  $\Delta_h$

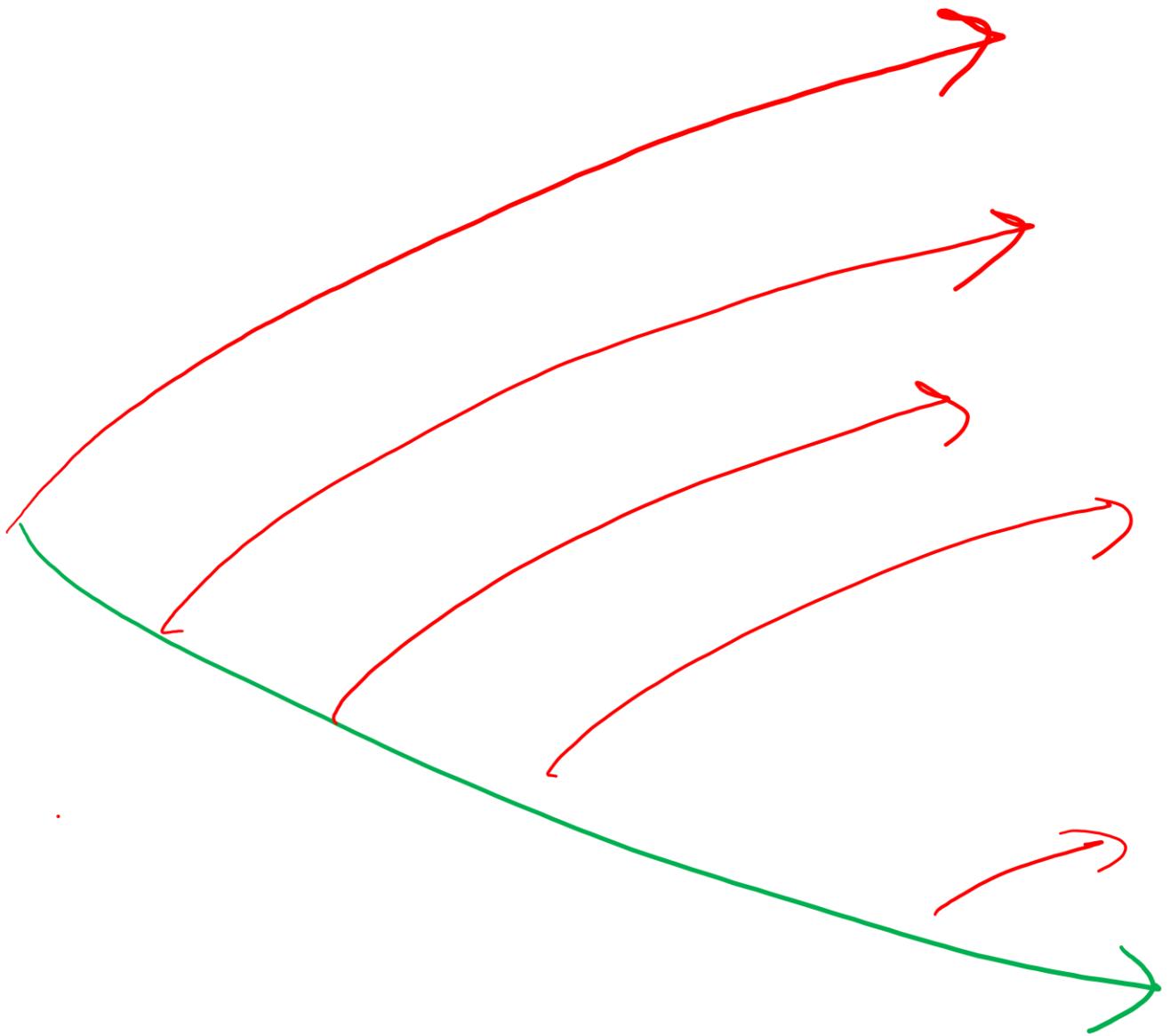
$Q^\pi(s_h, a_h) = \text{return by picking } a_h \text{ on } s_h, \text{ then follow } \pi \text{ afterwards}$

