



C2

Markov Decision Processes

COMP3270

Introduction to Artificial Intelligence

Chapter 2: Roadmap

Chapter 2: Roadmap

Model Uncertainty

States, actions, transitions, and rewards

Chapter 2: Roadmap

Model Uncertainty

States, actions, transitions, and rewards

Compare Decisions

Policies, returns, and values

Chapter 2: Roadmap

Model Uncertainty

States, actions, transitions, and rewards

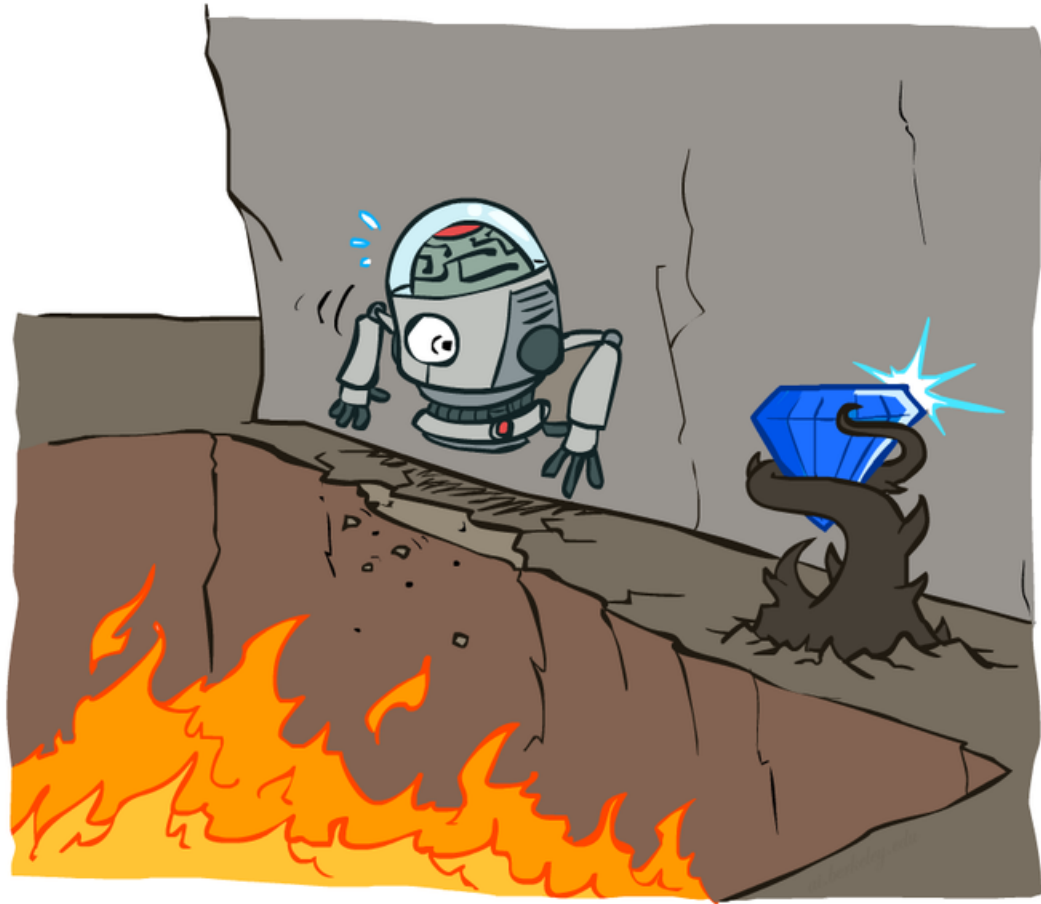
Compare Decisions

Policies, returns, and values

Compute a Policy

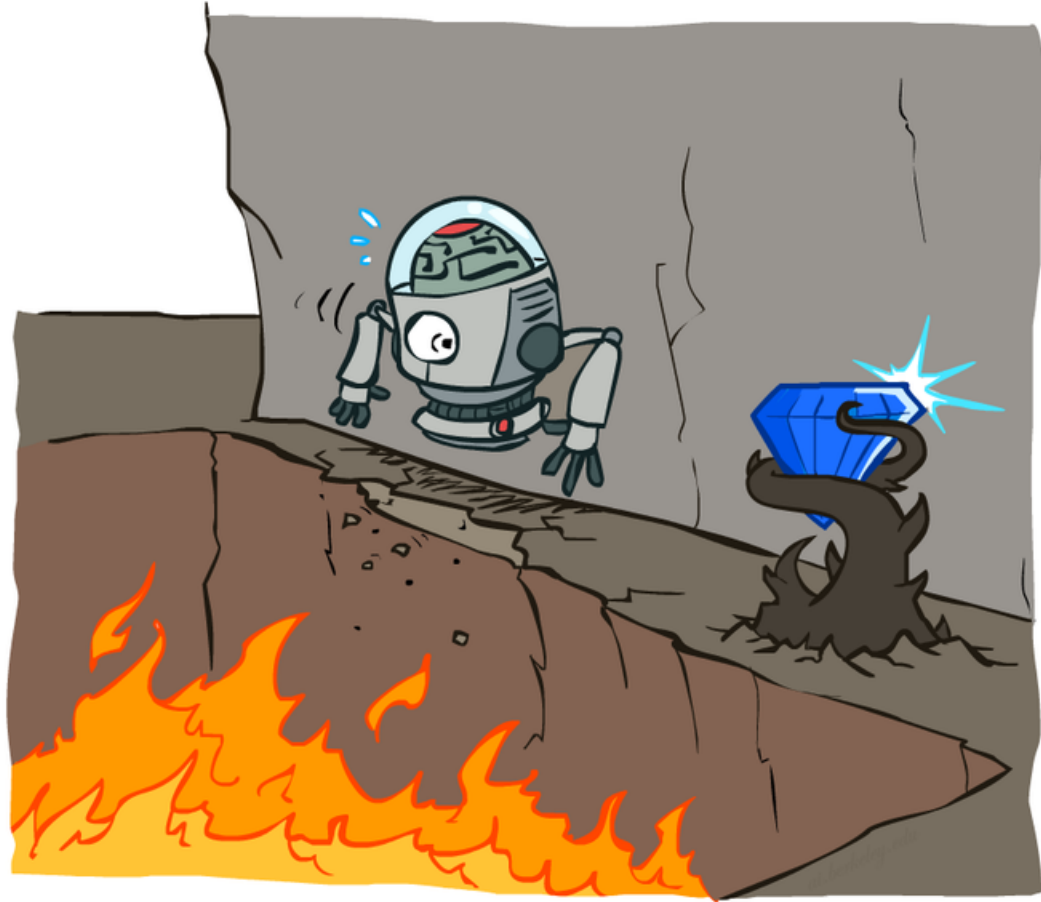
Value iteration and policy iteration

Uncertain Outcomes



C1: one action fixes the next state

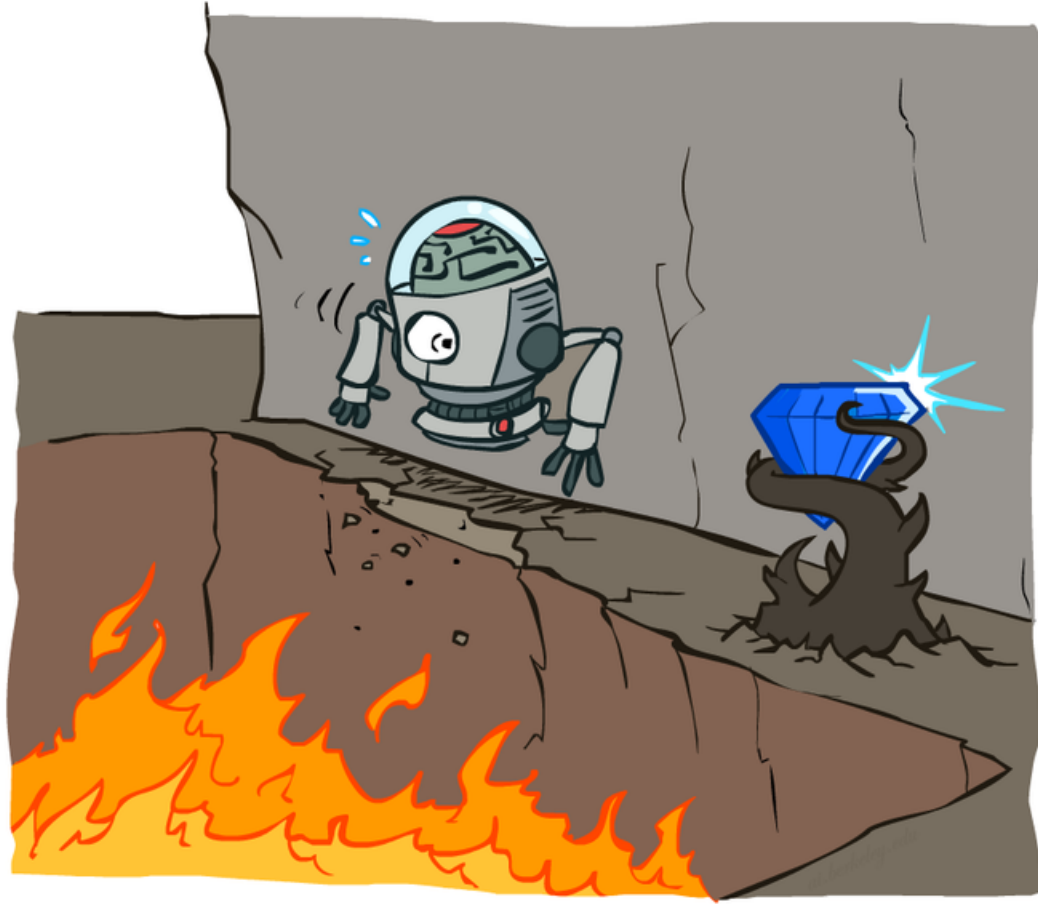
Uncertain Outcomes



C1: one action fixes the next state

C2: one action, several possible states

Uncertain Outcomes

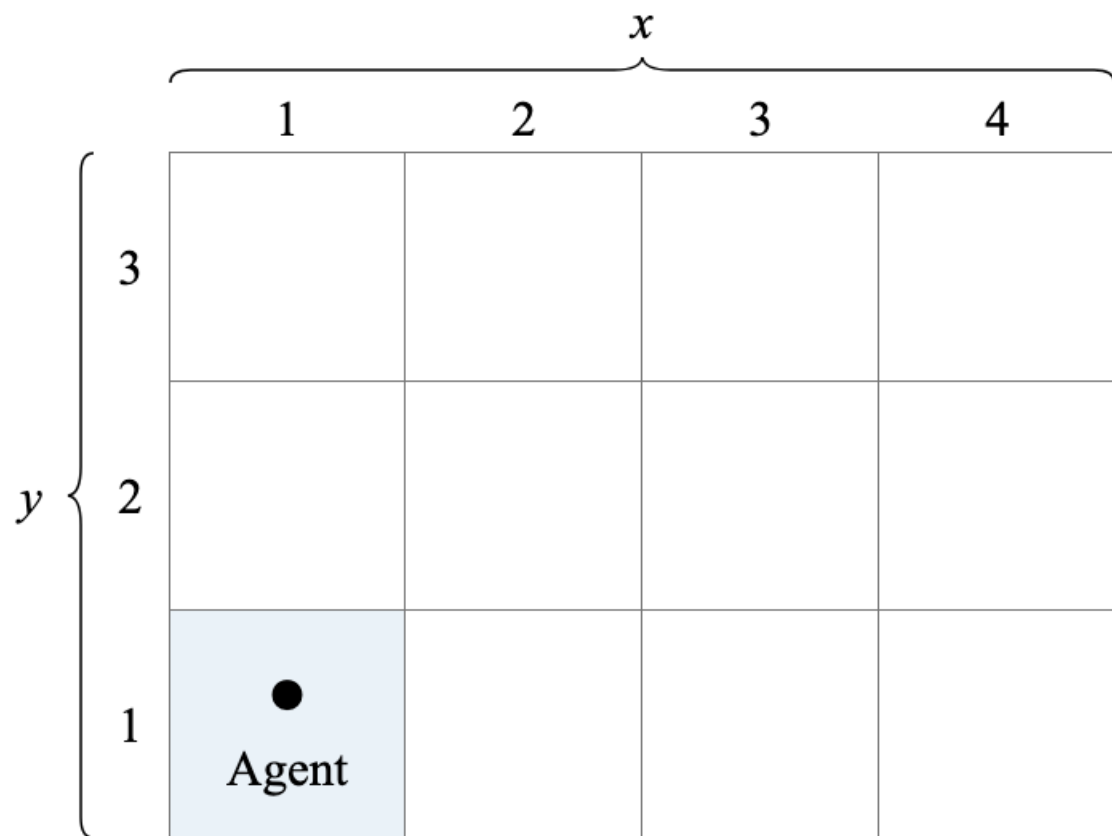


C1: one action fixes the next state

C2: one action, several possible states

Outcome probabilities guide decisions

A Grid World with Noisy Actions

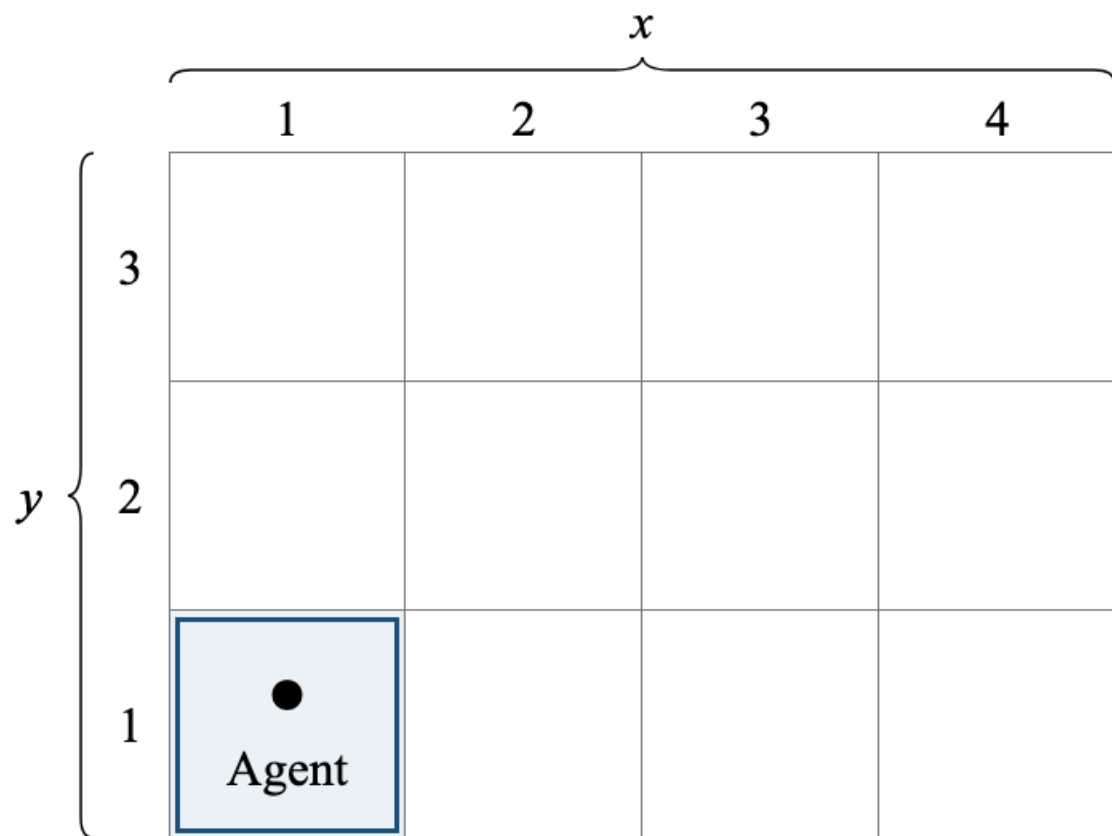


State (x, y)

Agent at $(1, 1)$

x increases right; y increases up

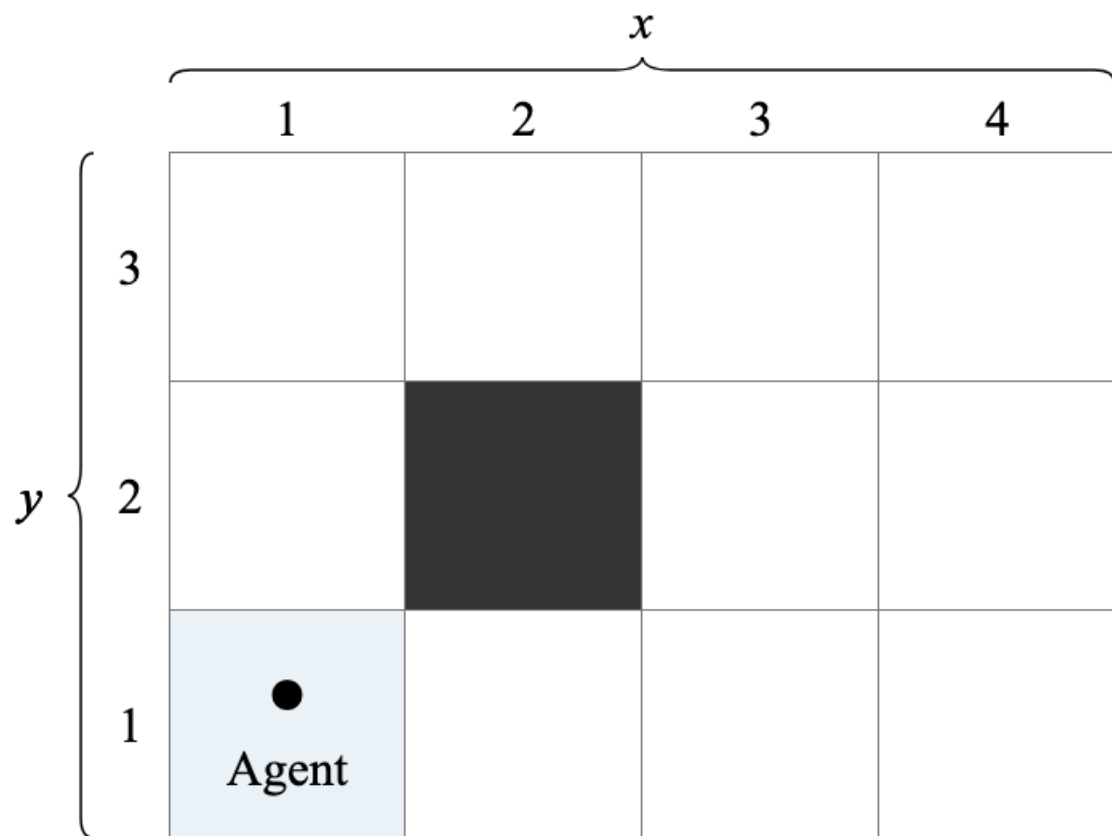
A Grid World with Noisy Actions



Regular states

Available actions: North, East, South,
West

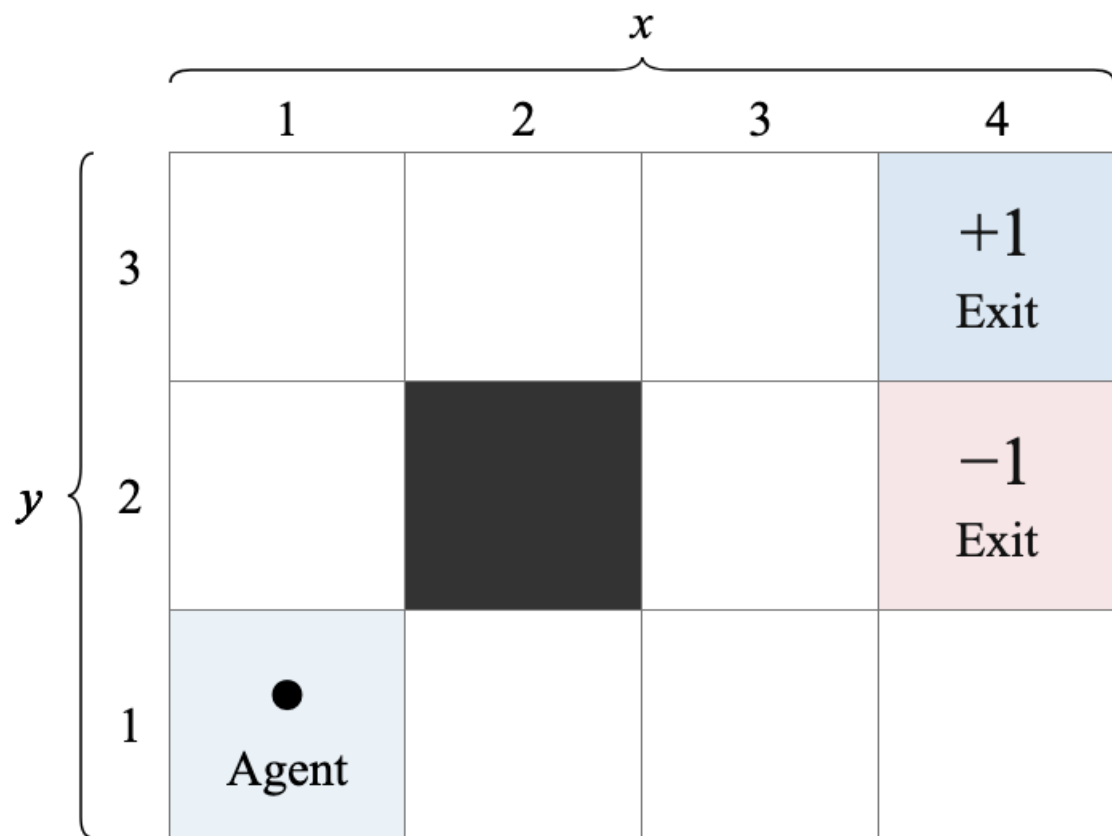
A Grid World with Noisy Actions



Wall at (2, 2)

