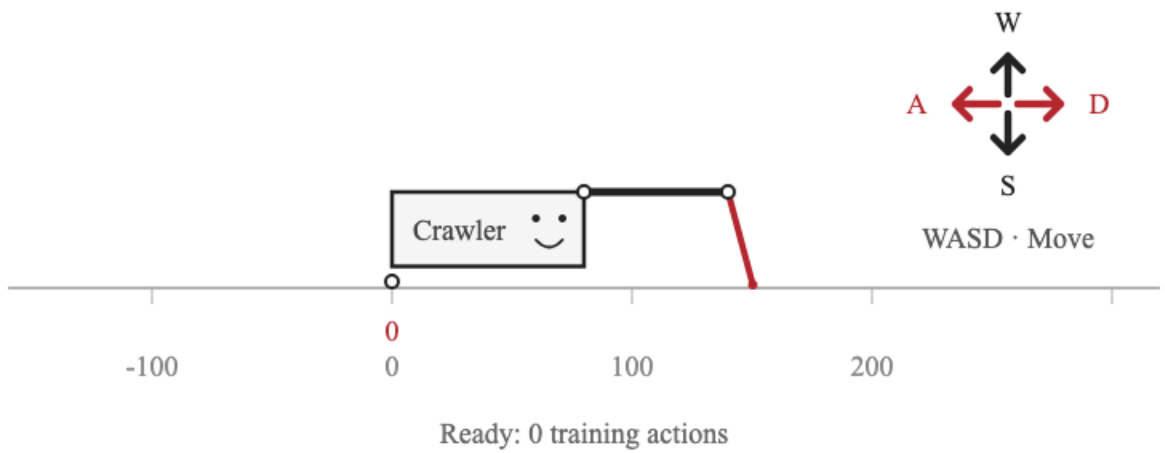
Please Sit With Your Group

L-7

Reinforcement Learning II

MAIB7002
Fundamentals of AI Models and Technologies

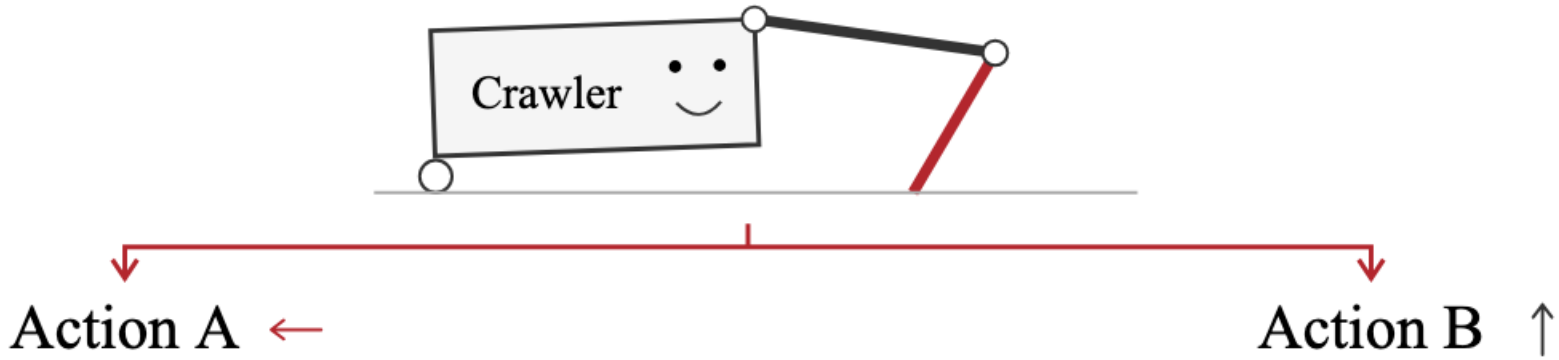
The Crawler Problem



Can experience improve a sequence of moves?