$$\begin{aligned}\pi^1 &= \pi' \text{ for step } 1:h, \pi \text{ for } (h+1):H \\ \underline{\pi^2} &= \pi' \text{ for step } 1:(h-1), \pi \text{ for } h:H\end{aligned}\quad \text{only differ in step } h$$

$$\mathbb{E}^{\pi^1} \left[ \sum_{k=1}^H r(s_k, a_k) \mid s_1 \sim p \right] - \mathbb{E}^{\pi^2} \left[ \sum_{k=1}^H r(s_k, a_k) \mid s_1 \sim p \right]$$

$$= \mathbb{E}^{\pi^1} \left[ \sum_{k=h}^H r(s_k, a_k) \mid \underbrace{s_1 \sim p}_{s_h \sim d_{p,h}} \right] - \mathbb{E}^{\pi^2} \left[ \sum_{k=h}^H r(s_k, a_k) \mid \underbrace{s_1 \sim p}_{s_h \sim d_{p,h}} \right]$$

$$\begin{aligned}&= \mathbb{E}_{\substack{s_h \sim d_{p,h} \\ a_h}} \left[ \sum_{a_h} \pi'(a_h \mid s_h) Q^\pi(s_h, a_h) \right] - \mathbb{E}_{\substack{s_h \sim d_{p,h} \\ a_h}} \left[ \sum_{a_h} \pi(a_h \mid s_h) Q_h^\pi(s_h, a_h) \right] \\ &\approx \sum_{s_h, a_h} d_{p,h}^{\pi'}(s_h) (\pi'(a_h \mid s_h) - \pi(a_h \mid s_h)) Q^\pi(s_h, a_h)\end{aligned}$$



# Unanswered Question 1

**Suboptimality  $\leq (1 - \gamma)^{-1}$  Value Error**

Let  $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  be **any** function

If

$$|f(s, a) - Q^*(s, a)| \leq \epsilon \quad \forall s, a$$

then

$$\underbrace{V^*(s) - V^{\pi_f}(s)}_{\text{Suboptimality}} \leq \frac{2\epsilon}{1 - \gamma} \quad \forall s$$

where  $\pi_f(s) = \operatorname{argmax}_a f(s, a)$

$$|f(s,a) - Q^*(s,a)| \leq \varepsilon$$

$$\sum_a (\pi^*(a|s) - \pi_f(a|s)) f(s,a) = \sum_a \pi^*(a|s) f(s,a) - \max_a f(s,a) \leq 0$$

$\pi_f(s) = \underset{a}{\operatorname{argmax}} f(s,a)$

$$\begin{aligned} \mathbb{E}_{s \sim p} [V^{\pi^*}(s)] - \mathbb{E}_{s \sim p} [V^{\pi_f}(s)] &= \sum_{s,a} d_p^{\pi_f}(s) \left( \pi^*(a|s) - \pi_f(a|s) \right) \overbrace{Q^*(s,a)} \\ &= \sum_{s,a} d_p^{\pi_f}(s) \left( \pi^*(a|s) - \pi_f(a|s) \right) \overbrace{f(s,a)} \leq 0 \\ &\quad + \sum_{s,a} d_p^{\pi_f}(s) \left( \pi^*(a|s) - \pi_f(a|s) \right) \overbrace{(Q^*(s,a) - f(s,a))} \\ &\leq \sum_{s,a} d_p^{\pi_f}(s) \left| \pi^*(a|s) - \pi_f(a|s) \right| \varepsilon \\ &\leq \left( \sum_s d_p^{\pi_f}(s) \right) 2\varepsilon = \frac{2\varepsilon}{1-\gamma} \end{aligned}$$

## Unanswered Question 2

$$\pi_i(s) = \arg\max_a Q^{\pi_{i-1}}(s, a)$$

Policy Iteration ensures

$$\forall s, a, \quad Q^{\pi_i}(s, a) \geq Q^{\pi_{i-1}}(s, a)$$

$Q^{\pi_i}(s, a) = R(s, a) + \gamma \sum_{s'} p(s'|s, a) V^{\pi_i}(s')$   
 $Q^{\pi_{i-1}}(s, a) = \underbrace{\sum_s}_{V^{\pi_{i-1}}(s)}$