A move into the wall leaves the agent in place

A Grid World with Noisy Actions

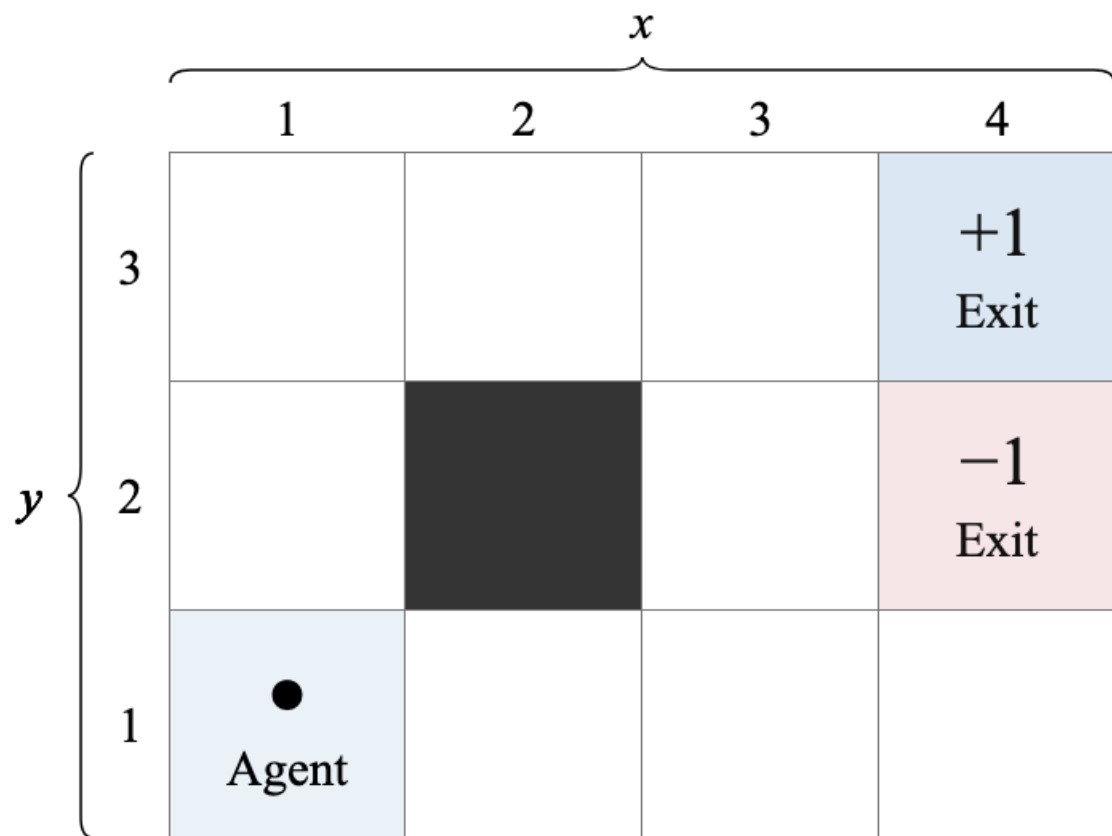


Exit states

Only available action: Exit

Exit pays +1 or -1, then the episode ends

A Grid World with Noisy Actions

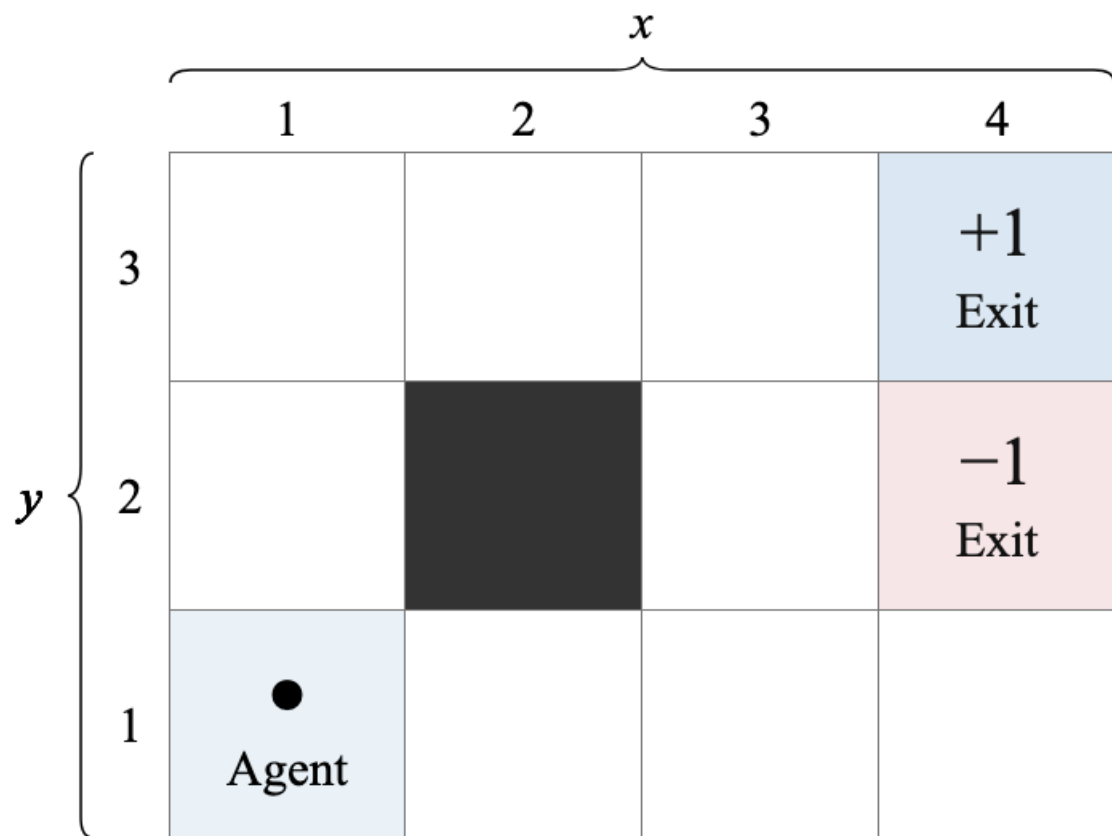


Movement reward

Every compass action pays -0.1

Including blocked moves and moves into exit squares

A Grid World with Noisy Actions

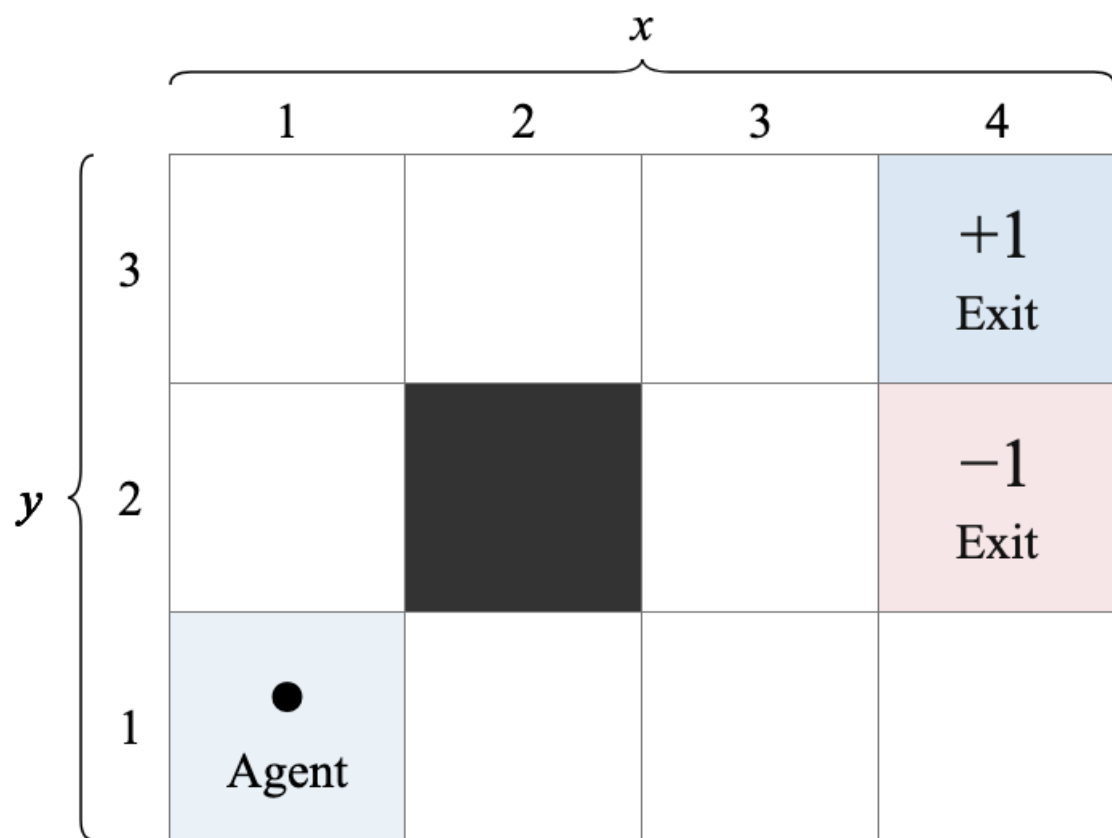


Noisy movement

Intended direction: 0.8

Each sideways direction: 0.1

A Grid World with Noisy Actions



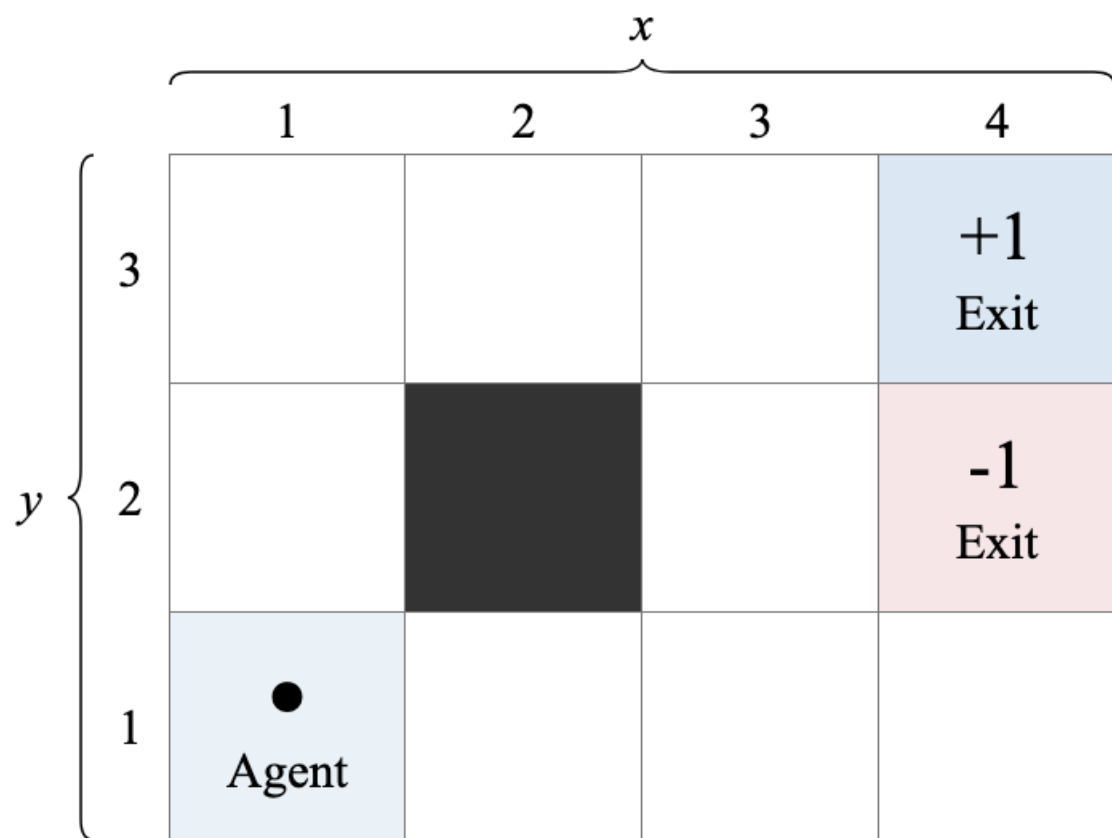
Noisy movement

Intended direction: 0.8

Each sideways direction: 0.1

The agent observes its actual square after every action

Deterministic Grid World

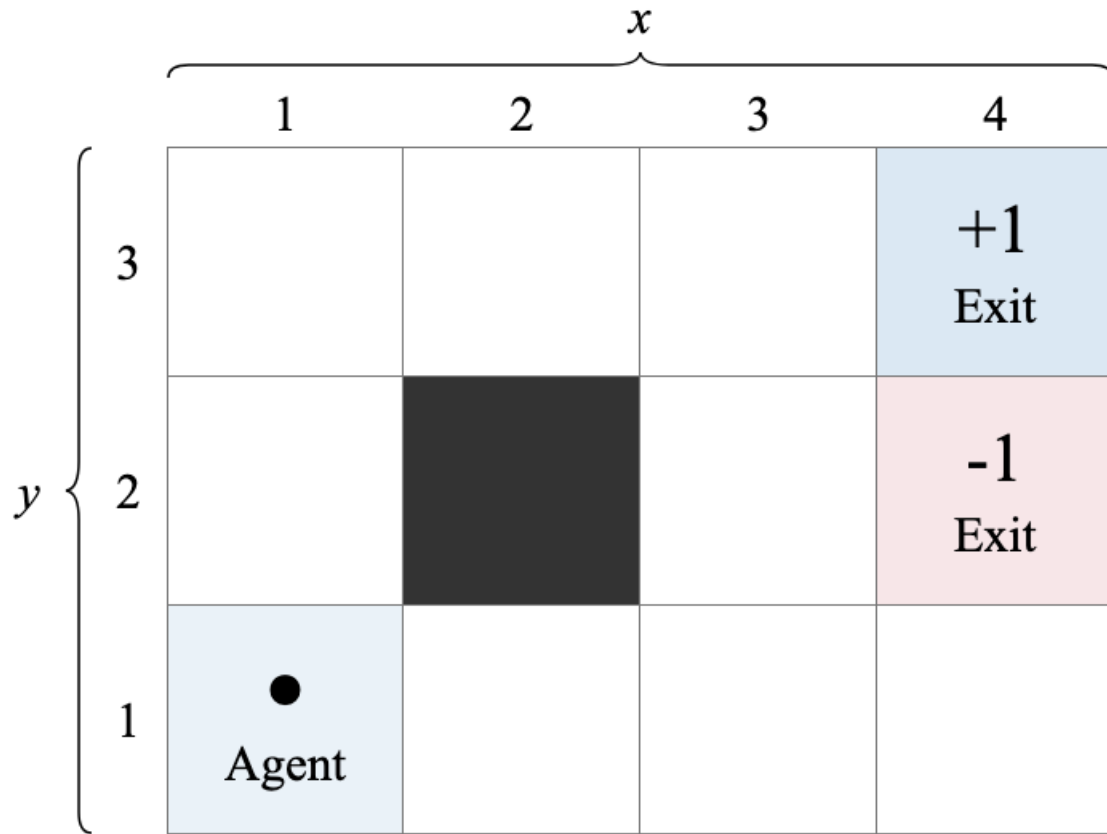


Choose a move

Last reward: 0.0

Total reward: 0.0

Stochastic Grid World



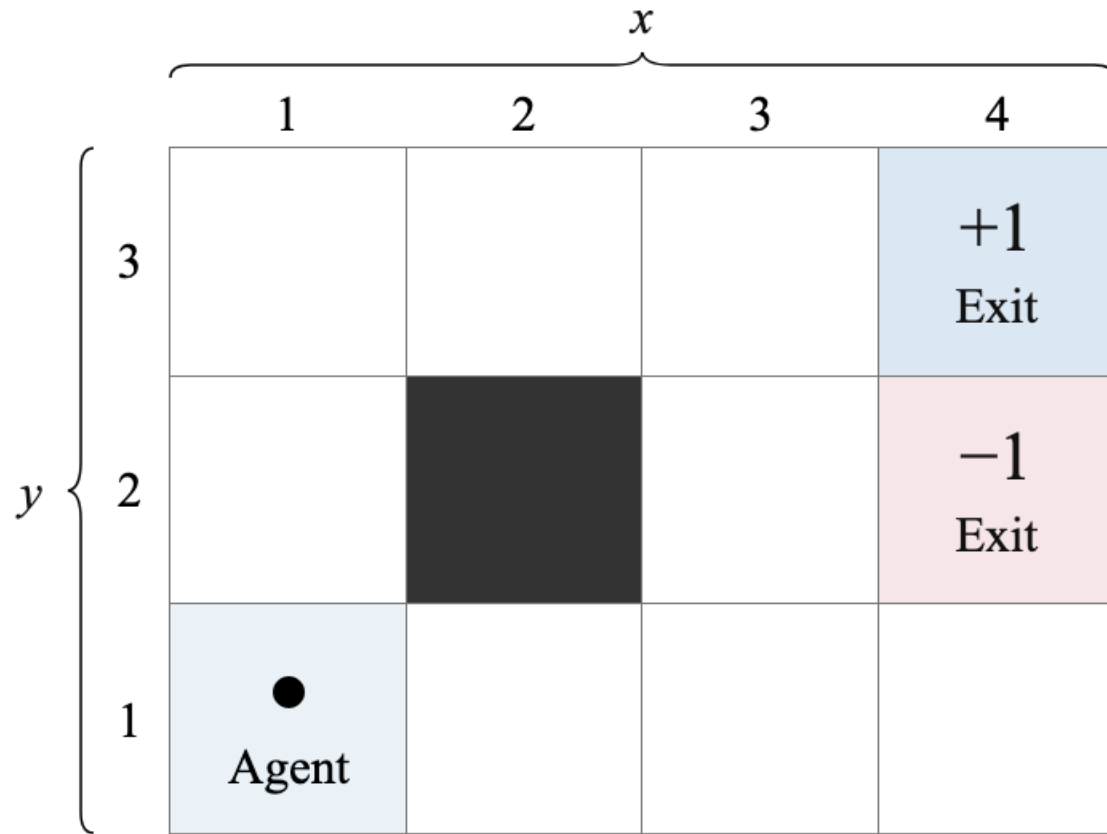
Click an available action

Intended direction: 0.8

Each sideways direction: 0.1

North, Then North

TASK

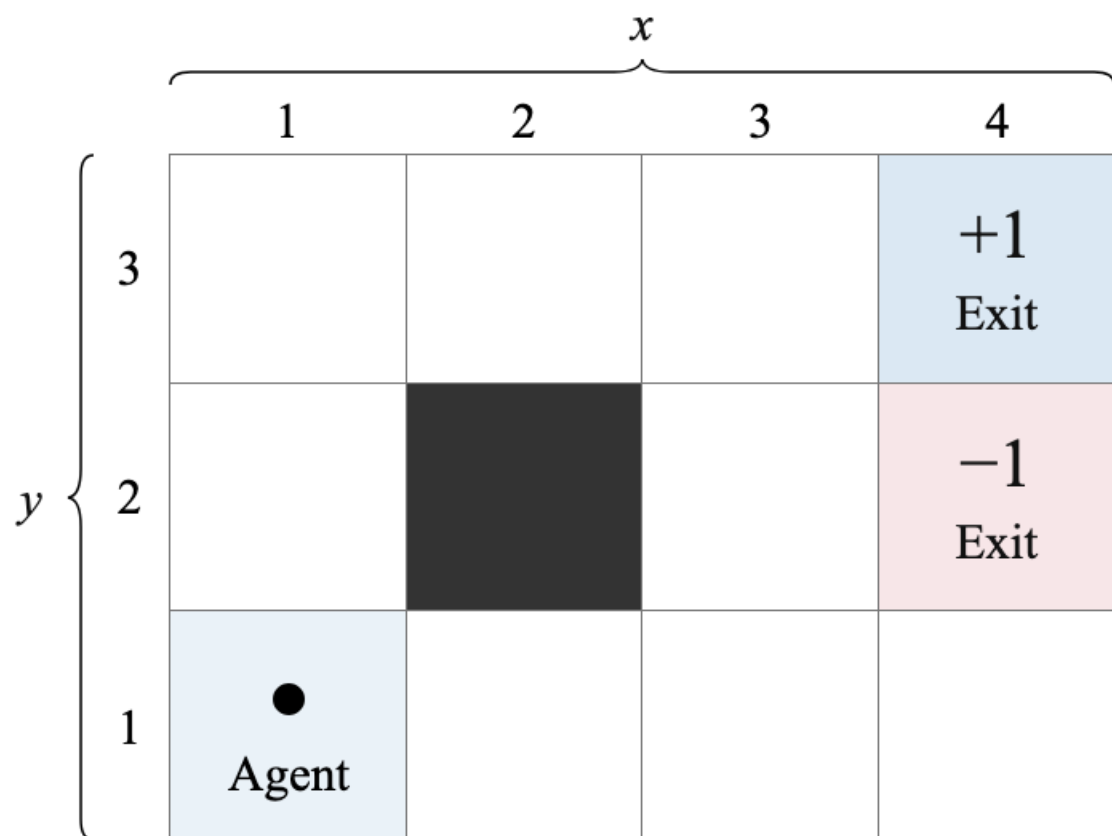


Start at (1, 1)

Choose North twice

What is the probability of ending at (1, 2)?

North, Then North

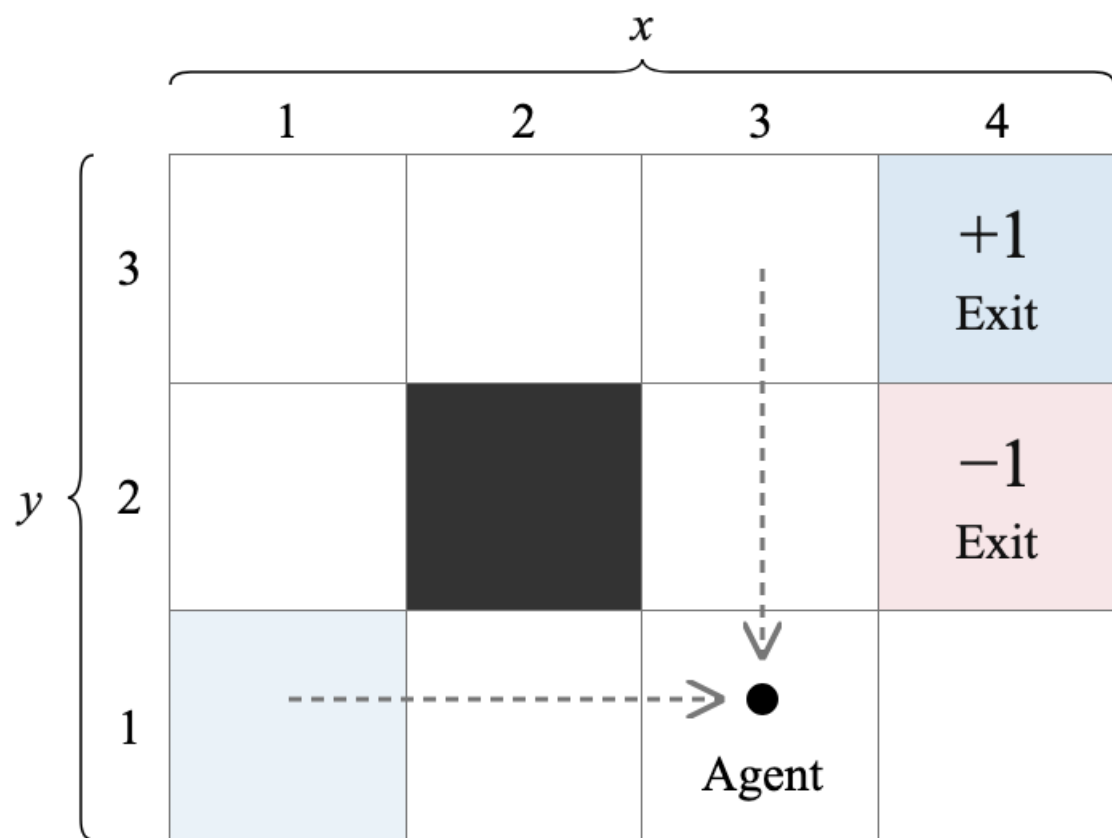


Start at (1, 1)

Choose North twice

What is the probability of ending at (1, 2)?

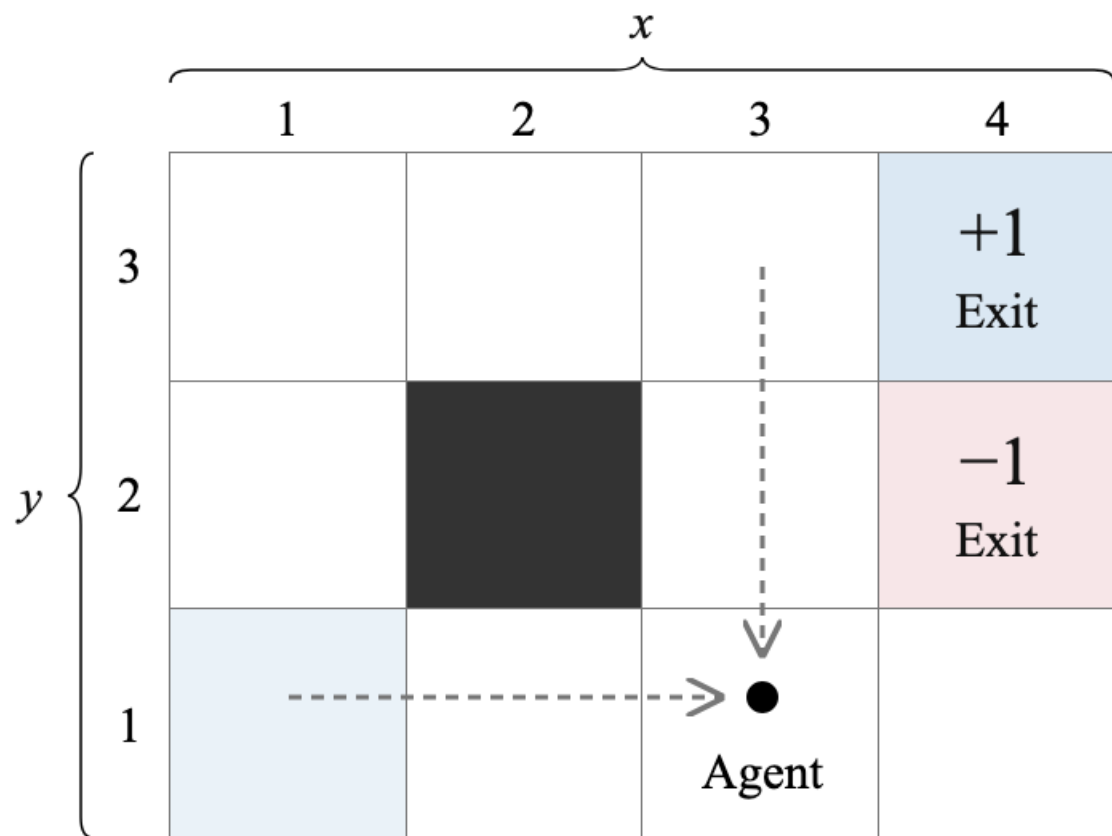
The Transition Model and the Markov Property



A: $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1)$

B: $(3, 3) \rightarrow (3, 2) \rightarrow (3, 1)$

The Transition Model and the Markov Property



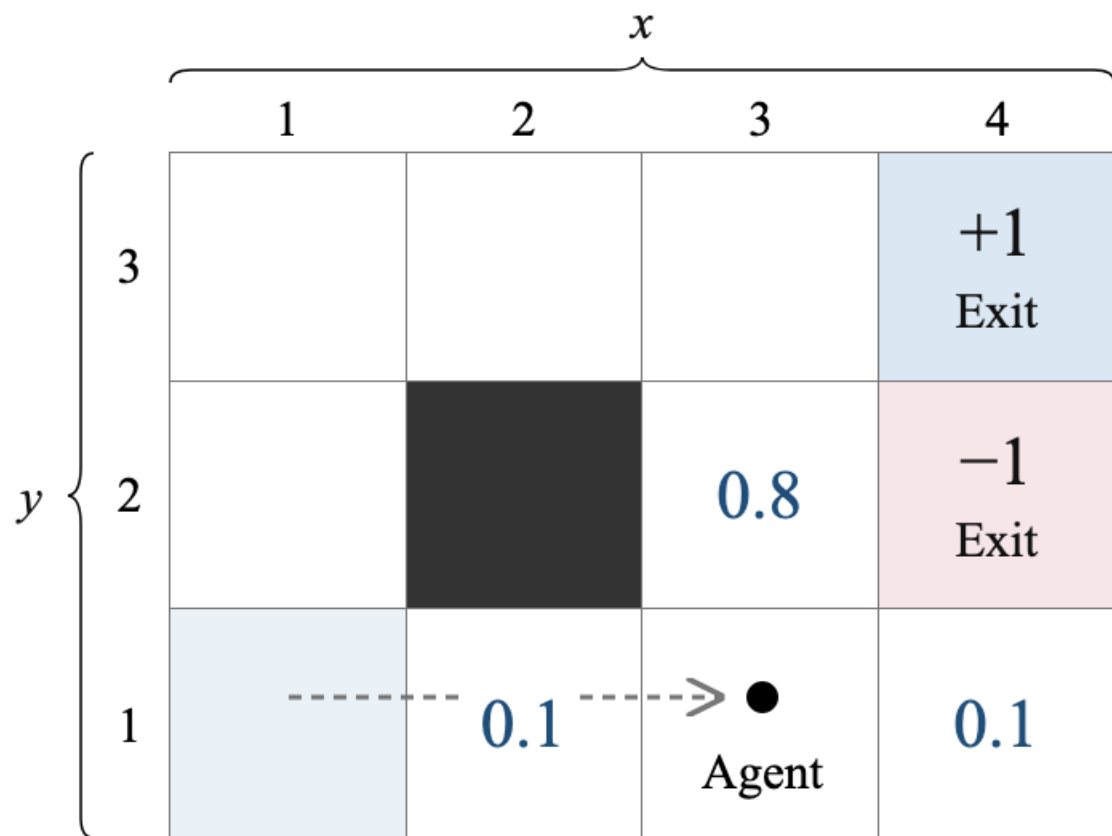
A: $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1)$