Immediate Rewards

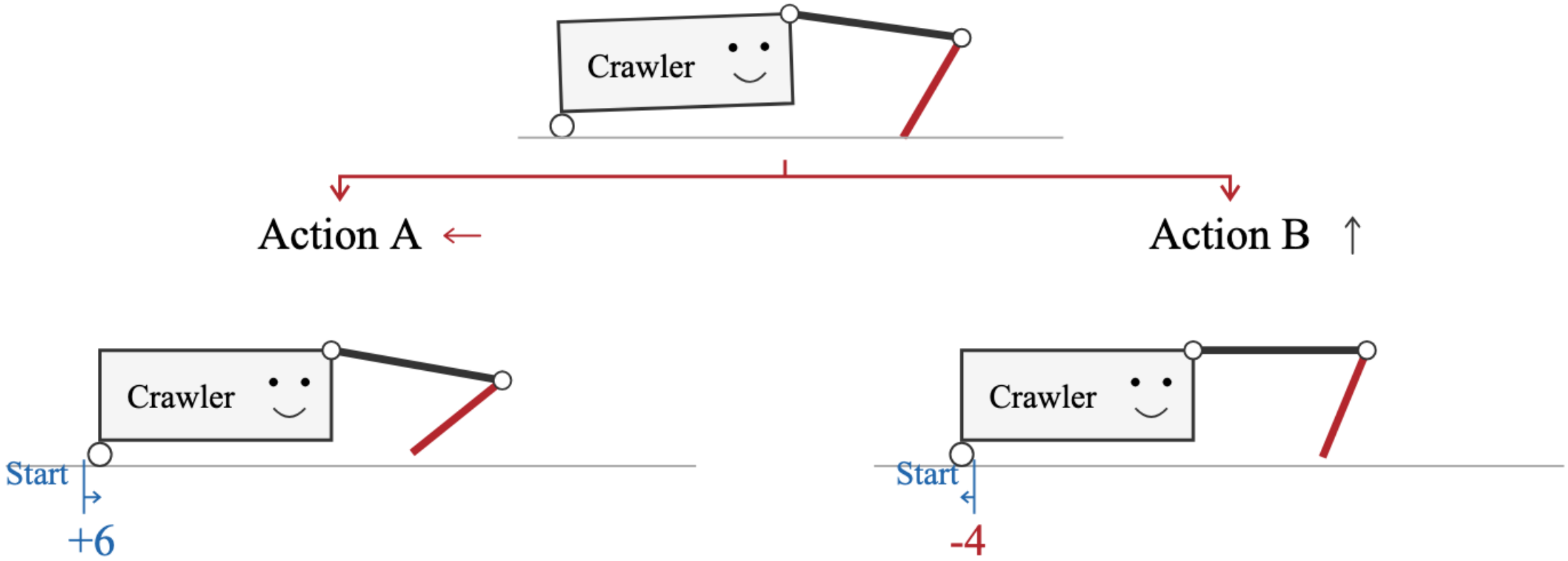
The Crawler Problem



Which move would you choose?