When converged (i.e.,  $\pi_i = \pi_{i-1}$ ), we have  $\pi_i = \pi^*$ .

$$\begin{aligned}
 \mathbb{E}_{s \sim p} \left[ V^{\pi_i}(s) \right] - \mathbb{E}_{s \sim p} \left[ V^{\pi_{i-1}}(s) \right] &= \sum_{s, a} d_p^{\pi_i}(s) \left( \underline{\pi_i(a|s)} - \underline{\pi_{i-1}(a|s)} \right) \underline{Q^{\pi_{i-1}}(s, a)} \\
 &= \sum_s d_p^{\pi_i}(s) \left( \underbrace{\max_a Q^{\pi_{i-1}}(s, a)}_{\geq 0} - \sum_a \pi_{i-1}(a|s) \underline{Q^{\pi_{i-1}}(s, a)} \right)
 \end{aligned}$$

$$\forall s, \quad V^{\pi_i}(s) \geq V^{\pi_{i-1}}(s) \quad \geq 0$$

If  $\pi_i = \pi_{i-1} = \hat{\pi} \Rightarrow Q^{\hat{\pi}}$  satisfies Bellman Optimality Equation  $\Rightarrow \text{Bellman Error}(Q^{\hat{\pi}}) = 0 \Rightarrow Q^{\hat{\pi}} = Q^*$   
 Value error  $\leq \frac{1}{1-\gamma}$  Bellman error

$$\pi_i = \pi_{i-1}$$

$$\Rightarrow \pi_i(s) = \operatorname{argmax}_a Q^{\pi_i}(s, a)$$

$$\Rightarrow Q^{\pi_i}(s, a) = R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi_i(a'|s') Q^{\pi_i}(s', a') = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^{\pi_i}(s', a')$$

$\Rightarrow Q^{\pi_i}$  satisfies the Bellman optimality equation

$$\Rightarrow \text{BellmanError}(Q^{\pi_i}) = 0$$

$$\Rightarrow Q^{\pi_i}(s, a) = Q^\star(s, a) \text{ by the "ValueError} \leq \frac{1}{1-\gamma} \text{ BellmanError" lemma on Page 38}$$

$$\Rightarrow \pi_i(s) = \operatorname{argmax}_a Q^\star(s, a) = \pi^\star(s).$$

# Recap: MDP

- Definitions of  $Q^\pi(s, a), V^\pi(s), Q^*(s, a), V^*(s)$
- Bellman equations (related to dynamic programming)
- Value Iteration to approximate  $Q^\pi(s, a)/V^\pi(s)$  or  $Q^*(s, a)/V^*(s)$
- Policy Iteration to find  $\pi^*$  --- involving  $Q^\pi(s, a)/V^\pi(s)$  approximation
- Unified by Generalized Policy Iteration
- Performance difference lemma to decompose  $\mathbb{E}_{s \sim \rho} [V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho} [V^\pi(s)]$ 
  - Credit assignment
  - $= \sum_{s,a} d_\rho^\pi(s, a) (V^{\pi'}(s) - Q^{\pi'}(s, a))$  (helpful in analyzing VI by letting  $\pi' = \pi^*$ )
  - $= \sum_{s,a} d_\rho^{\pi'}(s, a) (Q^\pi(s, a) - V^\pi(s))$  (helpful in analyzing PI by letting  $\pi' = \pi_{i+1}$ )

# Next

- Our discussion indicates there are two potential ways to find optimal policy
  - Value-Iteration-based: approximate  $\hat{Q}(s, a) \approx Q^*(s, a)$  and let  $\hat{\pi}(s) = \operatorname{argmax}_a \hat{Q}(s, a)$
  - Policy-Iteration-based: approximate  $\hat{Q}(s, a) \approx Q^\pi(s, a)$  and let  $\hat{\pi}(s) = \operatorname{argmax}_a \hat{Q}(s, a)$
  - ... or something in between (based on generalized policy iteration)
- RL algorithms we will discuss:
  - Performing approximate VI or PI using samples