B: $(3, 3) \rightarrow (3, 2) \rightarrow (3, 1)$

Same state: $(3, 1)$

Same action: North

The Transition Model and the Markov Property



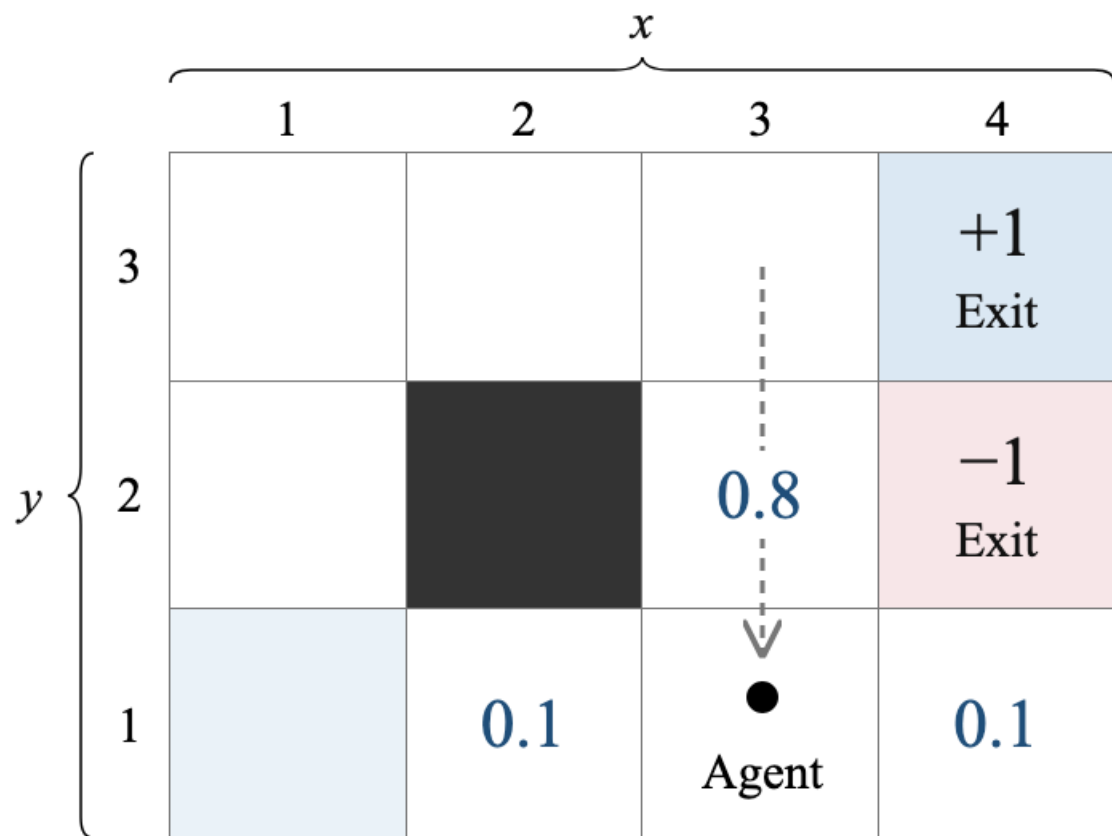
A: $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1)$

B: $(3, 3) \rightarrow (3, 2) \rightarrow (3, 1)$

Same state: $(3, 1)$

Same action: North

The Transition Model and the Markov Property



A: $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1)$

B: $(3, 3) \rightarrow (3, 2) \rightarrow (3, 1)$

Same state: $(3, 1)$

Same action: North

Different histories

Same next-state probabilities

The Transition Model and the Markov Property

		x			
		1	2	3	4
y	3				+1 Exit
	2			0.8	-1 Exit
	1		0.1	● Agent	0.1

A: $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1)$

B: $(3, 3) \rightarrow (3, 2) \rightarrow (3, 1)$

Same state: $(3, 1)$

Same action: North

$$T(s, a, s') = P(s' \mid s, a)$$

Current state, action, next state

The current state and action suffice

Earlier history adds nothing

The Transition Model and the Markov Property

		x			
		1	2	3	4
y	3				+1 Exit
	2			0.8	-1 Exit
	1		0.1	● Agent	0.1

A: $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1)$

B: $(3, 3) \rightarrow (3, 2) \rightarrow (3, 1)$

Same state: $(3, 1)$

Same action: North

$$T(s, a, s') = P(s' \mid s, a)$$

Current state, action, next state

The current state and action suffice

Earlier history adds nothing

$$P(s_{t+1} \mid s_t, a_t, s_{t-1}, a_{t-1}, \dots) = P(s_{t+1} \mid s_t, a_t)$$

Defining a Markov Decision Process

$s \in S$

The possible states

Defining a Markov Decision Process

$$s \in S$$

The possible states

$$a \in A(s)$$

The actions available in state s

Defining a Markov Decision Process

$$s \in S$$

The possible states

$$a \in A(s)$$

The actions available in state s

$$T(s, a, s')$$

The probability of each next state

Defining a Markov Decision Process

$$s \in S$$

The possible states

$$a \in A(s)$$

The actions available in state s

$$T(s, a, s')$$

The probability of each next state

$$R(s, a, s')$$

The reward for that transition

Defining a Markov Decision Process

$s \in S$

The possible states

$a \in A(s)$

The actions available in state s

$T(s, a, s')$

The probability of each next state

$R(s, a, s')$

The reward for that transition

Objective

Maximise expected cumulative reward

Also specify the initial state or distribution and any terminal states

A Fixed Plan Cannot Respond to a Slip

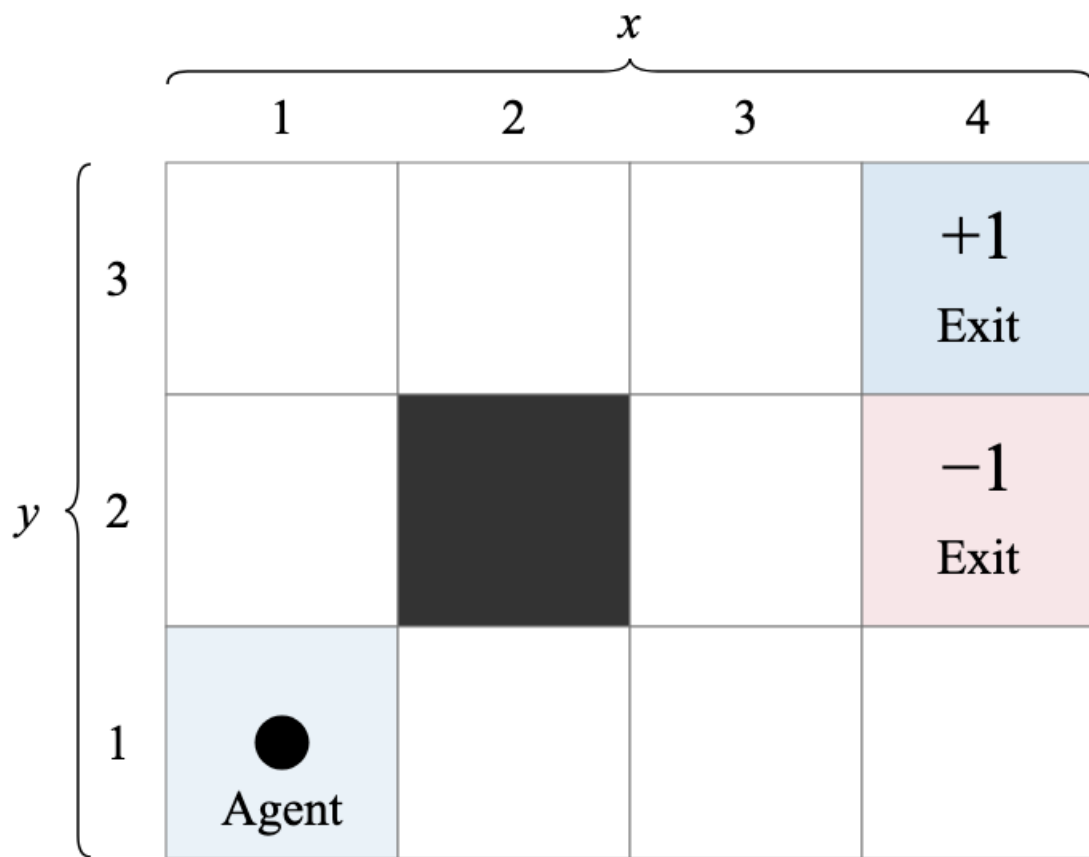
Plan

1. North

2. North

3. East

...



Start at (1, 1)

Follow a fixed list of actions

A Fixed Plan Cannot Respond to a Slip

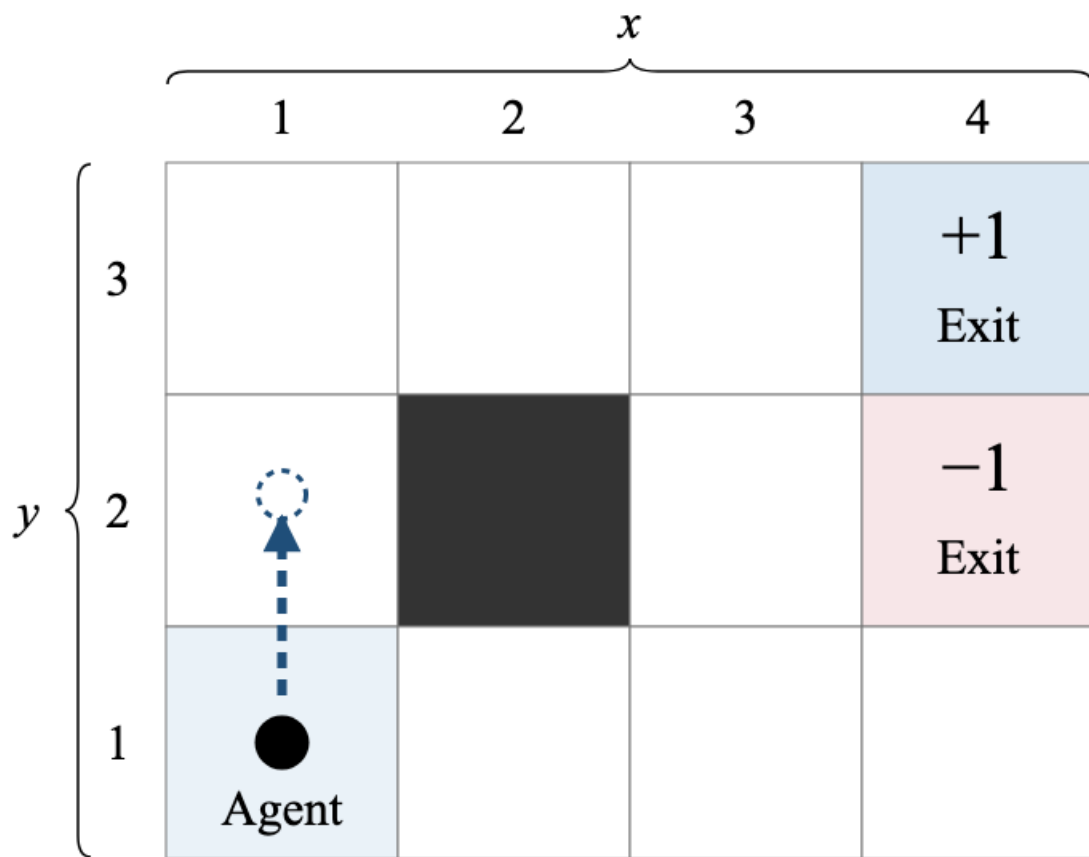
Plan

1. North

2. North

3. East

...



First action: North

We intend to move up

Dashed arrow: intended direction

A Fixed Plan Cannot Respond to a Slip

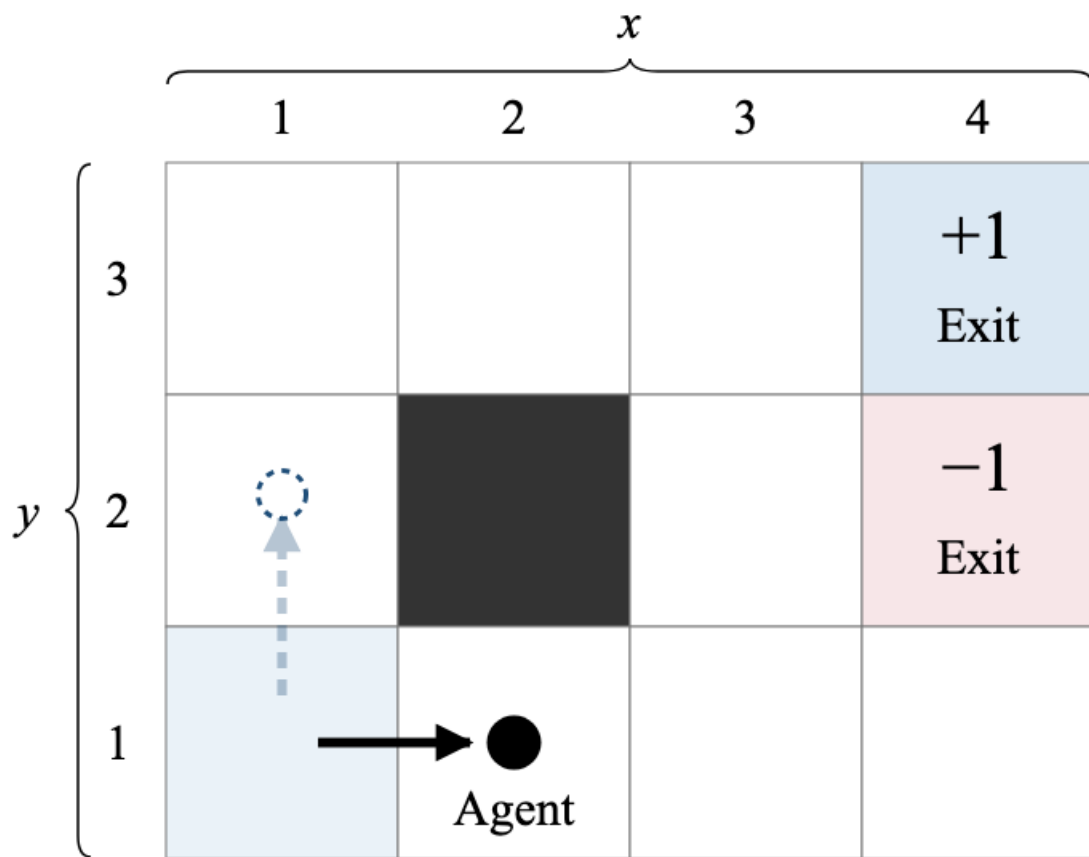
Plan

1. North

2. North

3. East

...



Chosen action: North

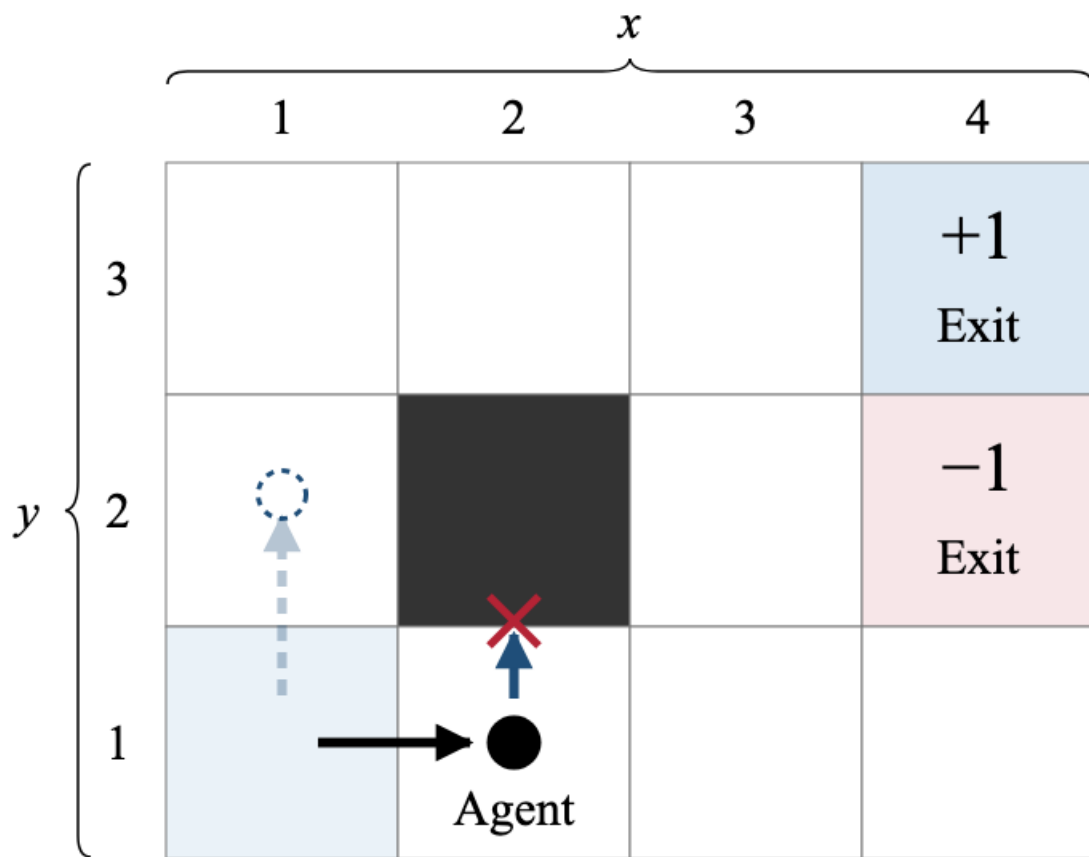
Actual move: East

The agent lands at (2, 1)

A sideways slip, probability 0.1

A Fixed Plan Cannot Respond to a Slip

Plan 1. North 2. North 3. East ...



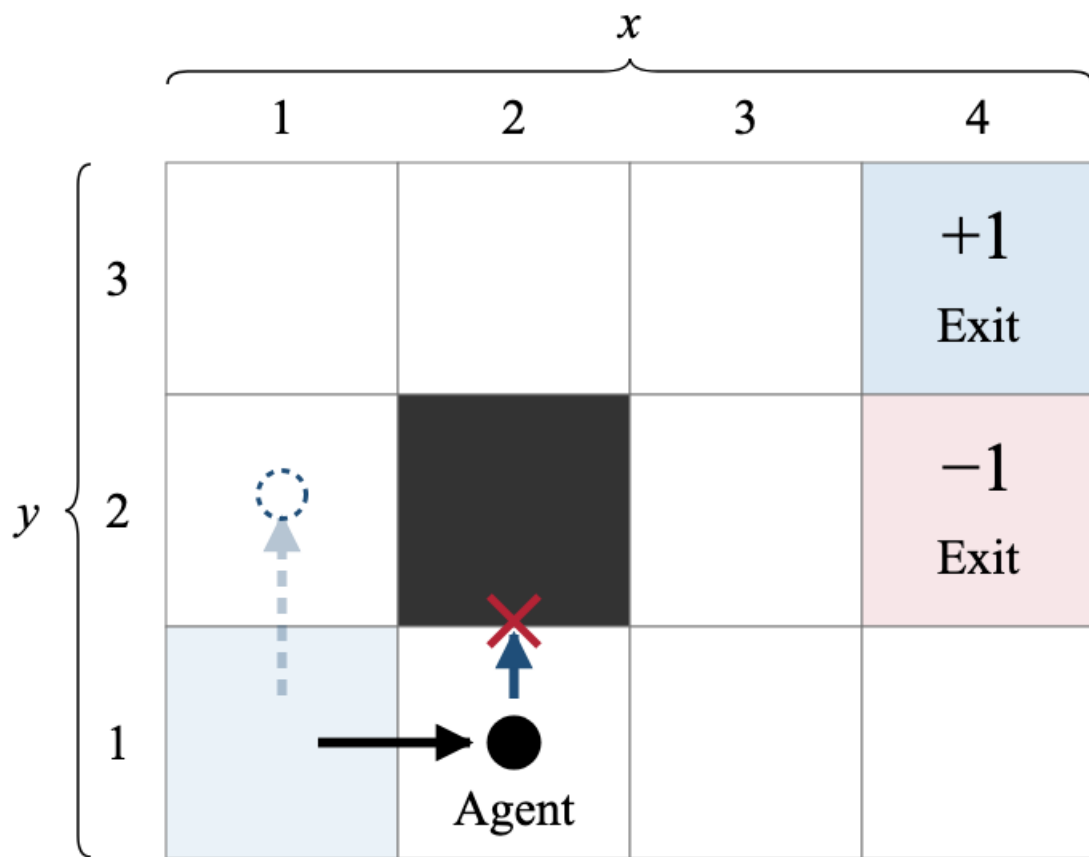
Next on the list: North

North now points into the wall

This move leaves the agent in place

A Fixed Plan Cannot Respond to a Slip

Plan 1. North 2. North 3. East ...