Immediate Rewards

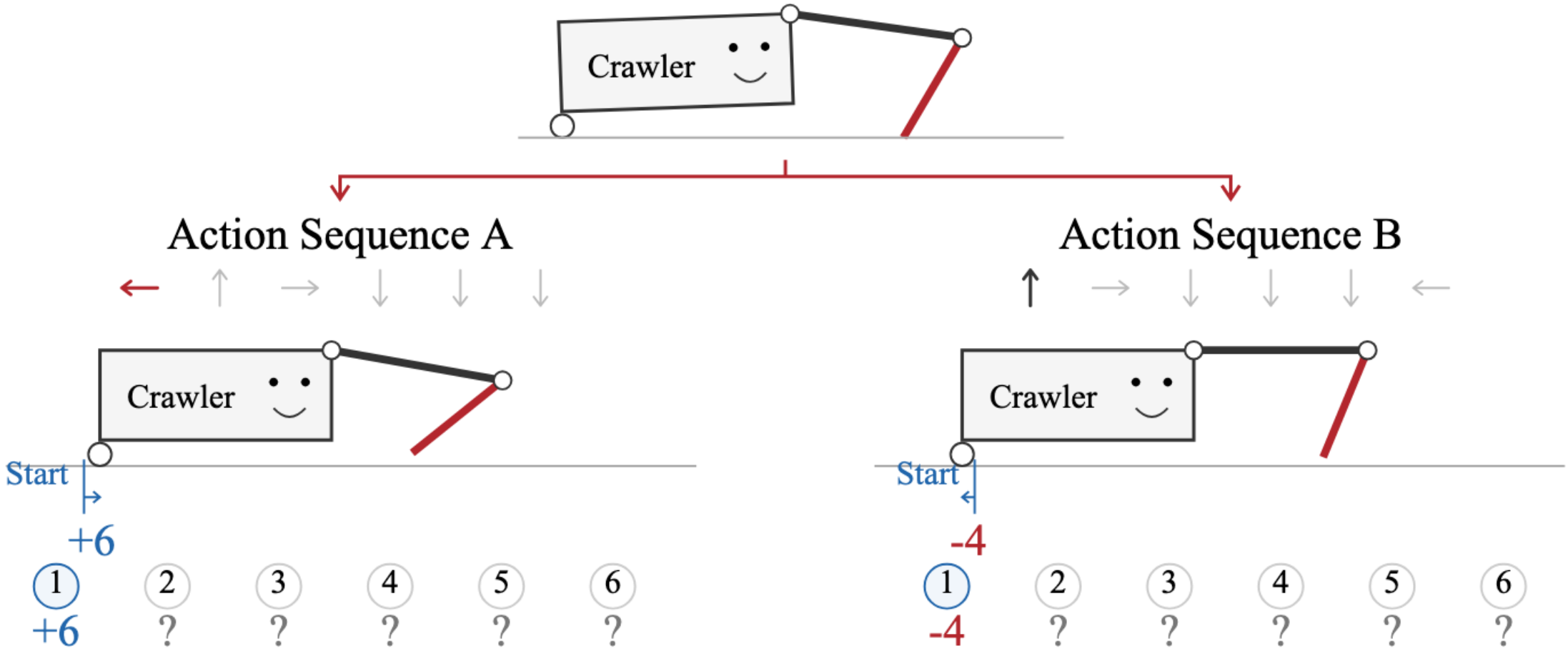
The Crawler Problem



Which move would you choose?