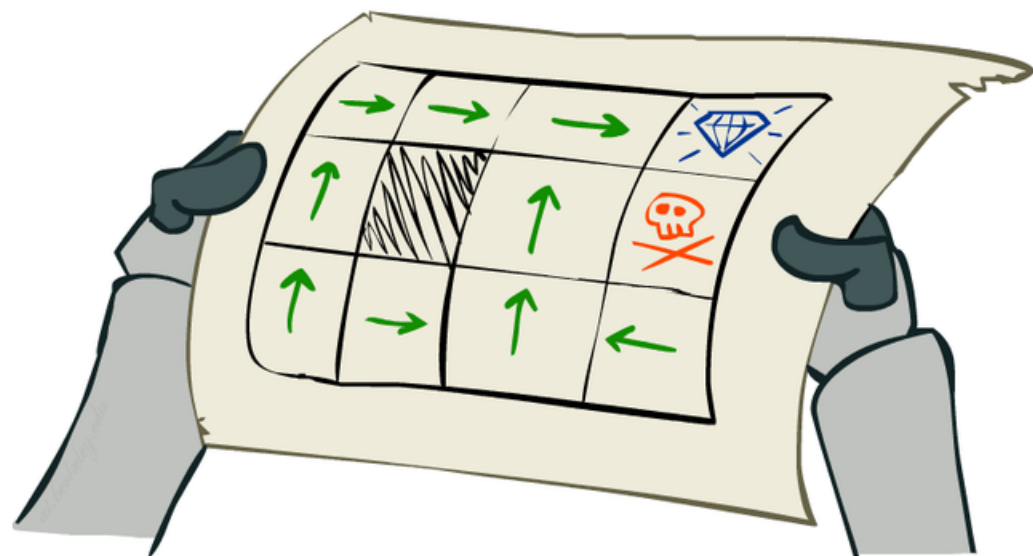


The plan still says North

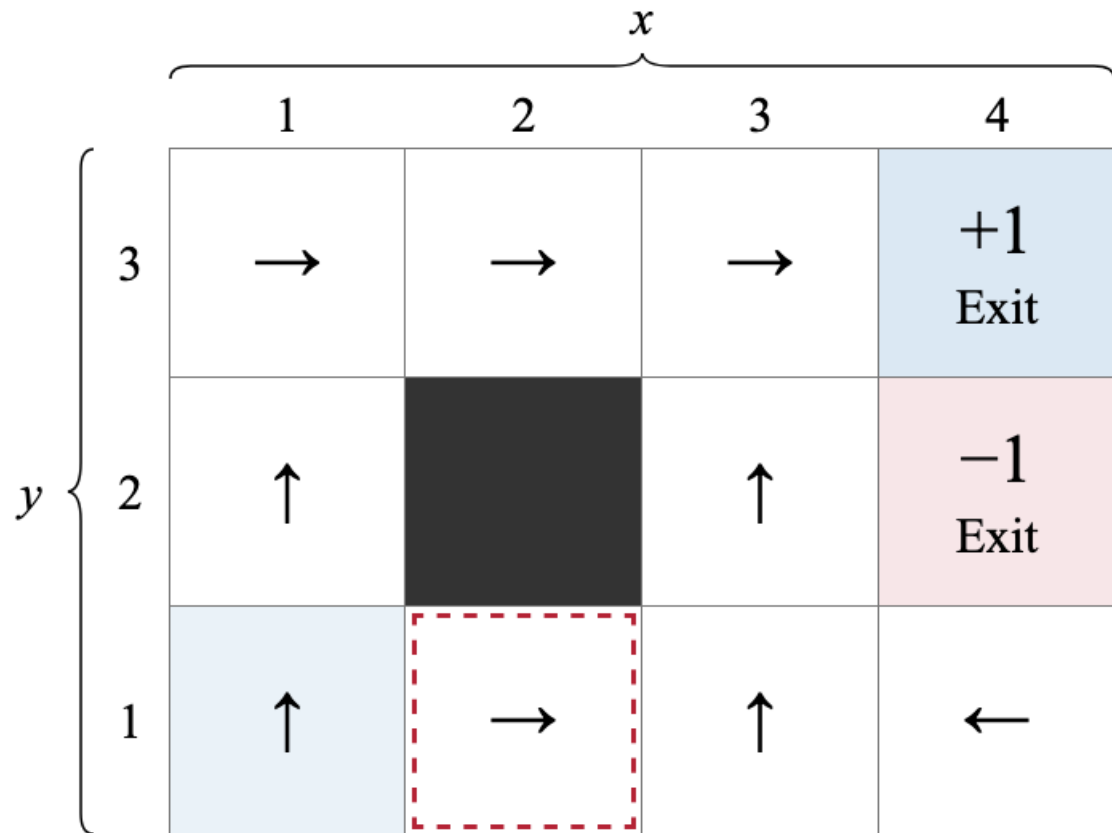
But we are in a different square

We need an action for
where we are now

An MDP Solution Is a Policy

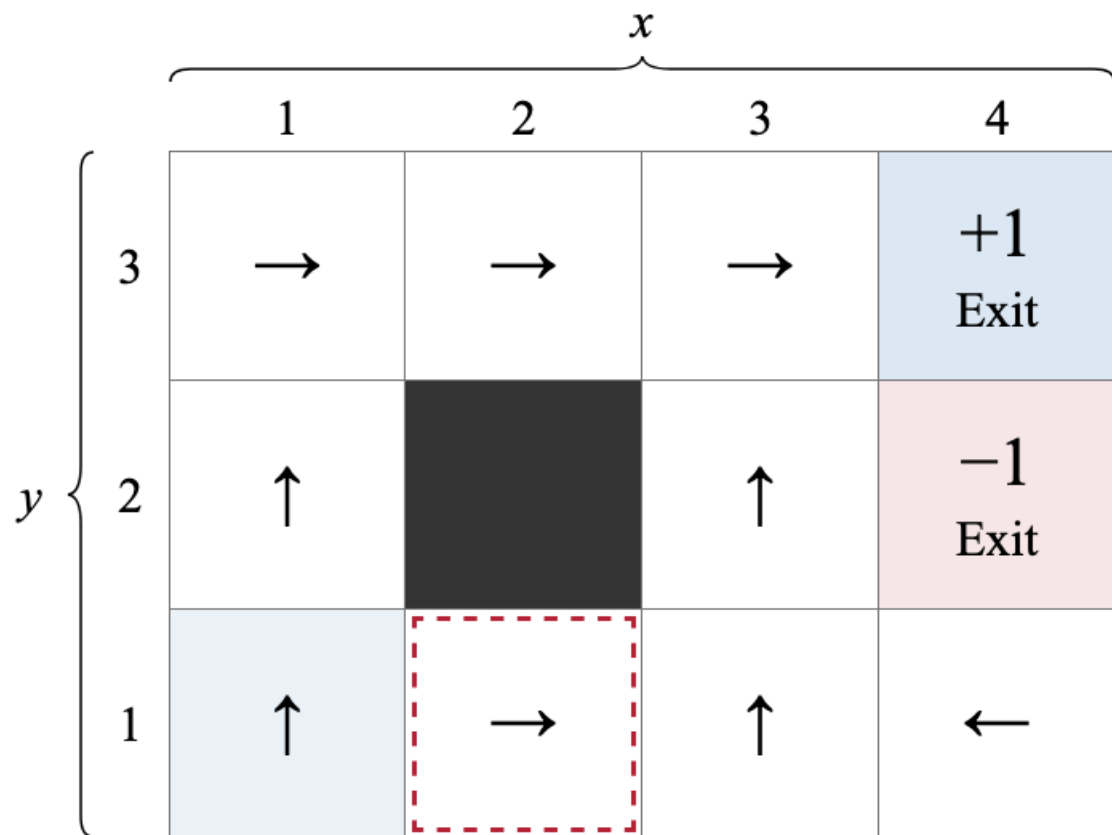


An MDP Solution Is a Policy



A policy π determines
what to do for each state

An MDP Solution Is a Policy

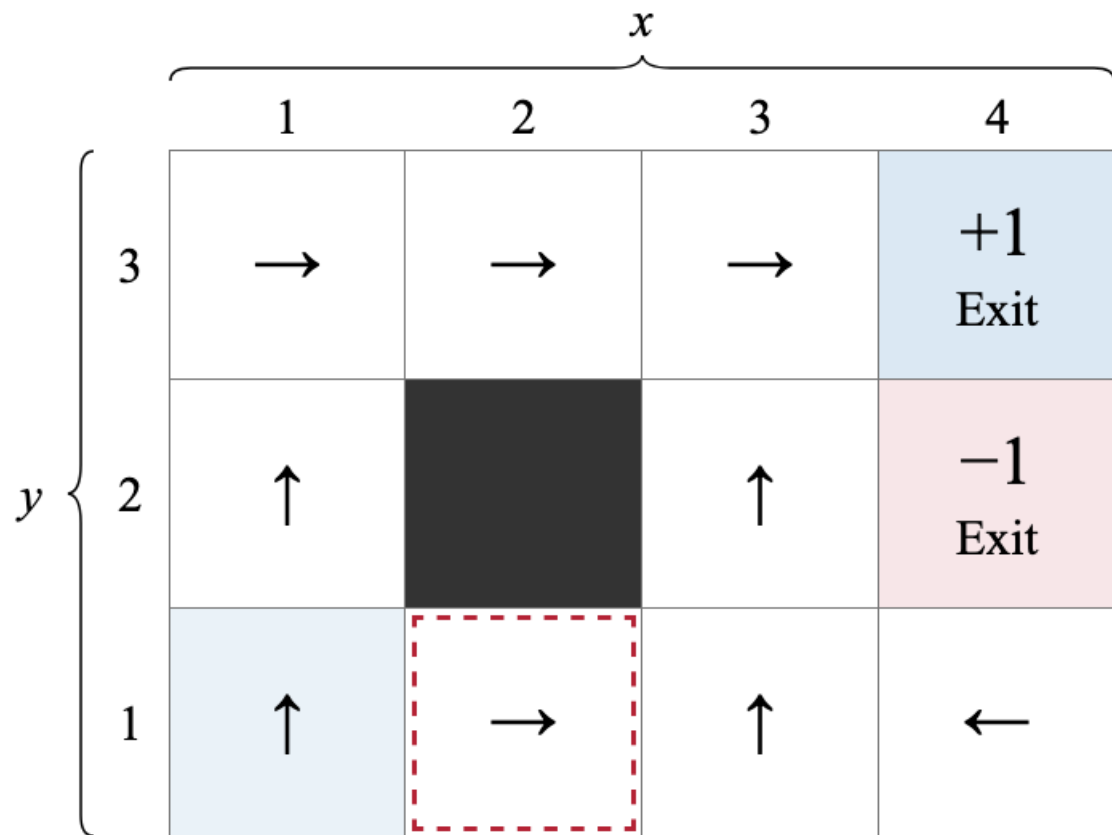


A policy π determines
what to do for each state

After the slip, we are at (2, 1)

$$\pi(2, 1) = \text{East}$$

An MDP Solution Is a Policy



A policy π determines
what to do for each state

After the slip, we are at (2, 1)

$$\pi(2, 1) = \text{East}$$

Observe the state, then use its action

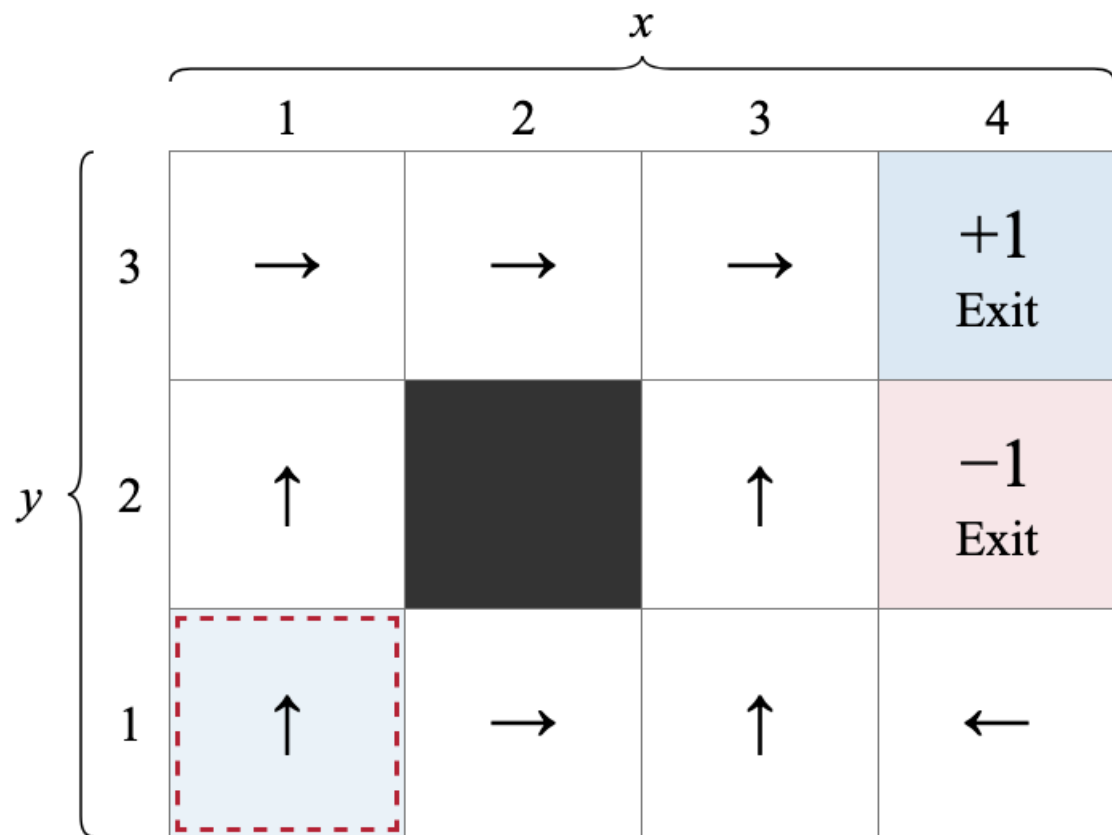
Which Policy Is Better?

TASK

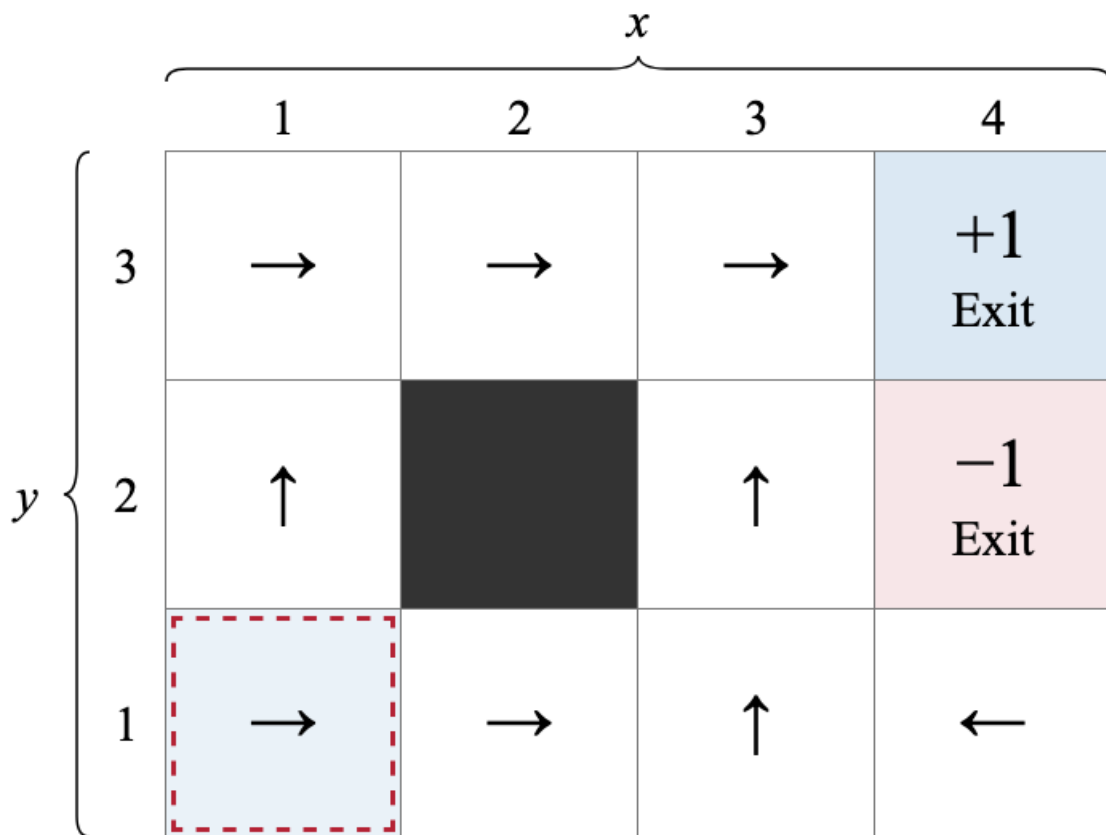
Start: (1, 1). Move: -0.1 . Exit: $+1$ or -1 .

Intended direction: 0.8; each sideways direction: 0.1

Policy A



Policy B

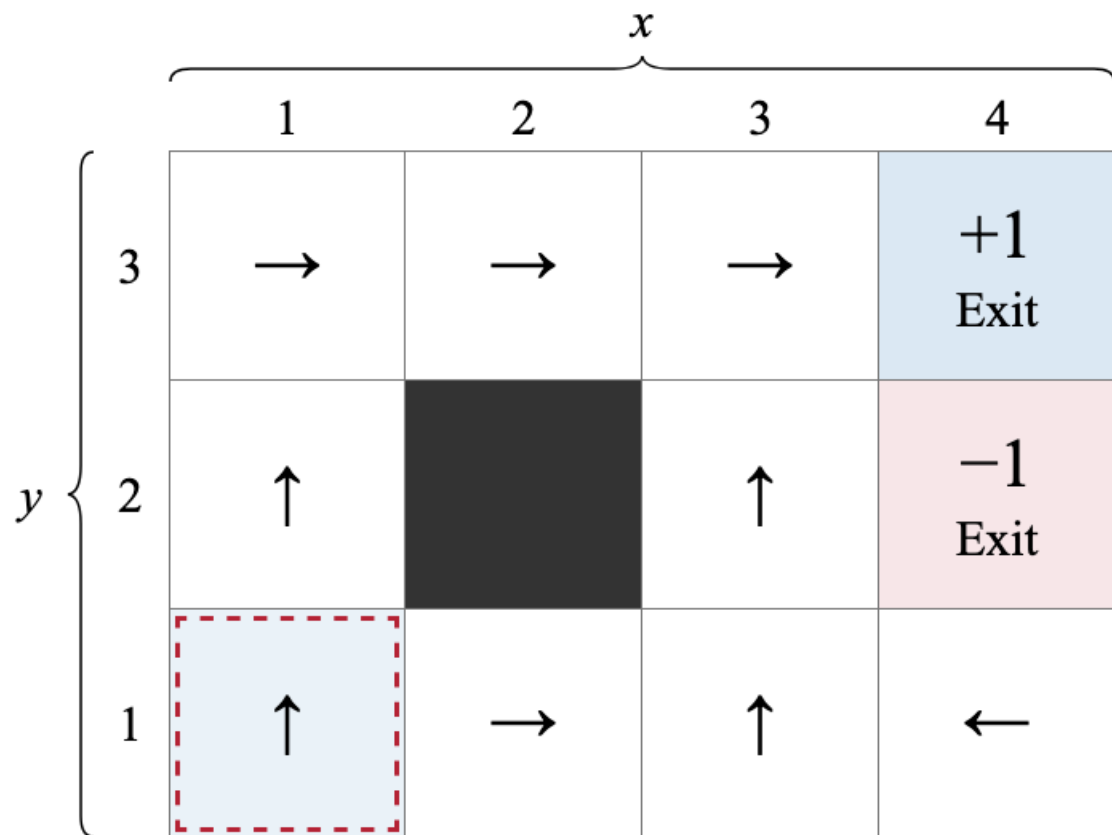


Which Policy Is Better?

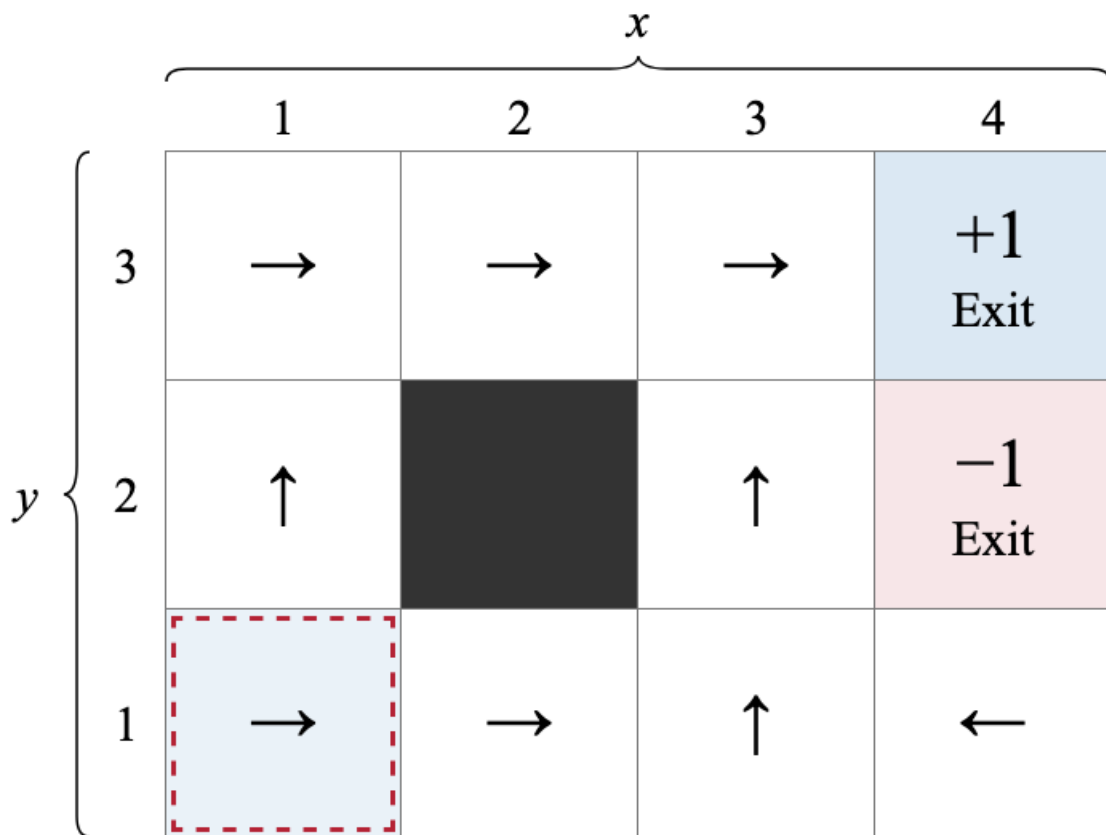
Start: (1, 1). Move: -0.1 . Exit: $+1$ or -1 .

Intended direction: 0.8 ; each sideways direction: 0.1

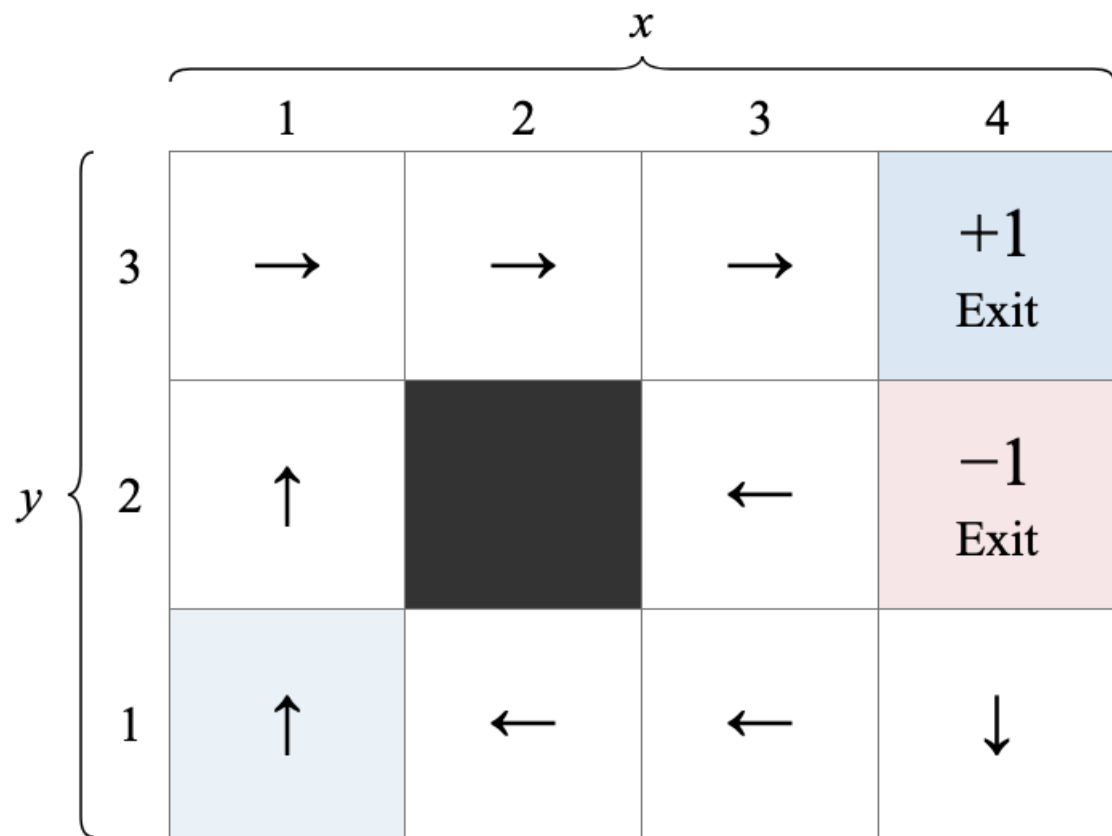
Policy A



Policy B



Living Rewards and the Optimal Policy



Living reward -0.01

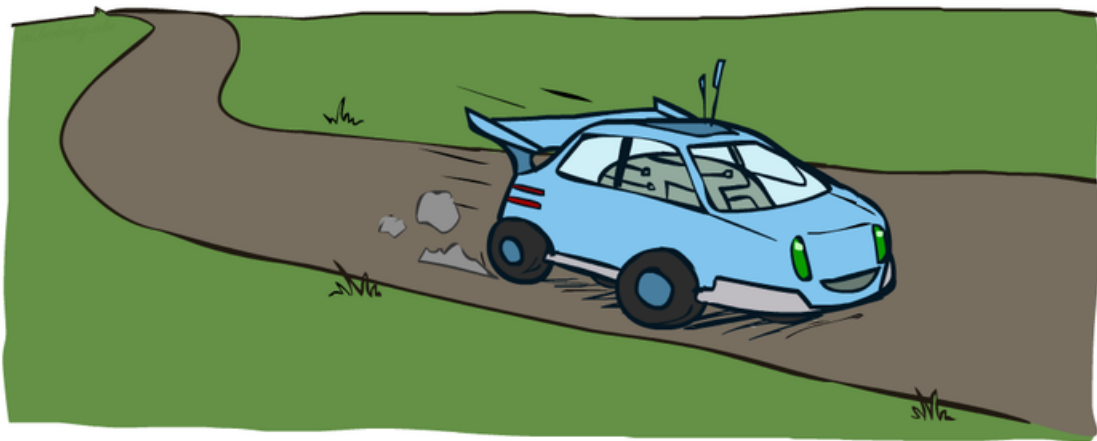
Reward received on every move

Intended direction: 0.8

Each sideways direction: 0.1

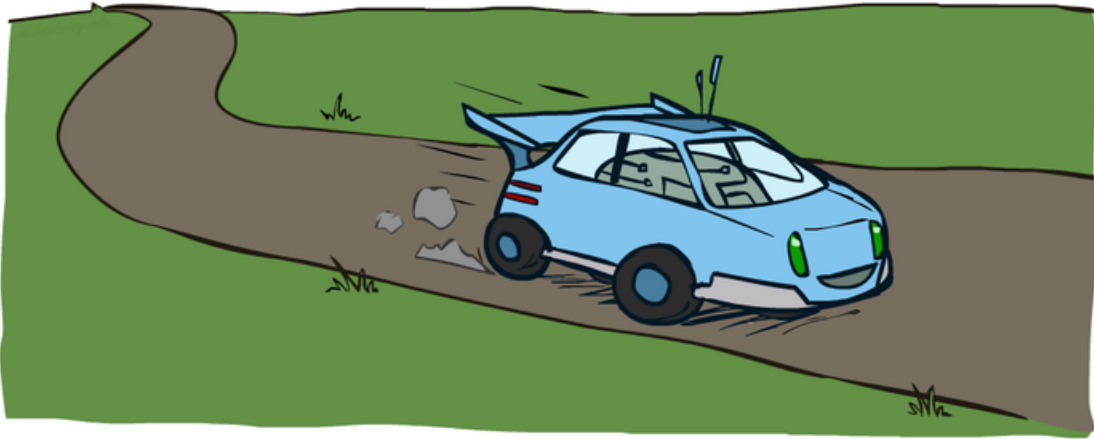
Exit rewards: +1 and -1

Racing Car



The car wants to travel quickly

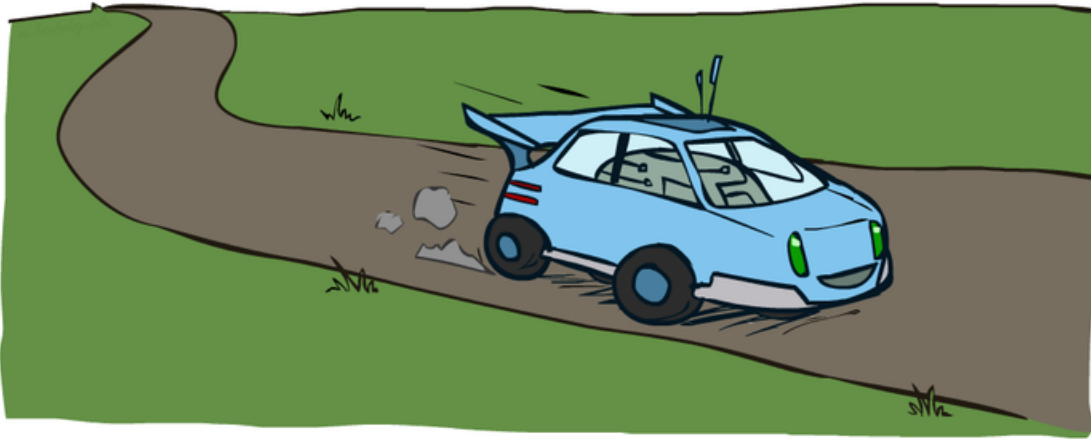
Racing Car



The car wants to travel quickly

Actions: Slow or Fast

Racing Car



The car wants to travel quickly

Actions: Slow or Fast

Temperature changes what happens next

Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward



Cool

Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward

Warm



Cool

Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward

Warm



Cool



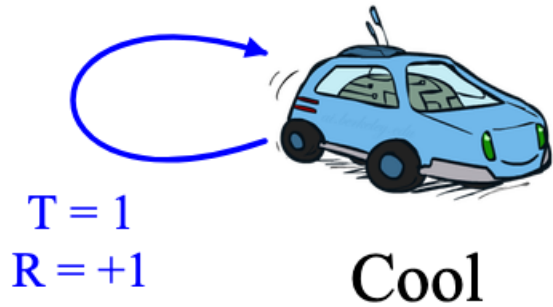
Overheated

Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward

Slow

Warm



Overheated

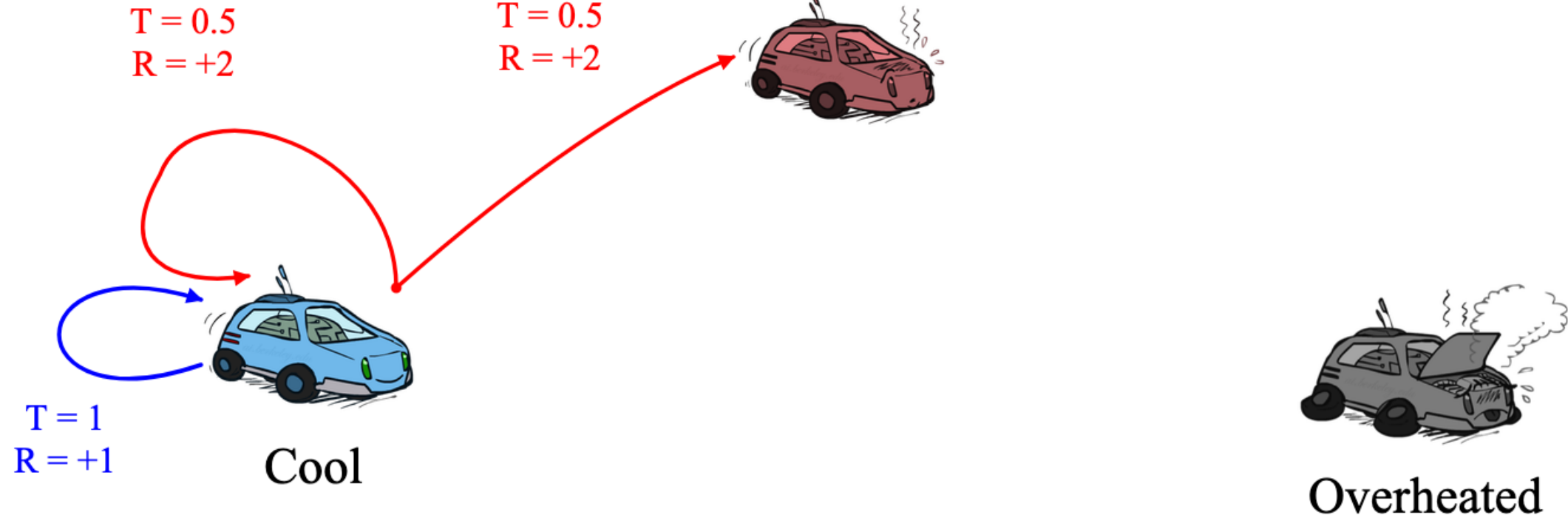
Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward

Slow

Fast

Warm

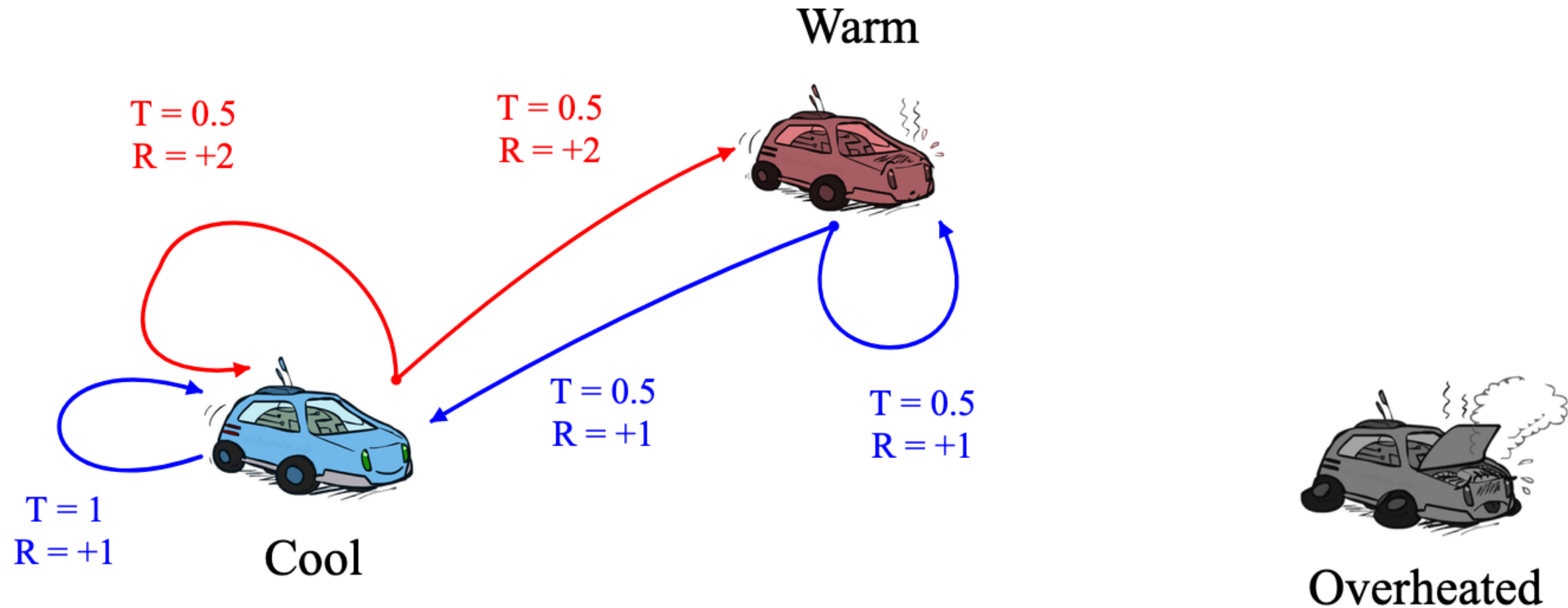


Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward

Slow

Fast

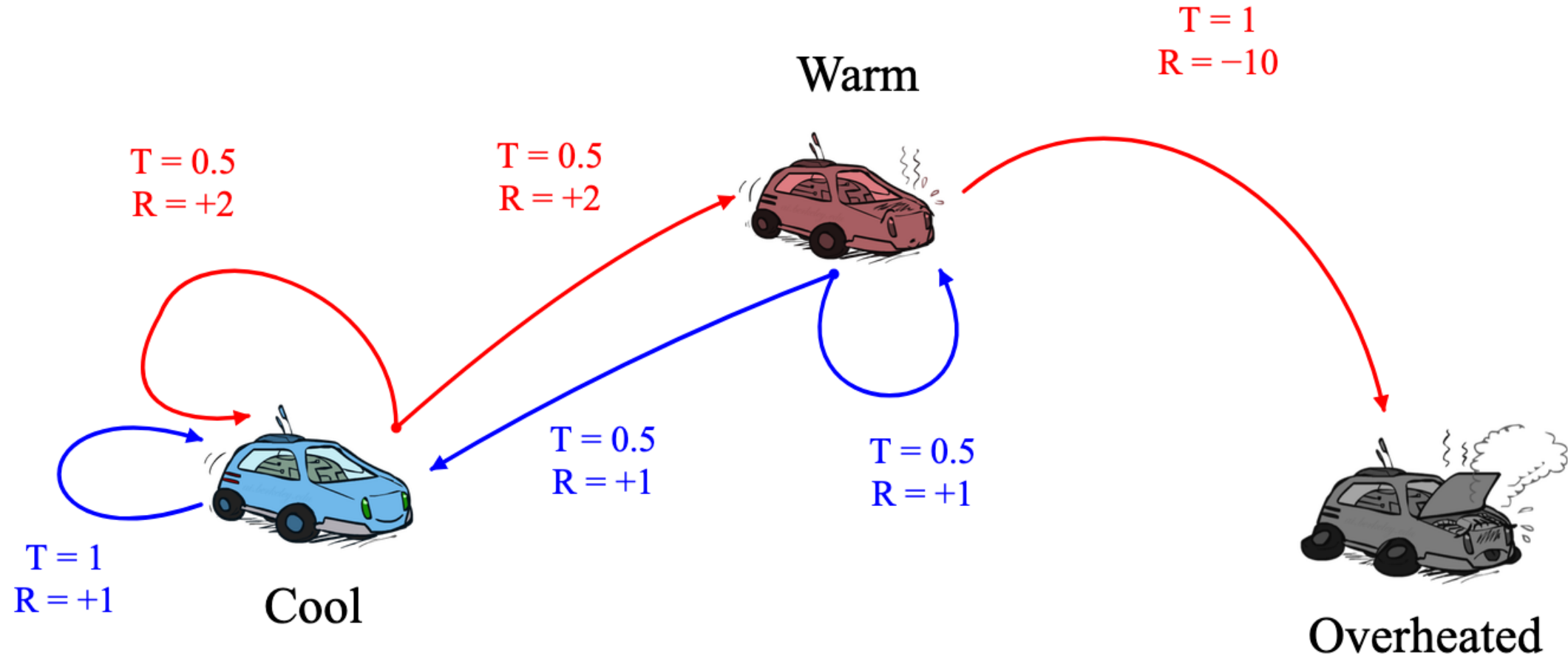


Racing-Car State Diagram

$T(s, a, s')$: transition probability · $R(s, a, s')$: reward

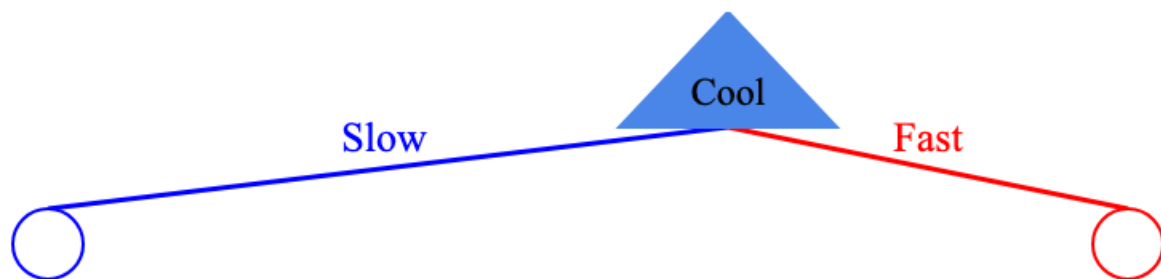
Slow

Fast



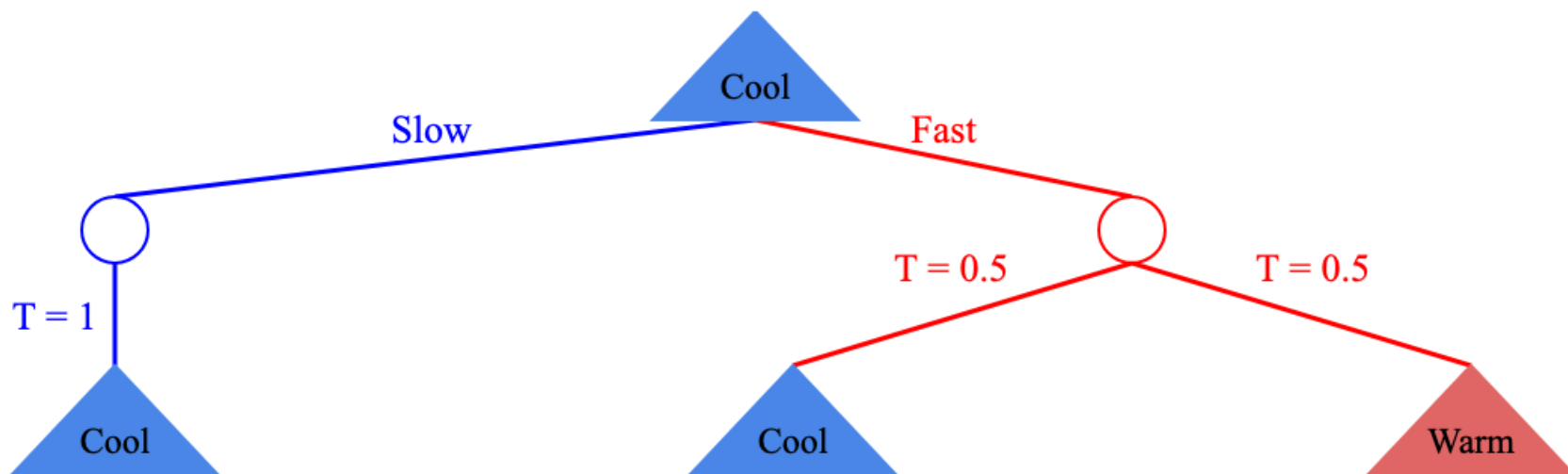
What Is the Optimal Action Here?

Slow: reward +1 · Fast: reward +2; overheating: $-10 \cdot T$: transition probability



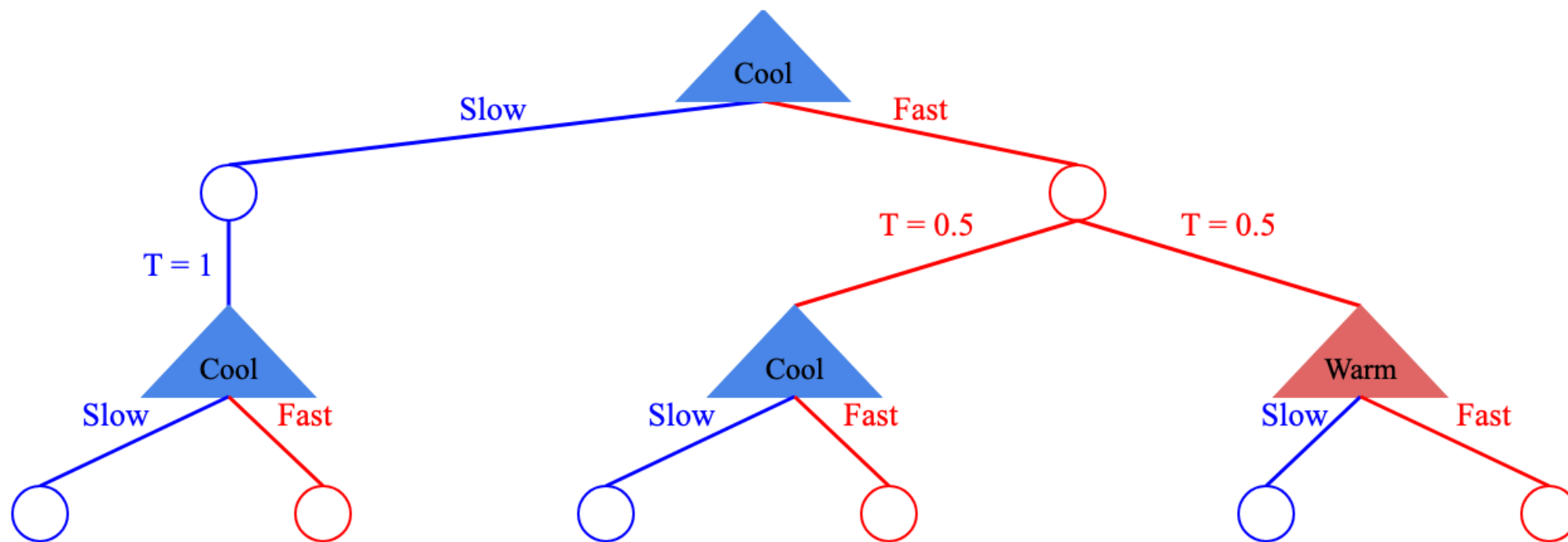
What Is the Optimal Action Here?

Slow: reward +1 · Fast: reward +2; overheating: -10 · T: transition probability



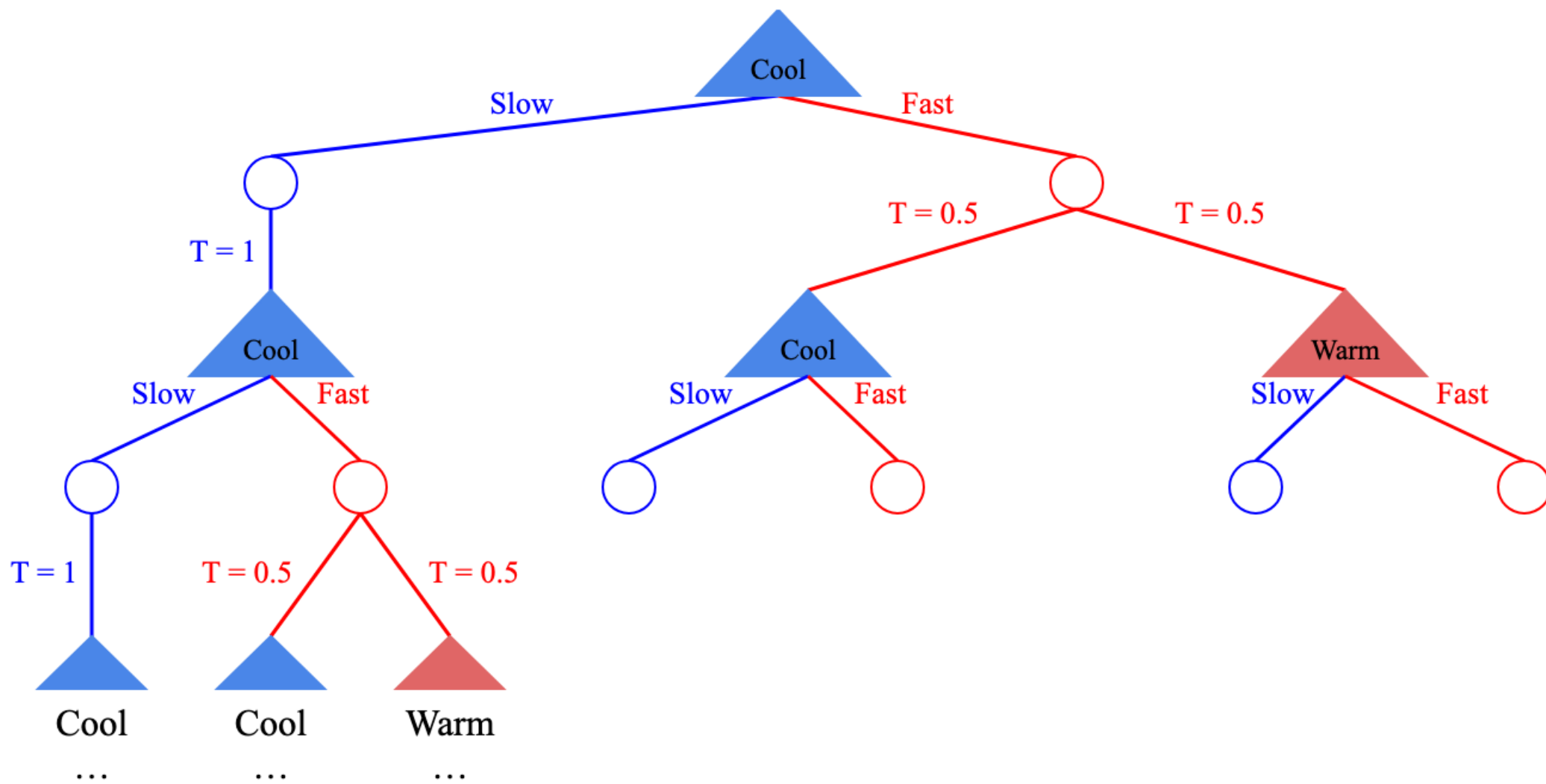
What Is the Optimal Action Here?

Slow: reward +1 · Fast: reward +2; overheating: -10 · T: transition probability



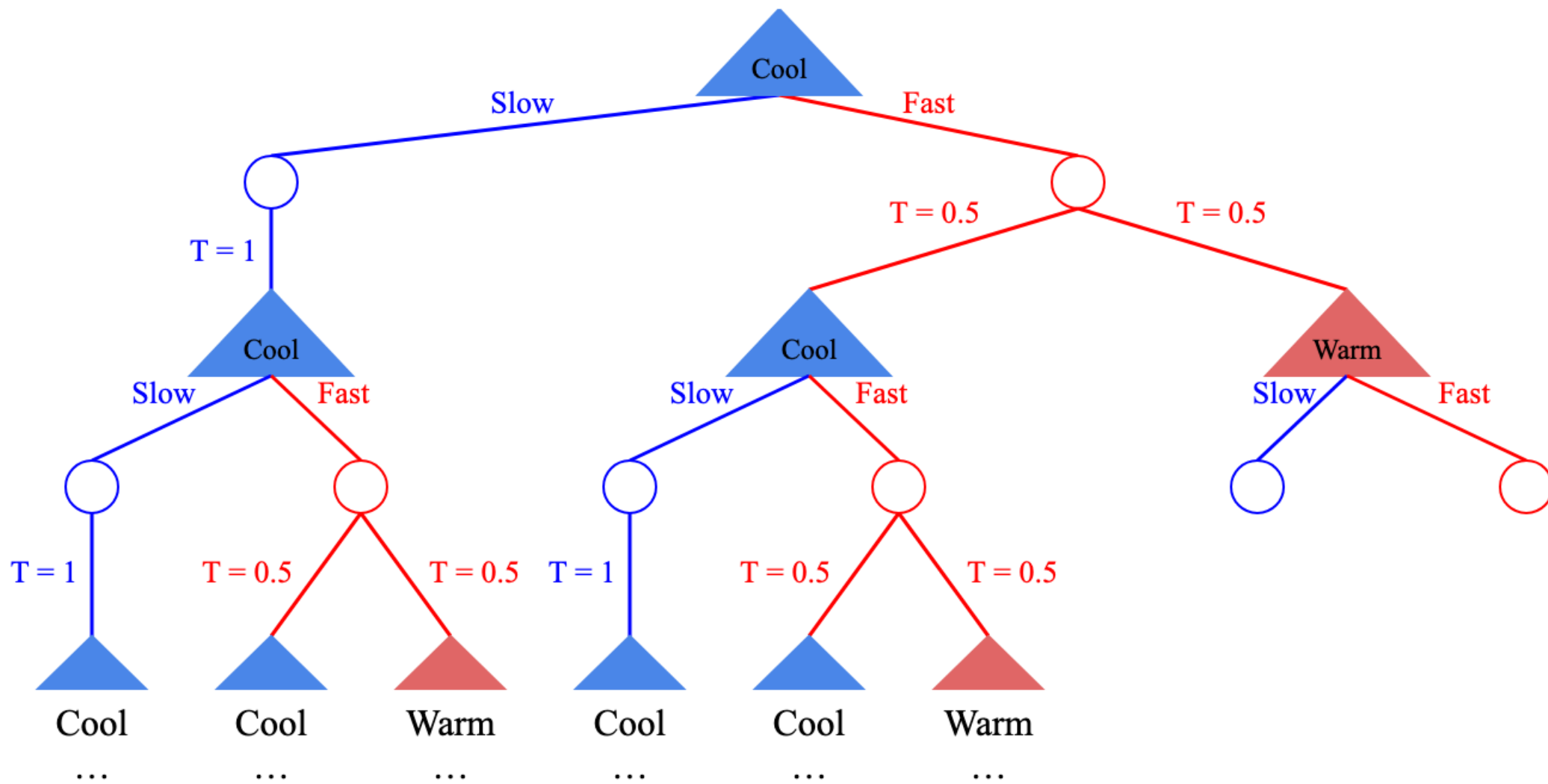
What Is the Optimal Action Here?

Slow: reward +1 · Fast: reward +2; overheating: -10 · T: transition probability



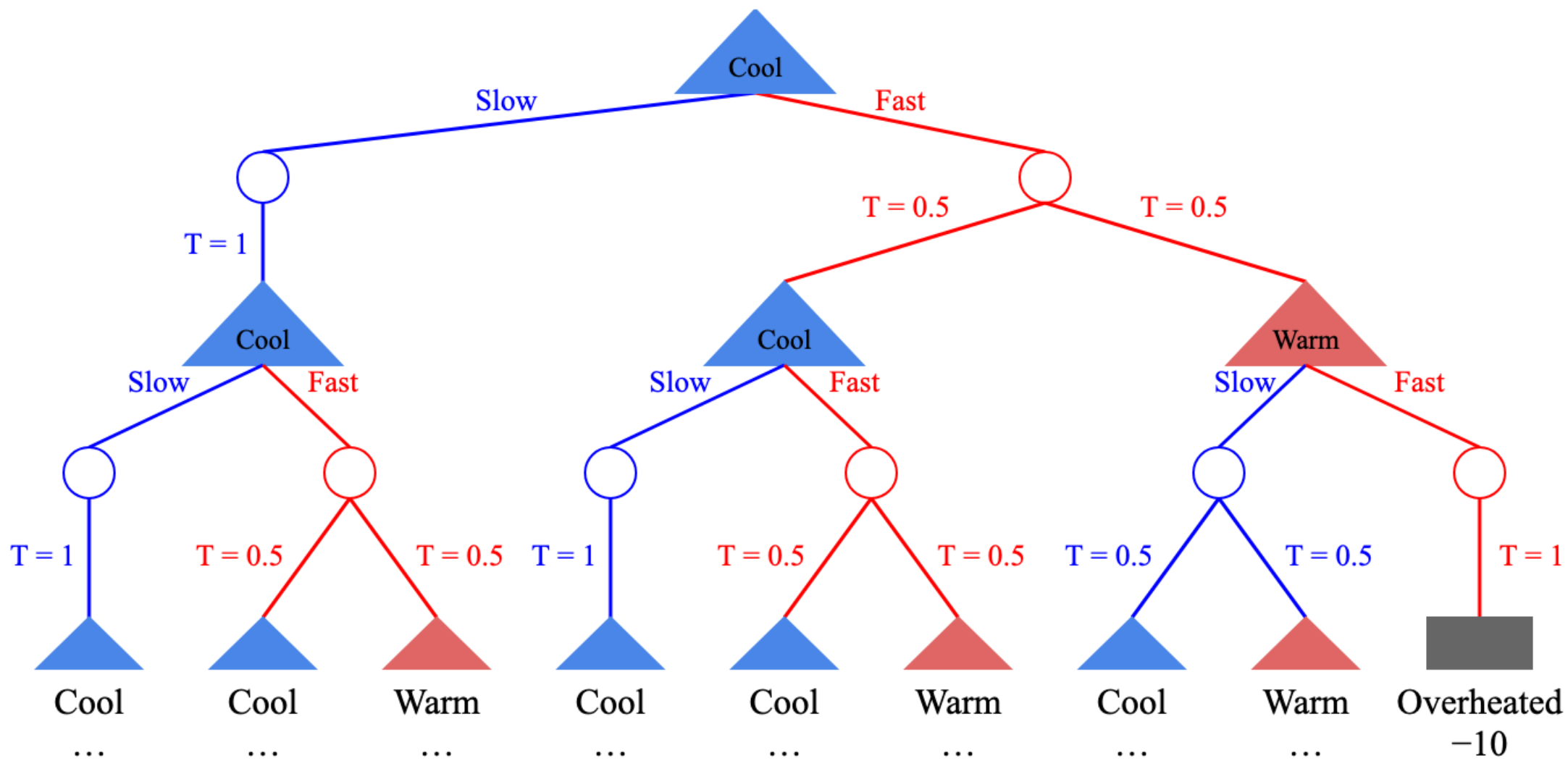
What Is the Optimal Action Here?