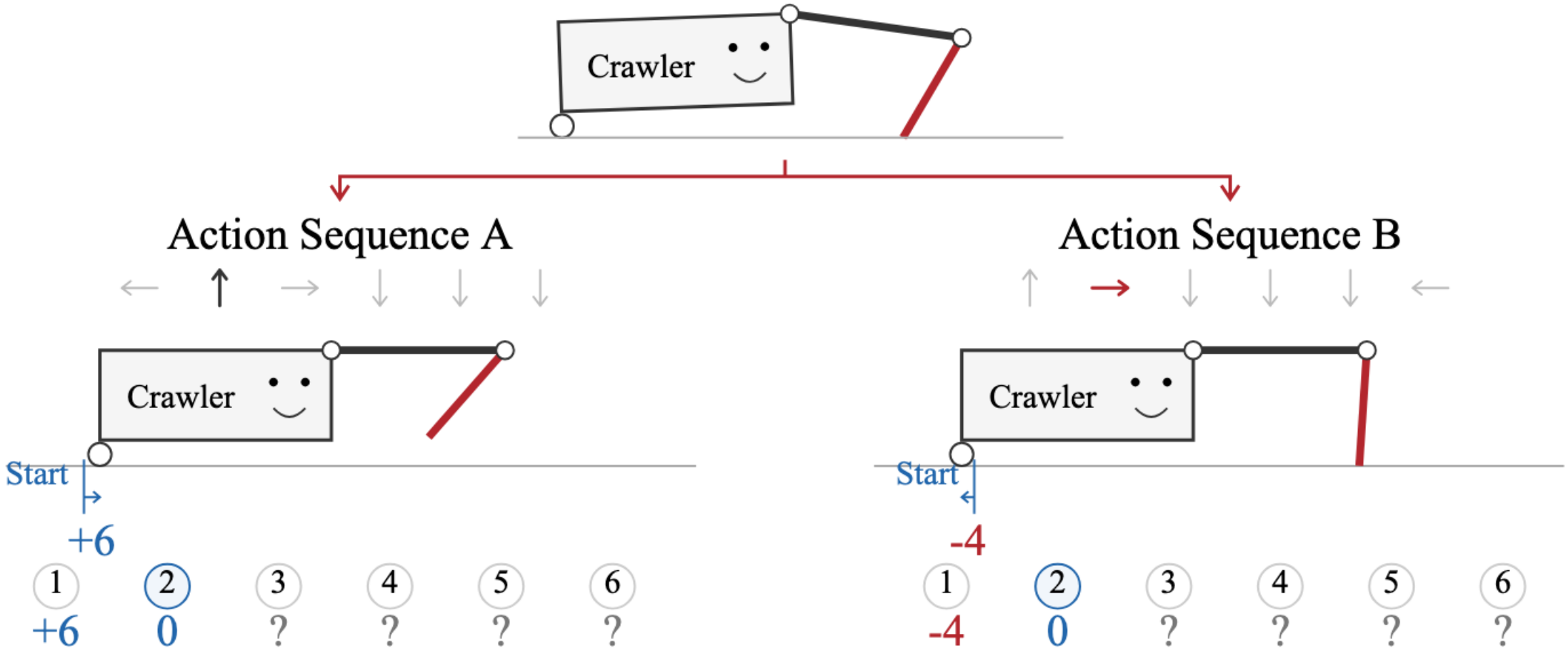
Immediate and Later Rewards

The Crawler Problem



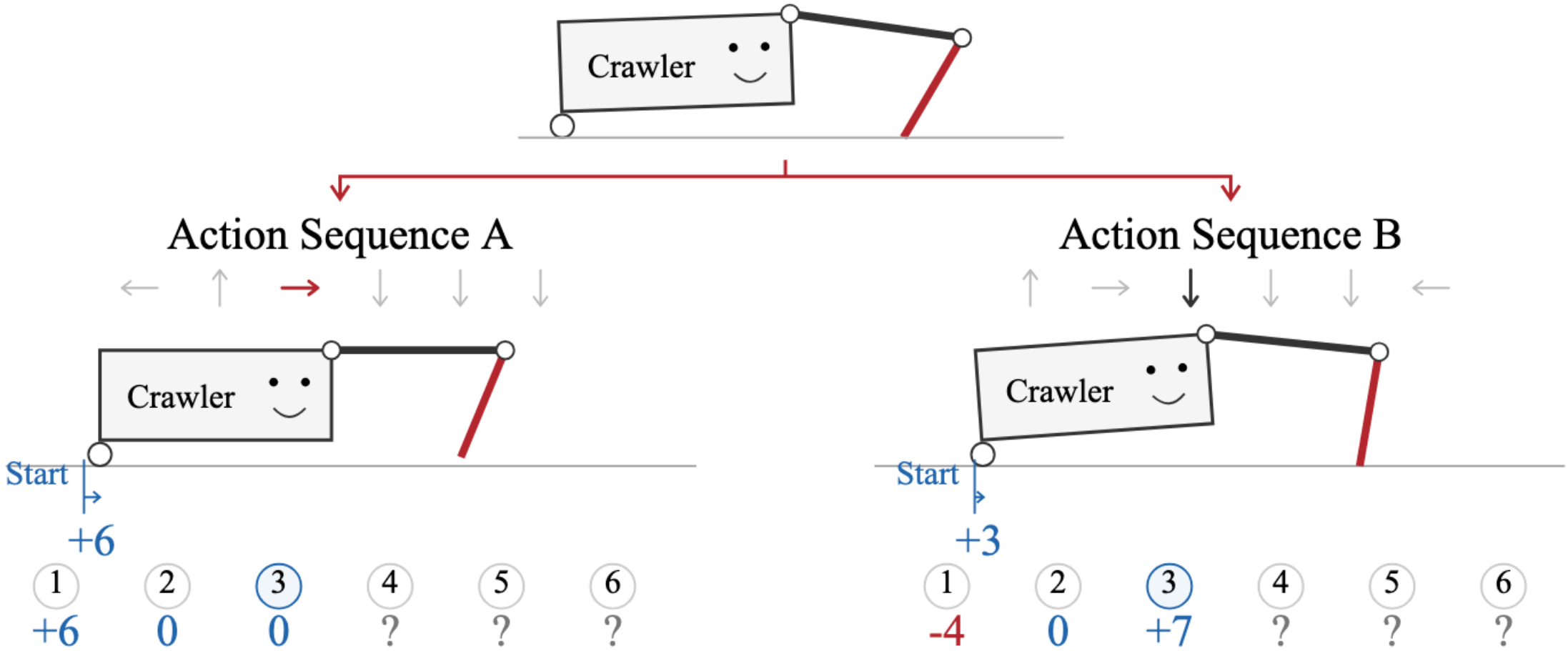
Immediate and Later Rewards

The Crawler Problem



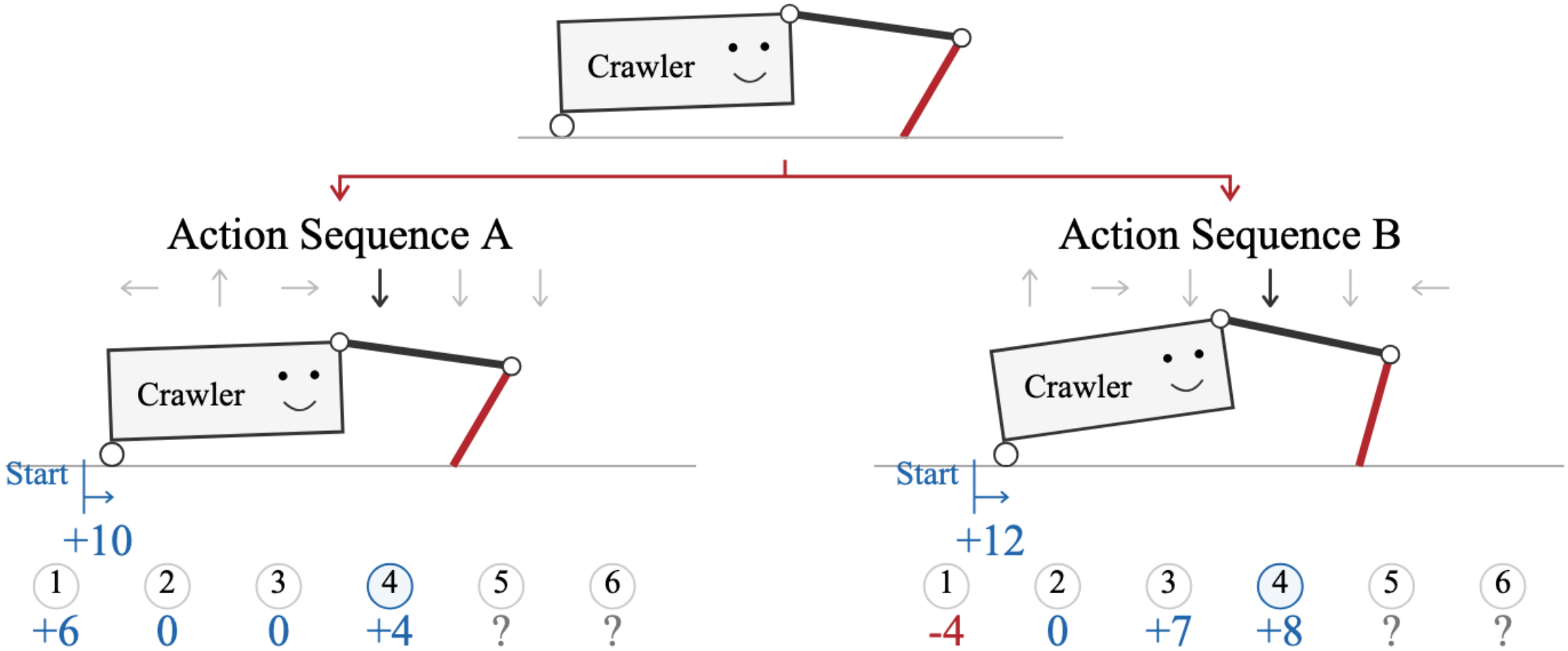