Slow: reward +1 · Fast: reward +2; overheating: -10 · T: transition probability



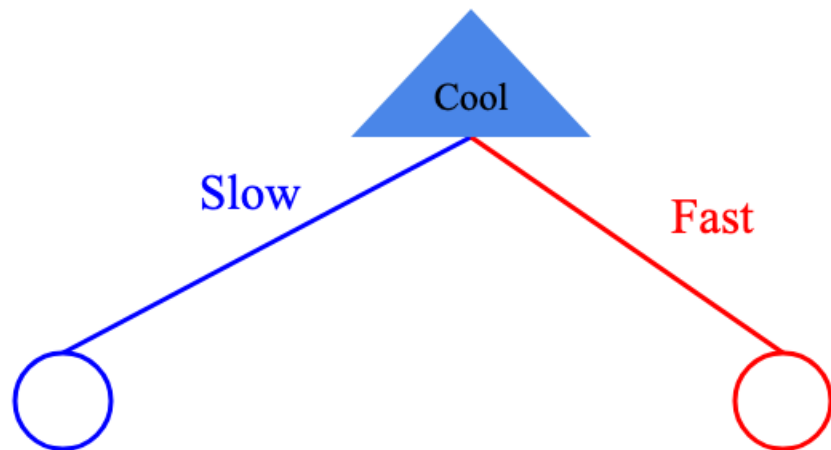
What Is the Optimal Action Here?

Slow: reward +1 · Fast: reward +2; overheating: -10 · T: transition probability



Choose an Action, Average the Outcomes

$T(s, a, s')$ is the transition probability

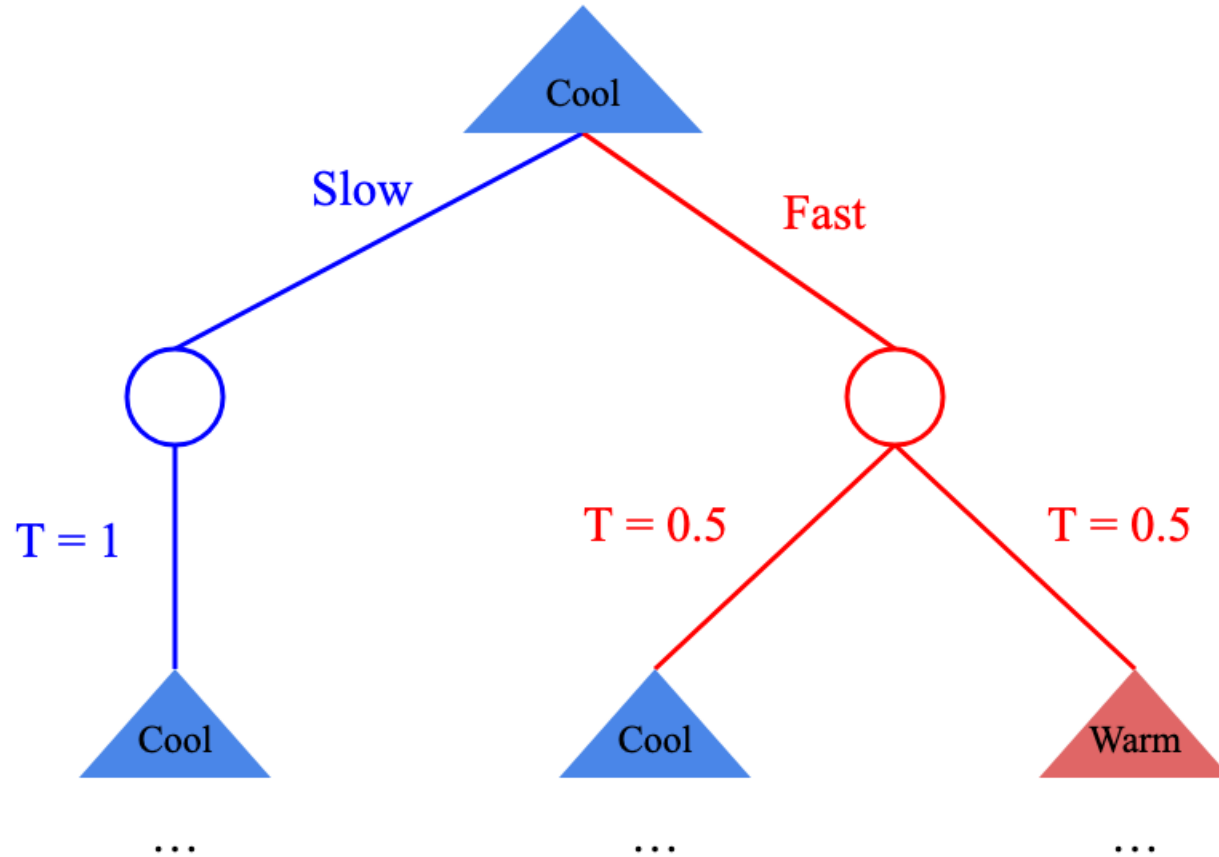


At a state: choose

Take the action with the
highest expected return

Choose an Action, Average the Outcomes

$T(s, a, s')$ is the transition probability



At a state: choose

Take the action with the highest expected return

After an action: average

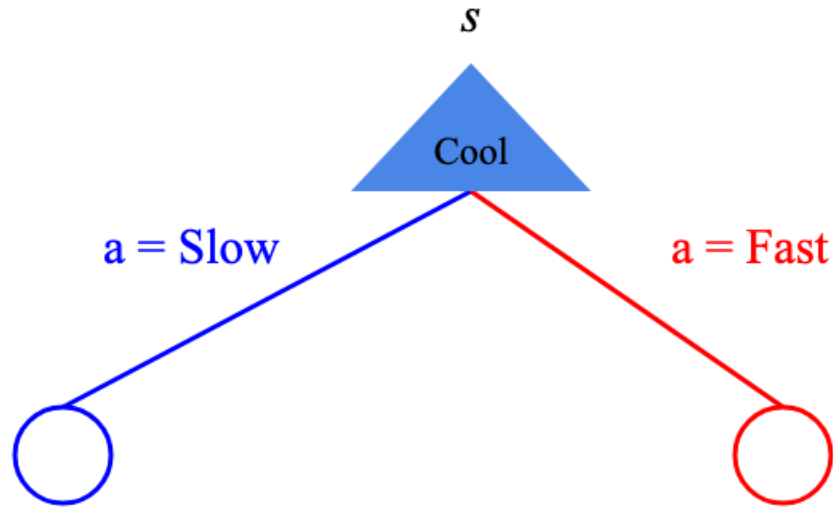
Weight each possible outcome by its probability

Fast: equal weights, 0.5 and 0.5

Choose the best action; average over chance outcomes

Q-State

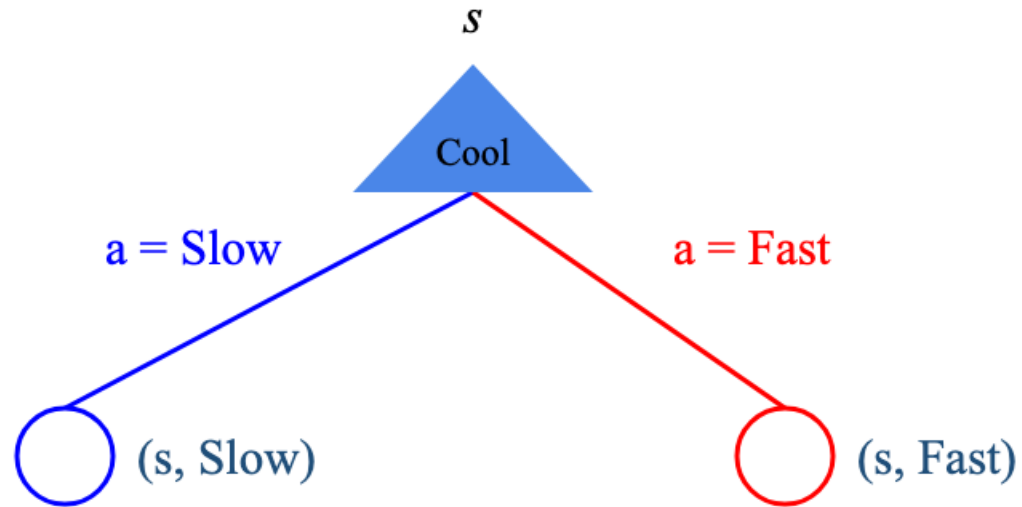
The agent has committed to the action but has not done it yet



s is the current state
 a is the chosen action

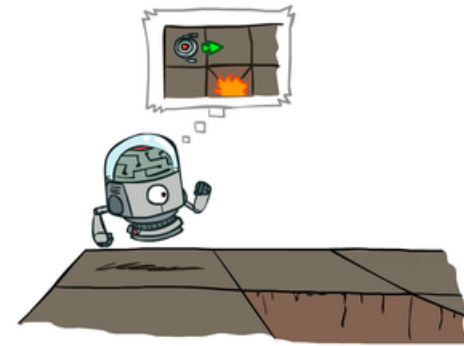
Q-State

The agent has committed to the action but has not done it yet



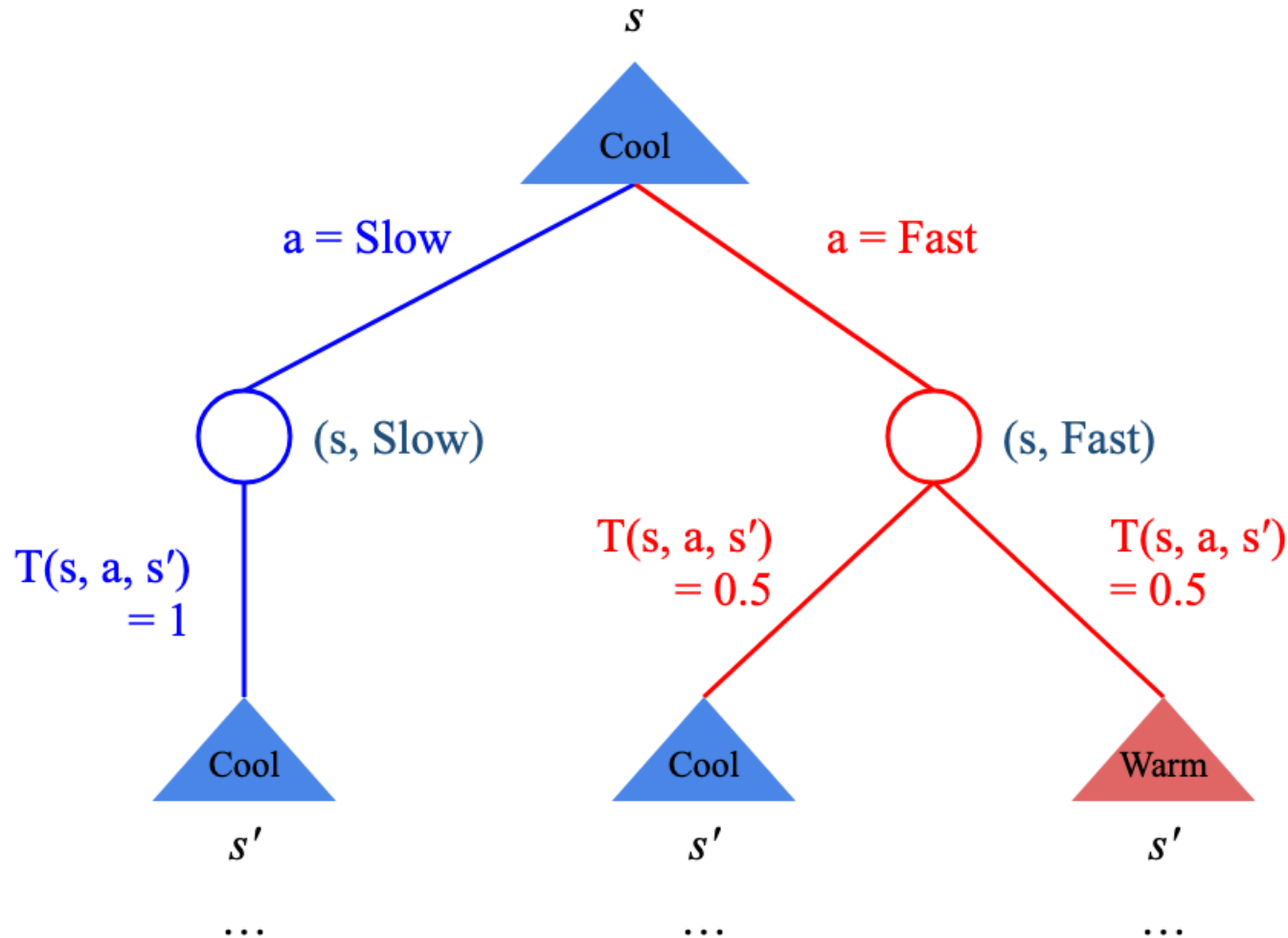
s is the current state
 a is the chosen action

(s, a) is a Q-state



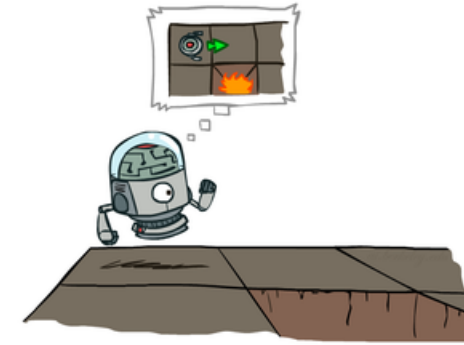
Q-State

The agent has committed to the action but has not done it yet



s is the current state
 a is the chosen action

(s, a) is a Q-state



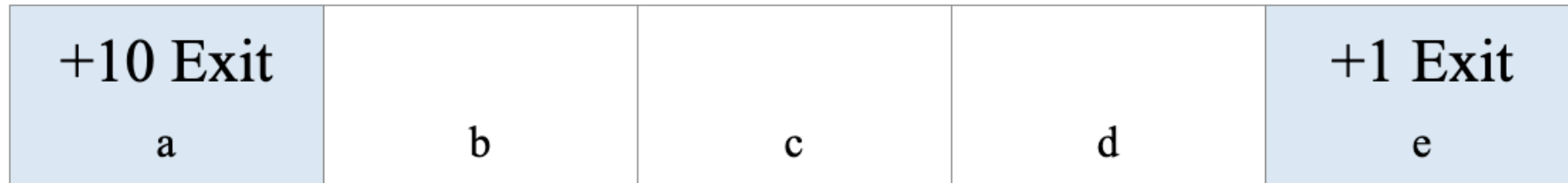
s' is the next state

$T(s, a, s')$

Probability of reaching s' after taking a in s

A Corridor MDP

$T(s, a, s')$: probability of next state s' after action a in current state s



East and West

0.8: move to the neighbour

Failed Movement

0.2: stay in the same square

Movement Reward

0 for every move

Exit at a or e

+10 or +1, then the episode ends

Where Can We Land?

TASK

Choose East from c

+10 Exit				+1 Exit
a	b	c	d	e

Transition

Probability

$T(c, \text{East}, d)$

$T(c, \text{East}, c)$

Any other state

Where Can We Land?

Choose East from c

+10 Exit				+1 Exit
a	b	c	d	e

Transition

Probability

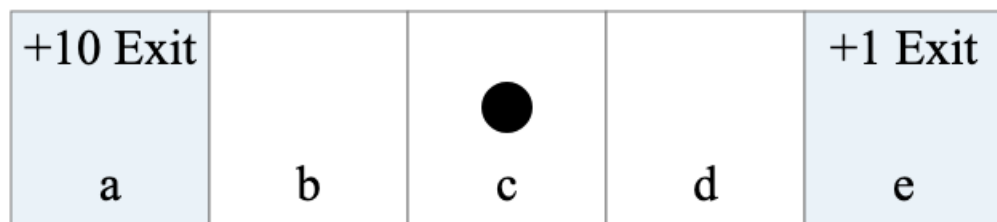
$T(c, \text{East}, d)$

$T(c, \text{East}, c)$

Any other state

Where Can We Land?

Choose East from c



Transition

Probability

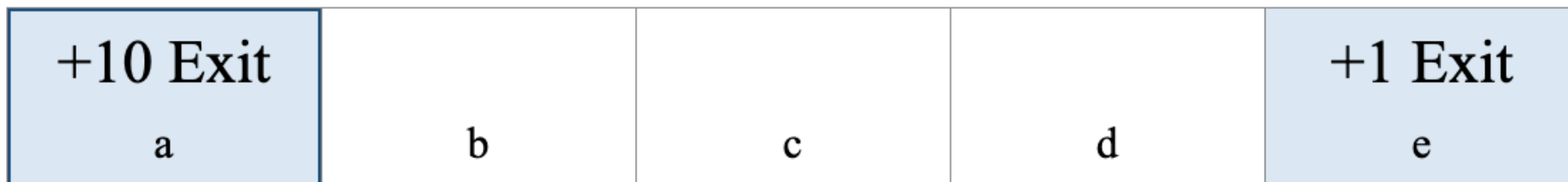
$T(c, \text{East}, d)$

$T(c, \text{East}, c)$

Any other state

Probability and Reward

TASK



At a, compare East with Exit

State s	Action a	Next State s'	Probability $T(s, a, s')$	Reward $R(s, a, s')$
a	East			
a	East			
a	Exit			

Probability and Reward

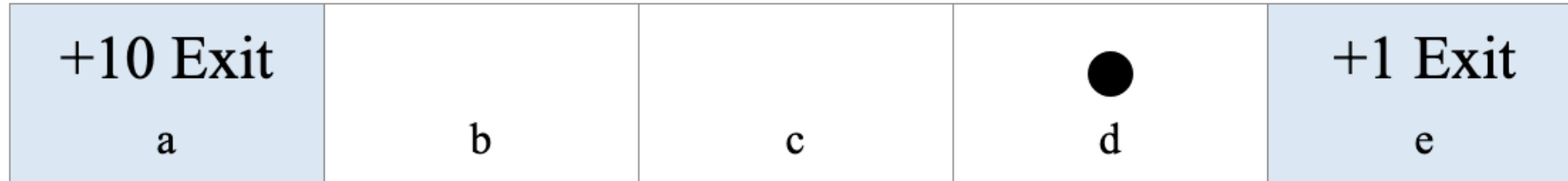
+10 Exit a	b	c	d	+1 Exit e
---------------	---	---	---	--------------

At a, compare East with Exit

State s	Action a	Next State s'	Probability $T(s, a, s')$	Reward $R(s, a, s')$
a	East			
a	East			
a	Exit			

From d, West or East?

Same model: 0.8 move, 0.2 stay · Movement reward 0



West toward +10

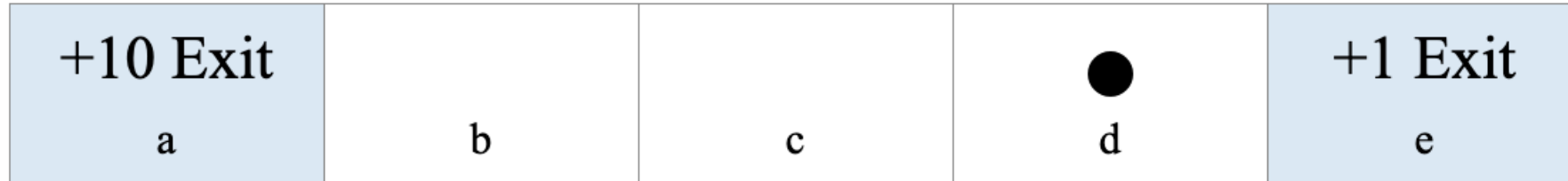
Three successful moves, then Exit

East toward +1

One successful move, then Exit

From d, West or East?

Same model: 0.8 move, 0.2 stay · Movement reward 0



West toward +10

Three successful moves, then Exit

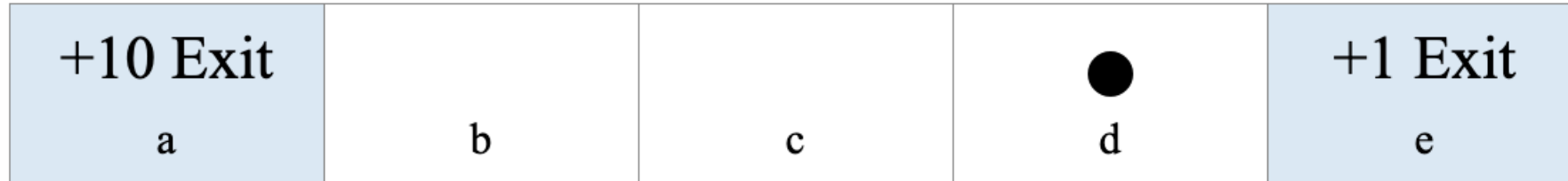
East toward +1

One successful move, then Exit

A larger reward later, or a smaller reward sooner?

From d, West or East?

Same model: 0.8 move, 0.2 stay · Movement reward 0



West toward +10

Three successful moves, then Exit

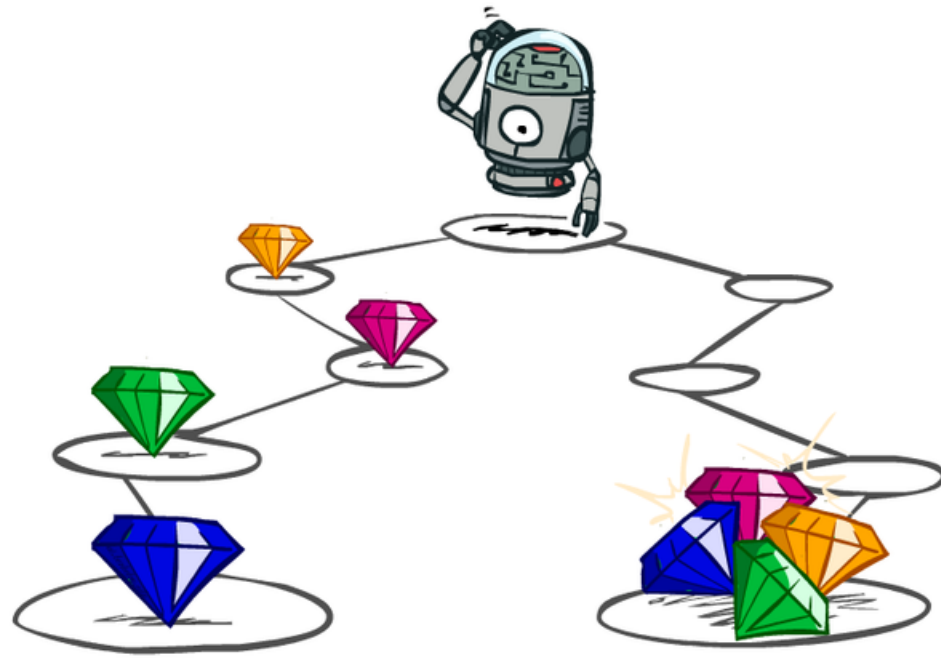
East toward +1

One successful move, then Exit

A larger reward later, or a smaller reward sooner?

How much should waiting reduce a reward's value?

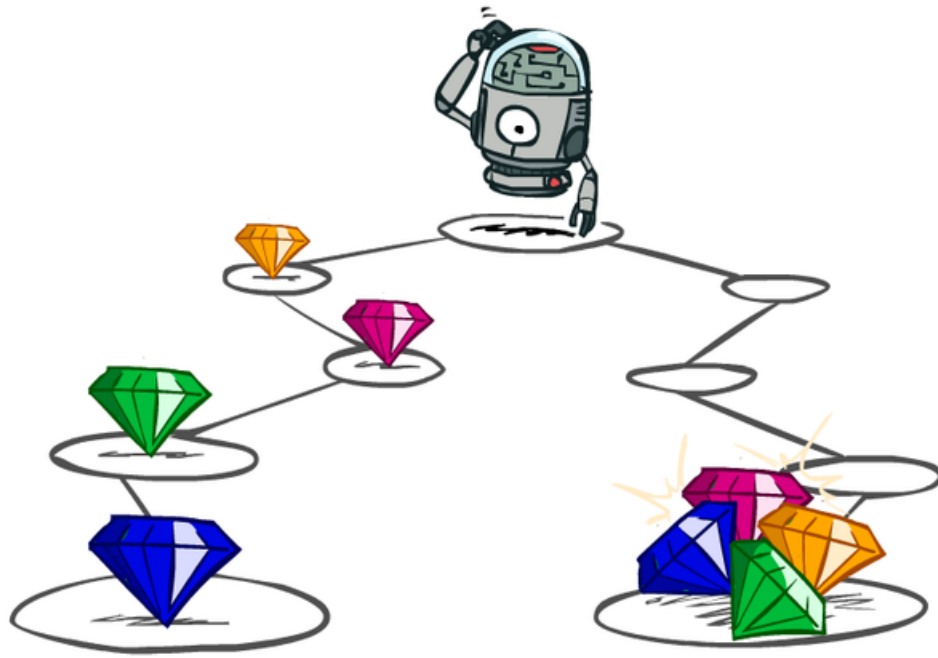
Comparing Reward Sequences



More reward?