Immediate and Later Rewards

The Crawler Problem



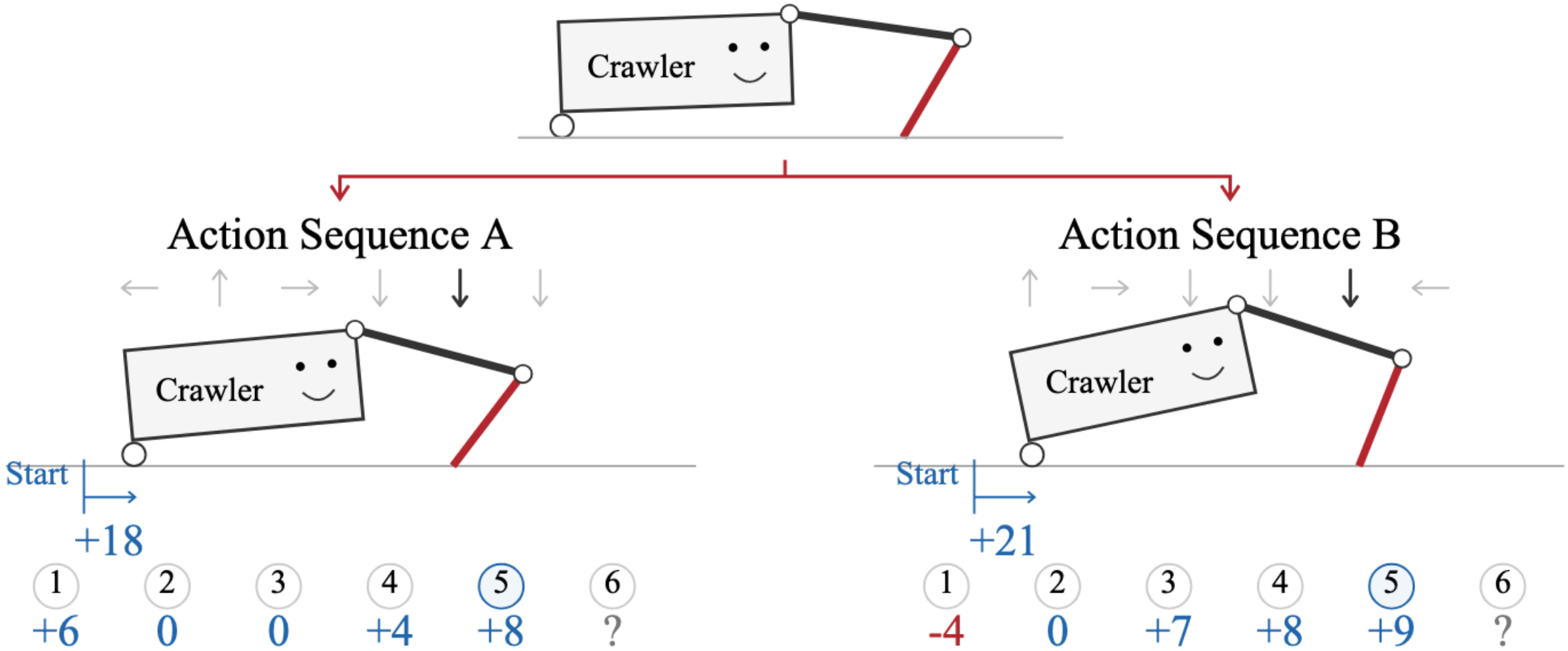
Immediate and Later Rewards