$[1, 2, 2]$ or $[2, 3, 4]$

Comparing Reward Sequences



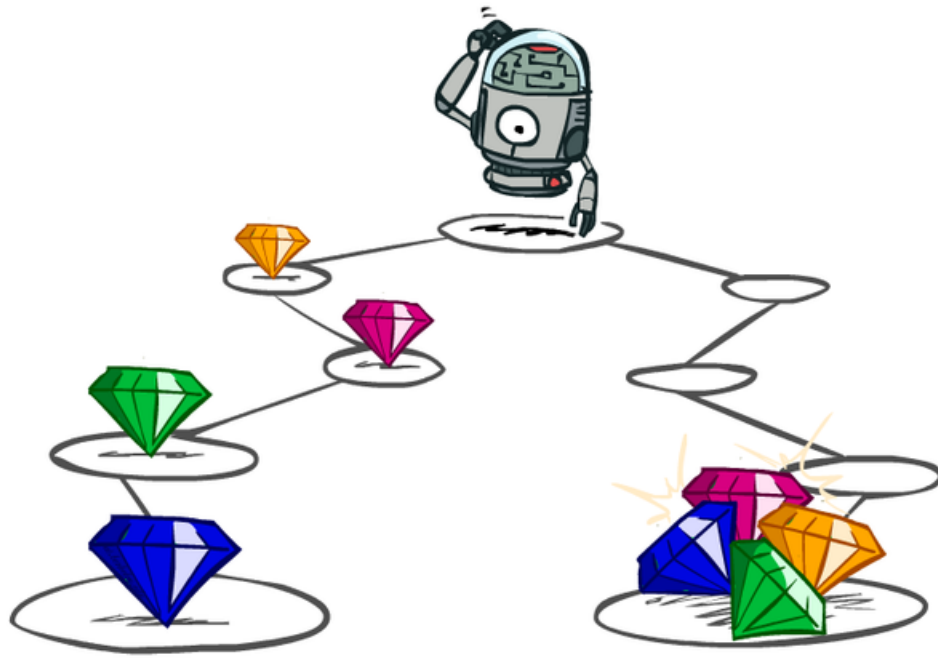
More reward?

$[1, 2, 2]$ or $[2, 3, 4]$

Earlier reward?

$[0, 0, 1]$ or $[1, 0, 0]$

Comparing Reward Sequences



More reward?

$[1, 2, 2]$ or $[2, 3, 4]$

Earlier reward?

$[0, 0, 1]$ or $[1, 0, 0]$

How should we compare whole reward sequences?

Discounting

Prefer rewards now to the same rewards later



1

Now

Discounting

Prefer rewards now to the same rewards later



1

Now



γ

One step later

Discounting

Prefer rewards now to the same rewards later



1

Now



γ

One step later



γ^2

Two steps later

Discounting

Prefer rewards now to the same rewards later



1

Now



γ

One step later



γ^2

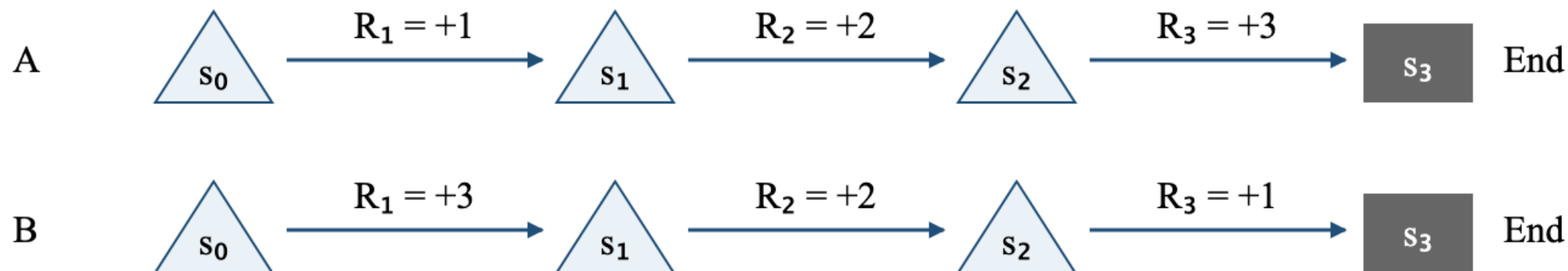
Two steps later

Each extra step multiplies the reward's value by γ

$$0 < \gamma < 1$$

Which State Sequence Is Better?

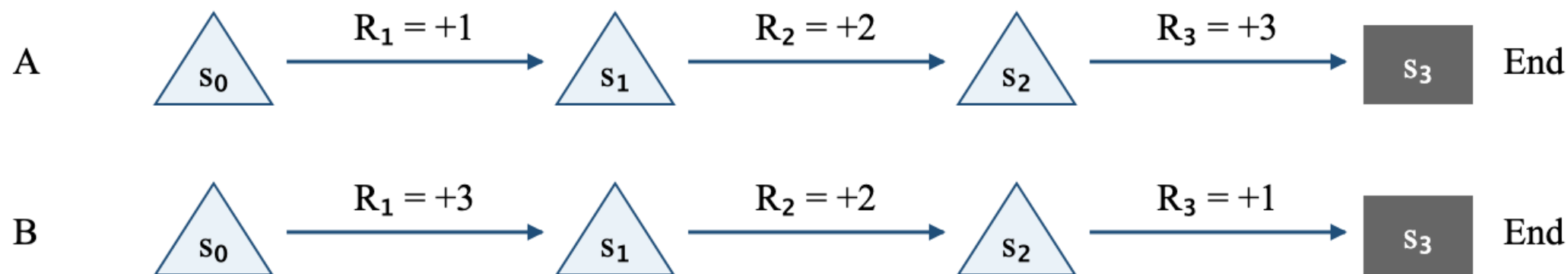
TASK



Return from the start: $G_0 = R_1 + \gamma R_2 + \gamma^2 R_3$

Calculate both returns for $\gamma = 1$ and $\gamma = 0.5$

Which State Sequence Is Better?



Return from the start: $G_0 = R_1 + \gamma R_2 + \gamma^2 R_3$

Without Discounting ($\gamma = 1$)

With $\gamma = 0.5$

A

B

Discounted Return

$0 \leq \gamma \leq 1$ for finite reward sequences

Example

1

2

3

Discounted Return

$0 \leq \gamma \leq 1$ for finite reward sequences

Example

1

2

3

Reward

R_{t+1}

R_{t+2}

R_{t+3}

Discounted Return

$0 \leq \gamma \leq 1$ for finite reward sequences

Example

1

2

3

Reward

R_{t+1}

R_{t+2}

R_{t+3}

Weight

1

γ

γ^2

Discounted Return

$0 \leq \gamma \leq 1$ for finite reward sequences

Example

1

2

3

Reward

R_{t+1}

R_{t+2}

R_{t+3}

Weight

1

γ

γ^2

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

Discounted Return

$0 \leq \gamma \leq 1$ for finite reward sequences

Example

1

2

3

Reward

R_{t+1}

R_{t+2}

R_{t+3}

Weight

1

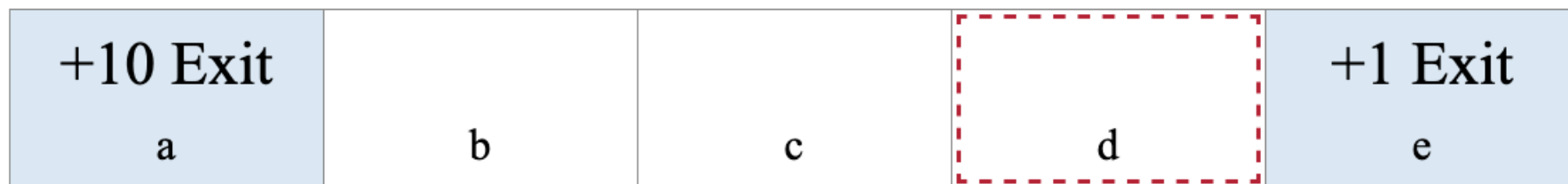
γ

γ^2

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

A smaller γ gives less weight to later rewards

Discounting and the Preferred Route



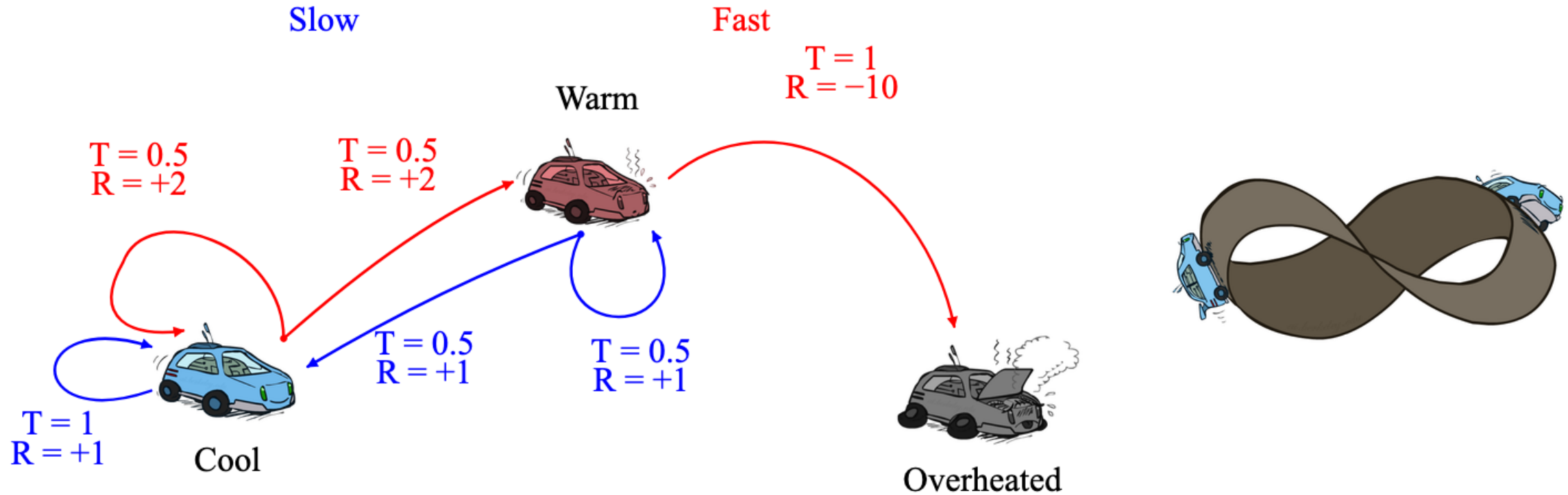
Start at d · Movement reward 0 · 0.8 move, 0.2 stay

Discount	West to +10	East to +1	Preferred
0.1	0.005	0.082	East
0.9	6.769	0.878	West

Expected discounted returns include failed-move delays and a separate Exit action

Infinite Reward Sequences

At $\gamma = 0.9$: Fast when Cool, Slow when Warm



The optimal policy keeps driving without overheating

Slow earns +1; Fast from Cool earns +2

Infinite Reward Sequences

The agent earns +1 on every move, forever

No discount

$$1 + 1 + 1 + \dots \rightarrow \infty$$

With $\gamma = 0.5$

$$1 + \frac{1}{2} + \frac{1}{4} + \dots = 2$$

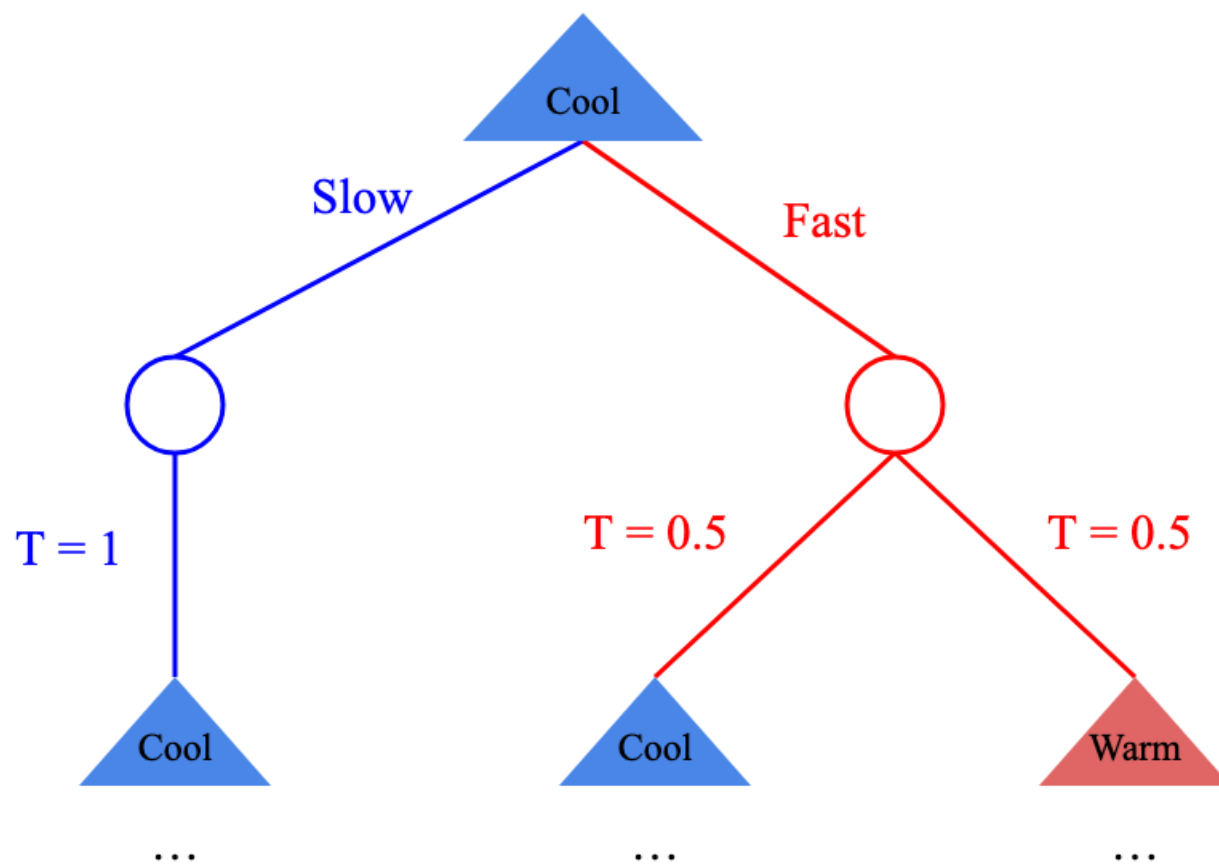
An endless reward sequence can have a finite discounted return

Every reward has magnitude at most $R_{\max}$; $0 \leq \gamma < 1$

$$|G_t| \leq \sum_{k=0}^{\infty} \gamma^k |R_{t+k+1}| \leq R_{\max} (1 + \gamma + \gamma^2 + \dots) = \frac{R_{\max}}{1 - \gamma}$$

State Values and Action Values

s is the current state; a is the first action

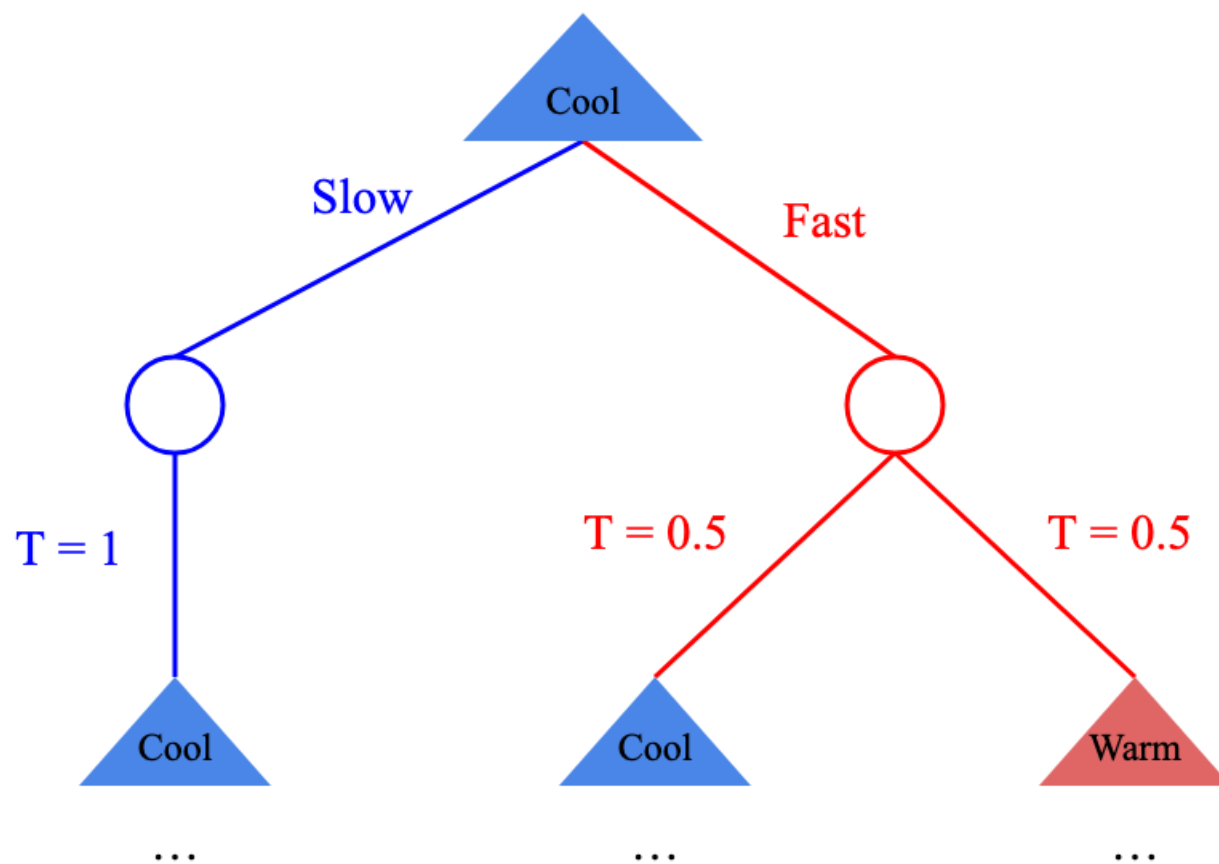


$$V^*(s)$$

Best expected return
starting in s

State Values and Action Values

s is the current state; a is the first action



$$V^*(s)$$

Best expected return
starting in s

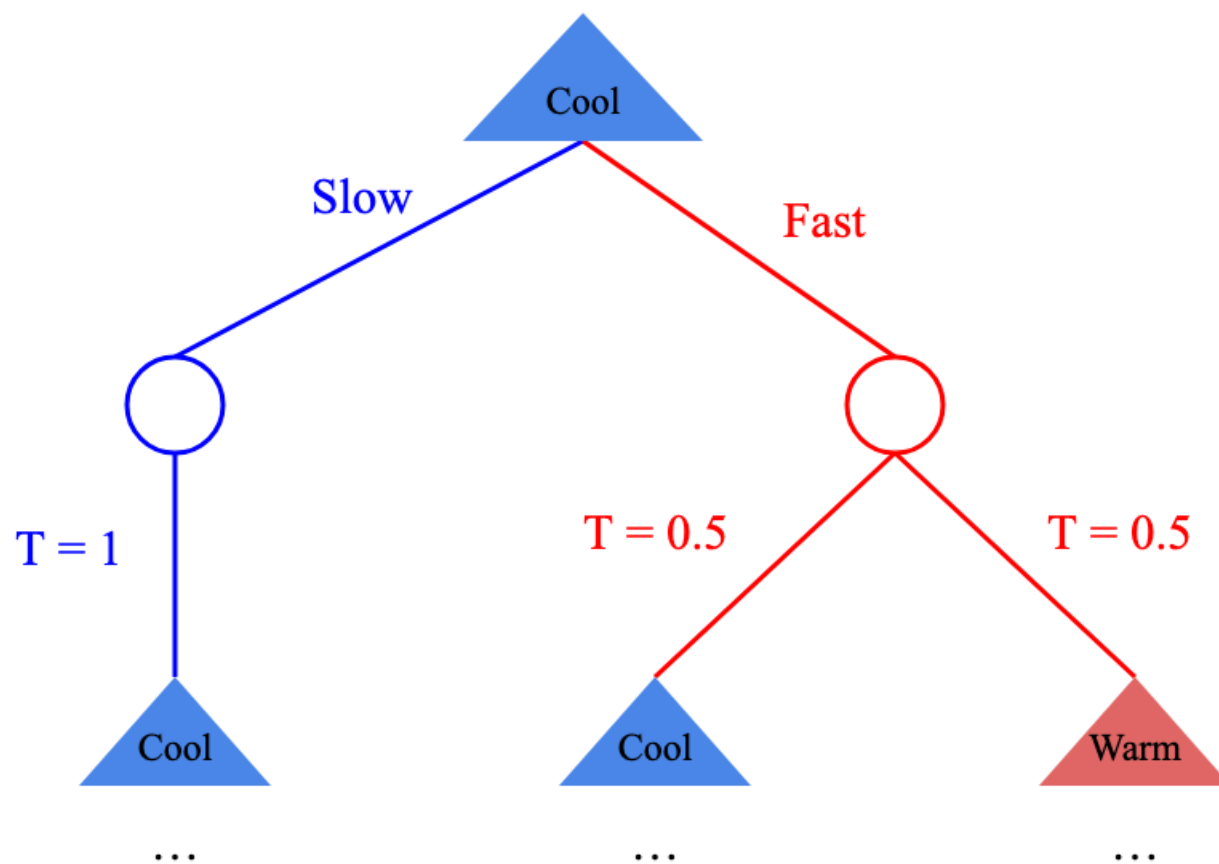
$$Q^*(s, a)$$

Take a in s , then act optimally

Includes the reward from the first action

State Values and Action Values

s is the current state; a is the first action



$$V^*(s)$$

Best expected return
starting in s

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

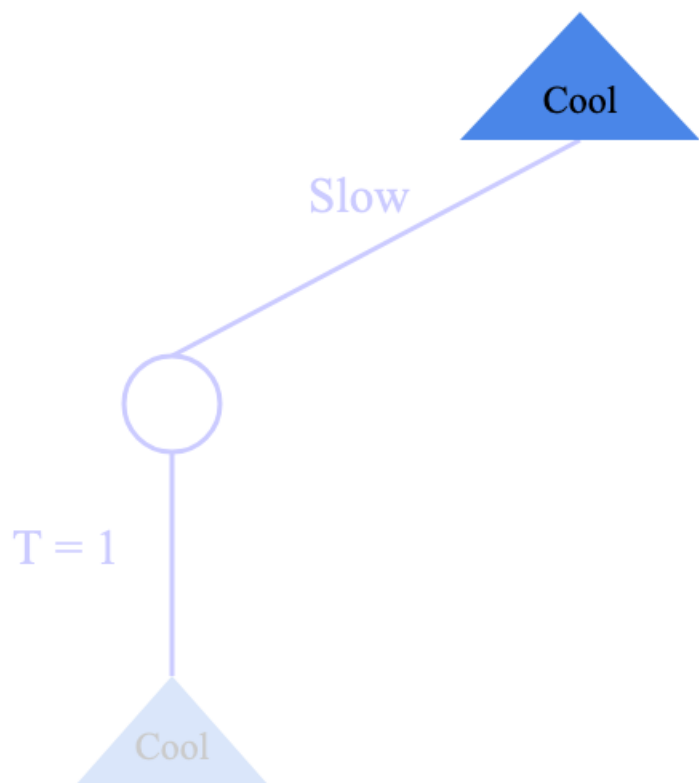
Take a in s , then act optimally

Includes the reward from the first action

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

A Backup Combines Reward and Future Value

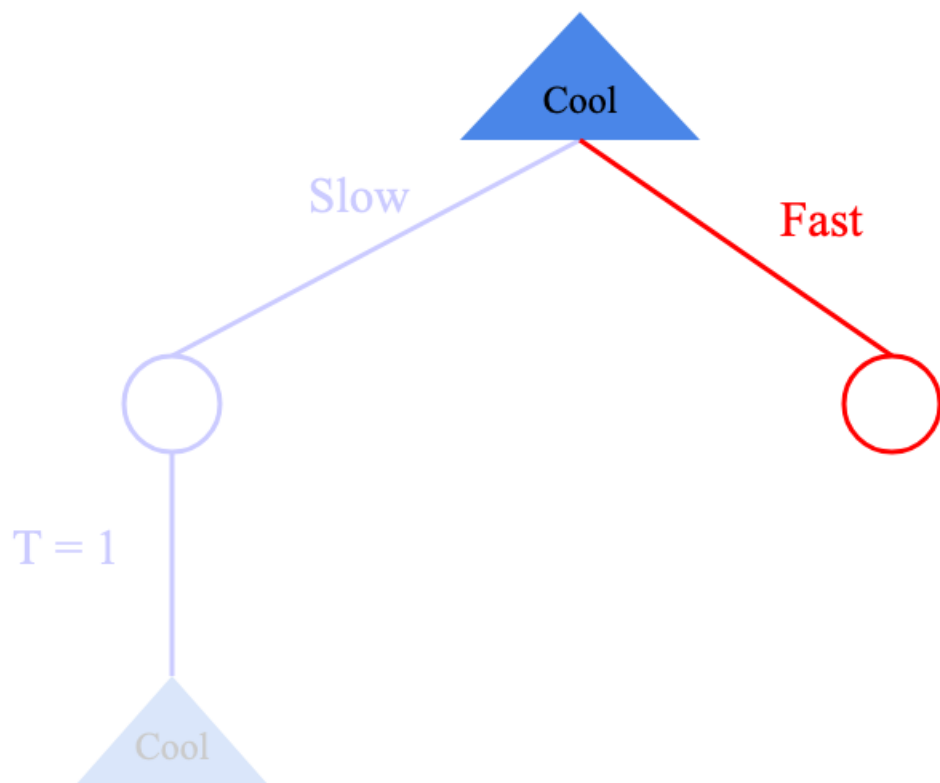
Discount 0.9 · Supplied optimal continuation values



Choose Fast at Cool

A Backup Combines Reward and Future Value

Discount $0.9 \cdot$ Supplied optimal continuation values



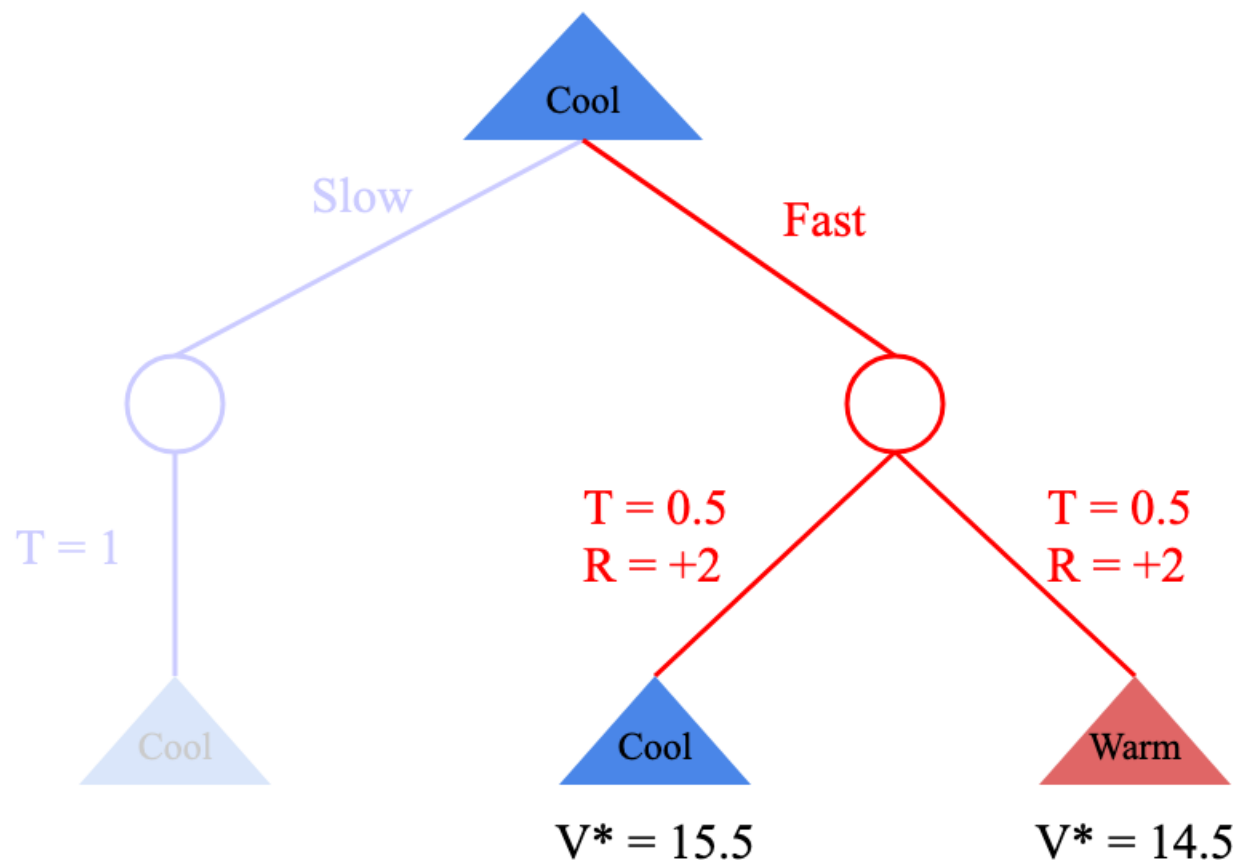
Choose Fast at Cool

Reward from This Action

2

A Backup Combines Reward and Future Value

Discount 0.9 · Supplied optimal continuation values



Choose Fast at Cool

Reward from This Action

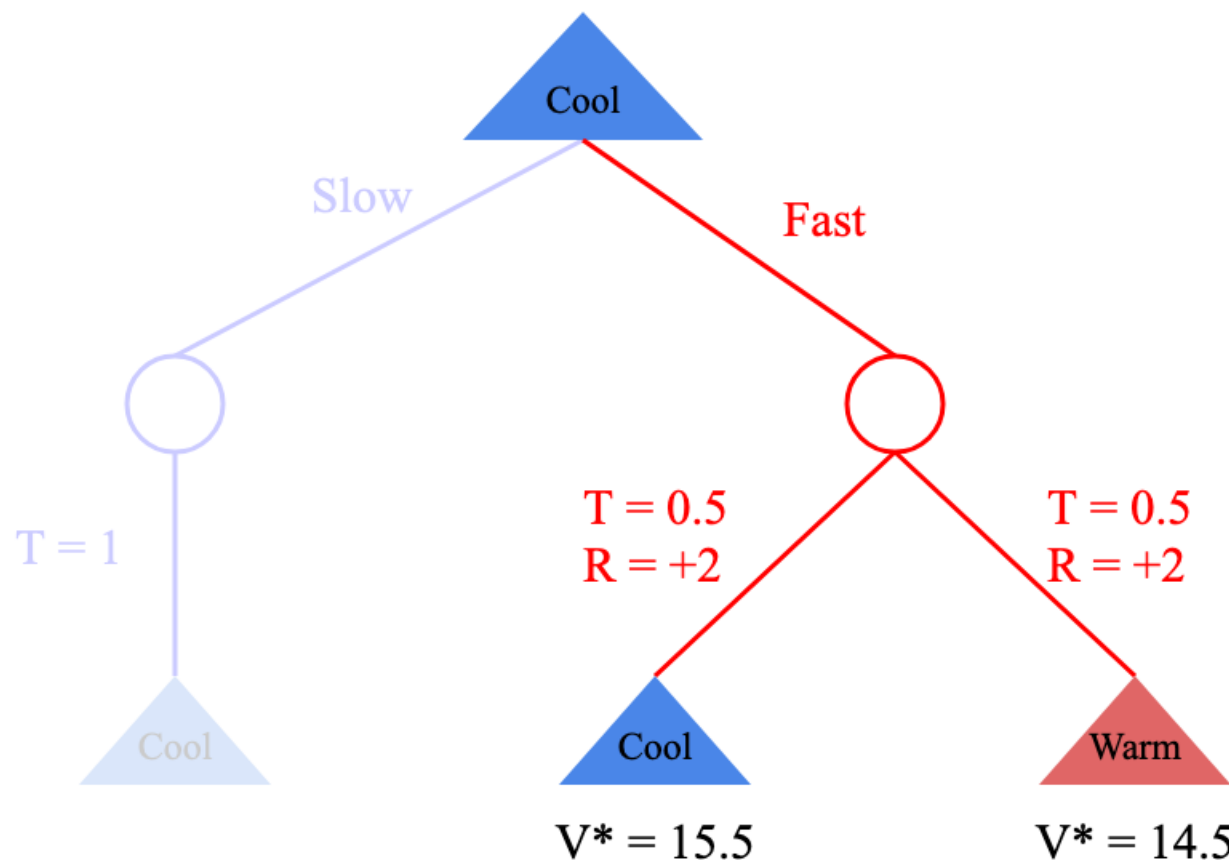
2

Average Future Value

$$0.5(15.5) + 0.5(14.5) = 15$$

A Backup Combines Reward and Future Value

Discount 0.9 · Supplied optimal continuation values



Choose Fast at Cool

Reward from This Action

2

Average Future Value

$$0.5(15.5) + 0.5(14.5) = 15$$

$$2 + 0.9(15) = 15.5$$

$$Q^*(\text{Cool}, \text{Fast}) = 0.5[2 + 0.9V^*(\text{Cool})] + 0.5[2 + 0.9V^*(\text{Warm})]$$

The Bellman Optimality Equations

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Average over outcomes, then maximise over actions

The Bellman Optimality Equations

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

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

Average over outcomes, then maximise over actions

The Bellman Optimality Equations

$$Q^*(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

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

$$\pi^*(s) \in \arg \max_a Q^*(s, a)$$

Average over outcomes, then maximise over actions

Bellman in Gridworld

Movement reward -0.1 · Discount 0.9 · Initial Classic configuration

		x			
		1	2	3	4
y	3	0.31	0.51	0.72	+1 1.00
	2	0.15		0.36	-1 -1.00
	1	0.01	0.01	0.15	-0.09

Choose East at (3, 3)

Next State	Probability	Value
(4, 3)	0.80	1.00
(3, 3)	0.10	0.72
(3, 2)	0.10	0.36

$$Q^*((3, 3), \text{East}) \approx 0.72$$

Each term: probability $\times$ (transition reward + discounted next-state value)

Value Iteration

$V_k(s)$ is the best expected return with k actions remaining

Value Iteration

$V_k(s)$ is the best expected return with k actions remaining

$$V_0(s) = 0$$

Value Iteration

$V_k(s)$ is the best expected return with k actions remaining

$$V_0(s) = 0$$

$$V_{k+1}(s) = \max_{a \in A(s)} \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_k(s')]$$

One action now, then k actions remain; Exit also counts as an action

Value Iteration in Gridworld

Movement reward 0.00 · Discount 0.90 · Two-action snapshot

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_k(s')]$$

		x			
		1	2	3	4
y	3	0.00	0.00	0.72	+1 1.00
	2	0.00		0.00	-1 -1.00
	1	0.00	0.00	0.00	0.00

$$V_0(s) = 0$$

One action: Exit gives +1 or -1

Two actions: move, then Exit

$$V_2((3, 3)) = 0.8[0 + 0.9(1)] + 0.2[0 + 0.9(0)] = 0.72$$

Every update reads the previous sweep

Q-Value Iteration

$Q_k(s, a)$ is the best expected return with k actions remaining

Take action a first, then act optimally

Q-Value Iteration

$Q_k(s, a)$ is the best expected return with k actions remaining

Take action a first, then act optimally

$$Q_0(s, a) = 0$$

Q-Value Iteration

$Q_k(s, a)$ is the best expected return with k actions remaining

Take action a first, then act optimally

$$Q_0(s, a) = 0$$

$$Q_{k+1}(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_k(s', a')]$$

Q-Value Iteration

$Q_k(s, a)$ is the best expected return with k actions remaining

Take action a first, then act optimally

$$Q_0(s, a) = 0$$

$$Q_{k+1}(s, a) = \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma \max_{a'} Q_k(s', a')]$$

$$V_k(s) = \max_a Q_k(s, a)$$

Q-Value Iteration in Gridworld

Movement reward 0.00 · Discount 0.90 · Two-action snapshot

		x			
		1	2	3	4
y	3	0.00	0.00	0.72	+1 1.00
	2	0.00		0.00	-1 -1.00
	1	0.00	0.00	0.00	0.00

Two actions remaining at (3, 3)

First Action	Q_2
North	0.09
East	0.72
South	0.09
West	0.00

$$V_2((3, 3)) = \max_a Q_2((3, 3), a) = 0.72$$

Each row keeps one action value; the maximum gives the state value

Convergence

Finite states and actions, bounded rewards, $0 \leq \gamma < 1$

$$V_k \rightarrow V^*, \quad Q_k \rightarrow Q^*$$

Convergence

Finite states and actions, bounded rewards, $0 \leq \gamma < 1$

$$V_k \rightarrow V^*, \quad Q_k \rightarrow Q^*$$

Stop when the largest change is below a tolerance

$$\max_s |V_{k+1}(s) - V_k(s)| < \varepsilon$$

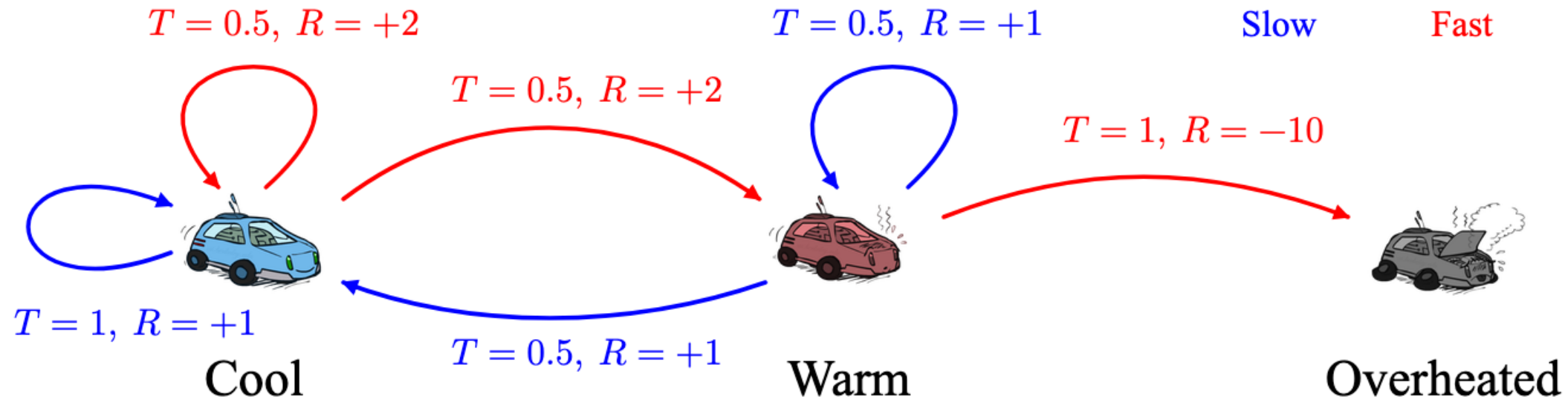
$$\max_{s, a} |Q_{k+1}(s, a) - Q_k(s, a)| < \varepsilon$$

Quiz: Value Iteration

TASK

$\gamma = 1 \cdot$ Find V_1 and V_2

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_k(s')]$$

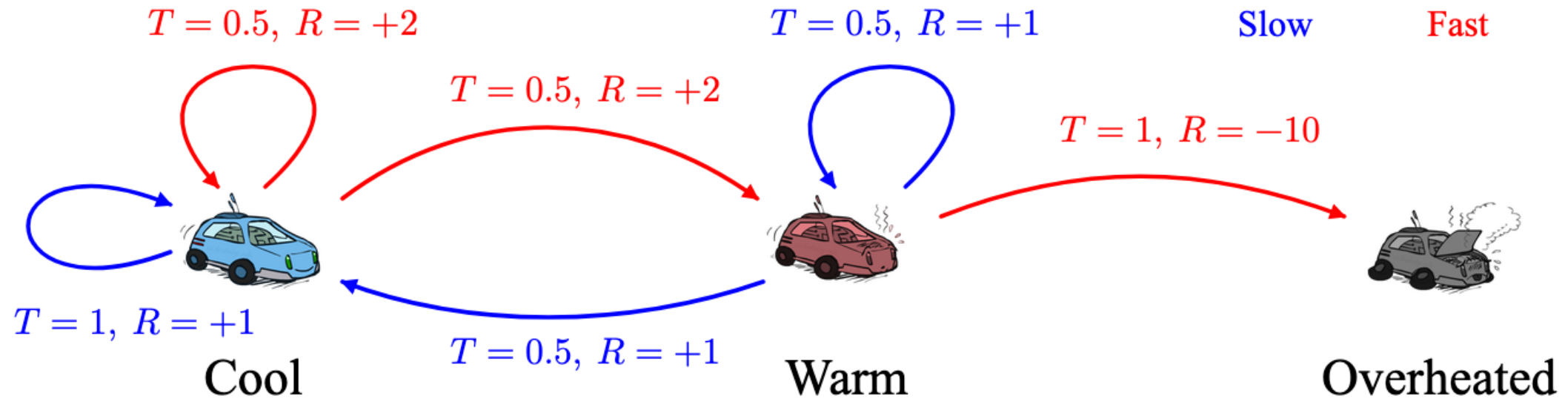


Round	Cool	Warm	Overheated
V_0	0	0	0
V_1			
V_2			

Quiz: Value Iteration

$\gamma = 1 \cdot$ Find V_1 and V_2

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_k(s')]$$



Round	Cool	Warm	Overheated
V_0	0	0	0
V_1			
V_2			

Evaluate a Fixed Policy

Given this policy, how do we determine its values?

→	→	↑	Exit
→		→	Exit
→	→	→	→

Evaluate a Fixed Policy

Given this policy, how do we determine its values?

→	→	↑	Exit
→		→	Exit
→	→	→	→

$$V_{k+1}^{\pi}(s) = \sum_{s'} T(s, \pi(s), s') [R(s, \pi(s), s') + \gamma V_k^{\pi}(s')]$$

Evaluate a Fixed Policy

Movement reward $0.00 \cdot \text{Discount } 0.90 \cdot 80\% \text{ intended direction} \cdot \text{Exit rewards } +1 / -1$

Given this policy, how do we determine its values?

→ 0.31	→ 0.39	↑ 0.45	Exit 1.00
→ 0.00		→ -0.72	Exit -1.00
→ -0.31	→ -0.39	→ -0.45	→ -0.47

$$V_{k+1}^{\pi}(s) = \sum_{s'} T(s, \pi(s), s') [R(s, \pi(s), s') + \gamma V_k^{\pi}(s')]$$

Choose Actions from Values

Movement reward 0.00 · Discount 0.90 · 80% intended direction · Exit rewards +1 / -1

Given these optimal values V^* , which actions should we choose?

0.64	0.74	0.85	1.00
0.57		0.57	-1.00
0.49	0.43	0.48	0.28

Choose Actions from Values

Movement reward 0.00 · Discount 0.90 · 80% intended direction · Exit rewards +1 / -1

Given these optimal values V^* , which actions should we choose?

0.64	0.74	0.85	1.00
0.57		0.57	-1.00
0.49	0.43	0.48	0.28

$$\pi^*(s) \in \arg \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Choose Actions from Values

Movement reward 0.00 · Discount 0.90 · 80% intended direction · Exit rewards +1 / -1

Given these optimal values V^* , which actions should we choose?

→ 0.64	→ 0.74	→ 0.85	Exit 1.00
↑ 0.57		↑ 0.57	Exit -1.00
↑ 0.49	← 0.43	↑ 0.48	← 0.28

$$\pi^*(s) \in \arg \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Choose a Machine

Which machine would you choose for the next 100 plays?



Blue: \$1 every play

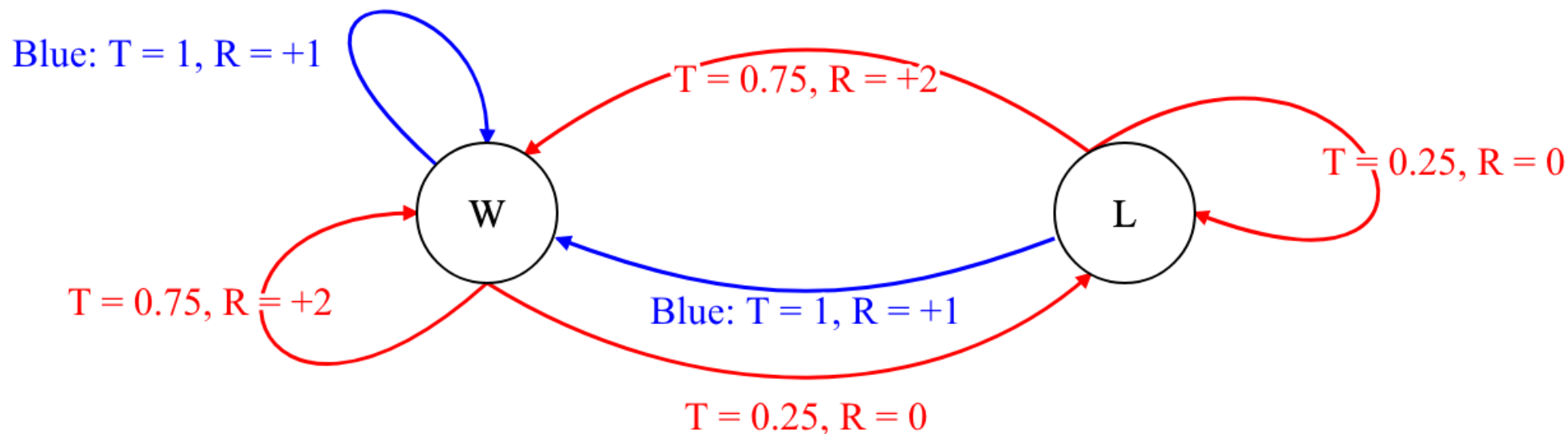


Red: \$2 with probability 0.75

\$0 with probability 0.25

The Machines as an MDP

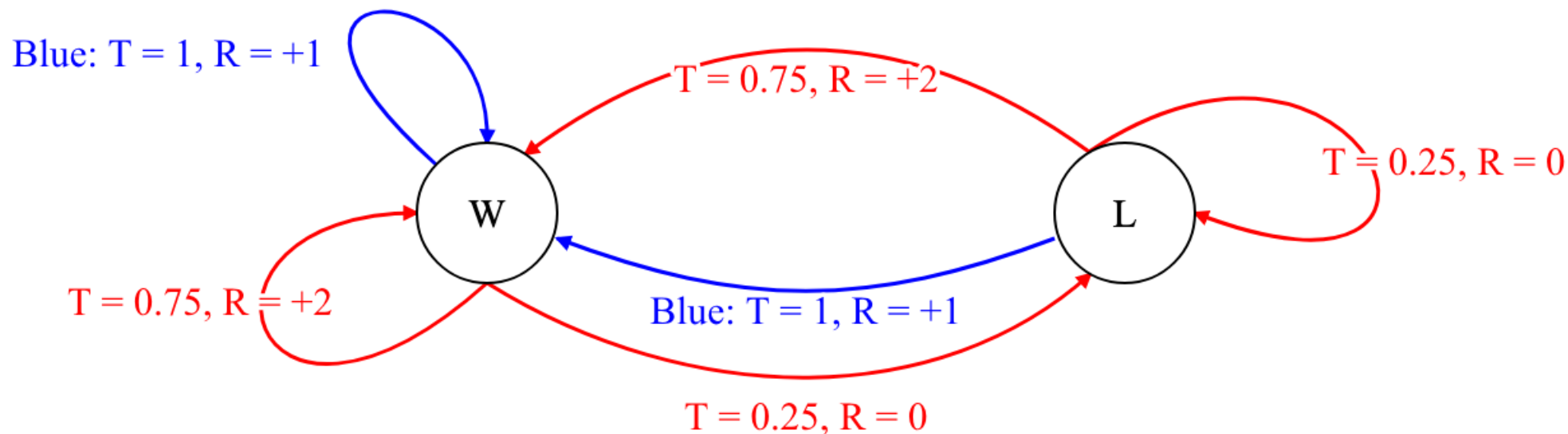
100 plays remain · $\gamma = 1$ · Compare two fixed policies



Always Blue or always Red, starting from either state

The Machines as an MDP

100 plays remain $\cdot \gamma = 1 \cdot$ Compare two fixed policies



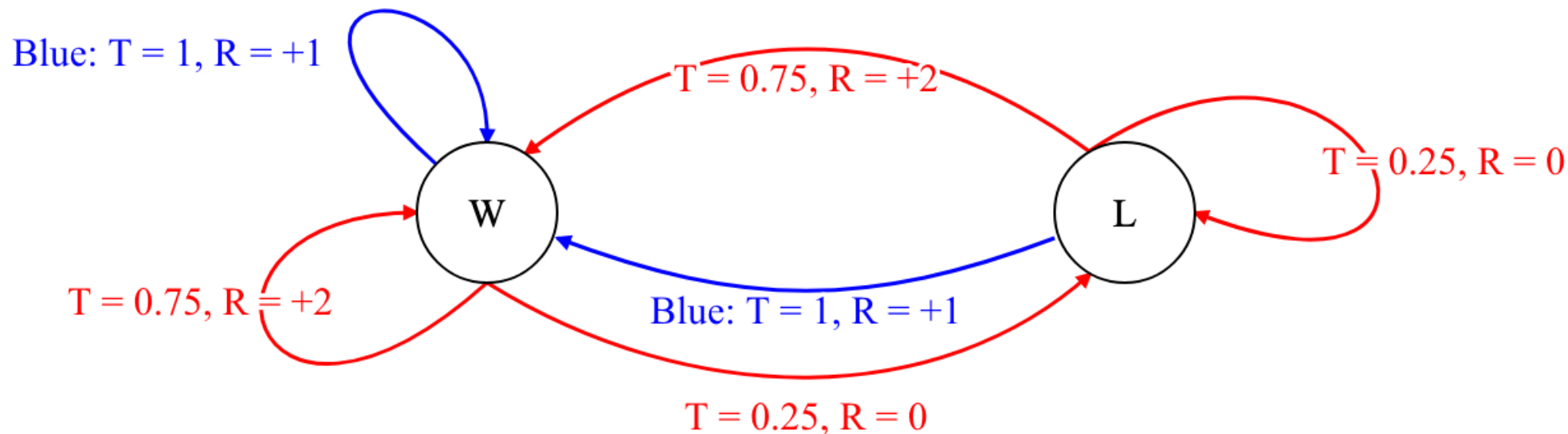
Always Blue or always Red, starting from either state

$$V_{100}^{\pi_{\text{Blue}}}(s) = 100(1) = 100$$

$$V_{100}^{\pi_{\text{Red}}}(s) = 100[0.75(2) + 0.25(0)] = 150$$

The Machines as an MDP

100 plays remain $\cdot \gamma = 1 \cdot$ Compare two fixed policies



Always Blue or always Red, starting from either state

$$V_{100}^{\pi_{\text{Blue}}}(s) = 100(1) = 100$$

$$V_{100}^{\pi_{\text{Red}}}(s) = 100[0.75(2) + 0.25(0)] = 150$$

$$V_{100}^*(W) = V_{100}^*(L) = 150 \rightarrow \text{Choose Red}$$

What If the Transition Probabilities Are Unknown?

In reinforcement learning, transitions, rewards, or both may be unknown

We can no longer solve this MDP
from the given information alone

What If the Transition Probabilities Are Unknown?

In reinforcement learning, transitions, rewards, or both may be unknown

We can no longer solve this MDP
from the given information alone

We must act, observe rewards,
and learn the missing probabilities

Learn by Playing

Choose a machine on each of 10 turns



Blue: \$1 every play



Red: \$2 or \$0, odds unknown

Illustrative 10-play record

Machine	Observed rewards	Sample average
Blue	1, 1, 1, 1	\$1.00
Red	0, 0, 0, 0, 0, 2	\$0.33

Each action earns a reward and may teach us something

Main Ideas in Reinforcement Learning

We needed to act to learn about the machines

Idea	What It Means

Main Ideas in Reinforcement Learning

We needed to act to learn about the machines

Idea	What It Means
Exploration	Try uncertain actions to gain information

Main Ideas in Reinforcement Learning

We needed to act to learn about the machines

Idea	What It Means
Exploration	Try uncertain actions to gain information
Exploitation	Use the best action according to current estimates

Main Ideas in Reinforcement Learning

We needed to act to learn about the machines

Idea	What It Means
Exploration	Try uncertain actions to gain information
Exploitation	Use the best action according to current estimates
Regret	Expected reward lost compared with choosing the best action

Main Ideas in Reinforcement Learning

We needed to act to learn about the machines

Idea	What It Means
Exploration	Try uncertain actions to gain information
Exploitation	Use the best action according to current estimates
Regret	Expected reward lost compared with choosing the best action
Sampling	Repeat observations because outcomes vary by chance

Main Ideas in Reinforcement Learning

We needed to act to learn about the machines

Idea	What It Means
Exploration	Try uncertain actions to gain information
Exploitation	Use the best action according to current estimates
Regret	Expected reward lost compared with choosing the best action
Sampling	Repeat observations because outcomes vary by chance

Learning can be harder than solving a known MDP