The Crawler Problem



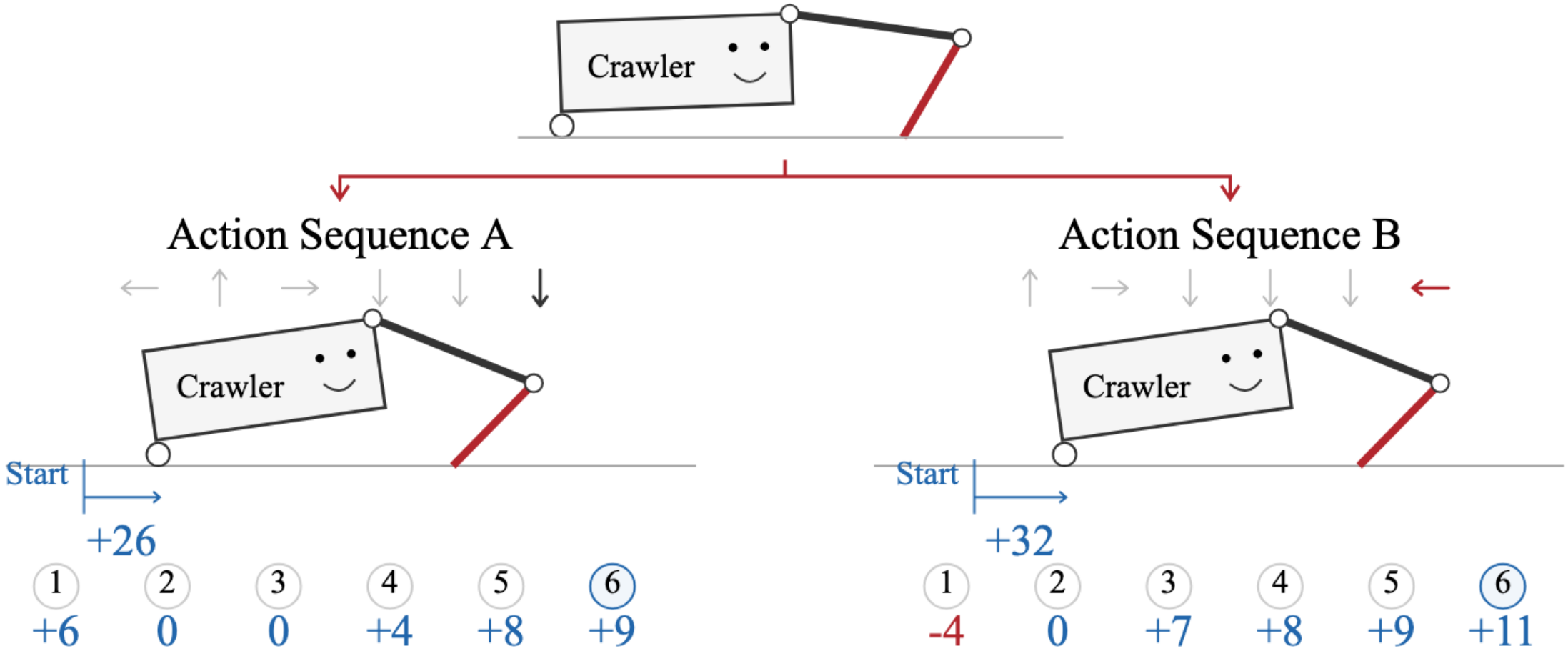
Immediate and Later Rewards

The Crawler Problem



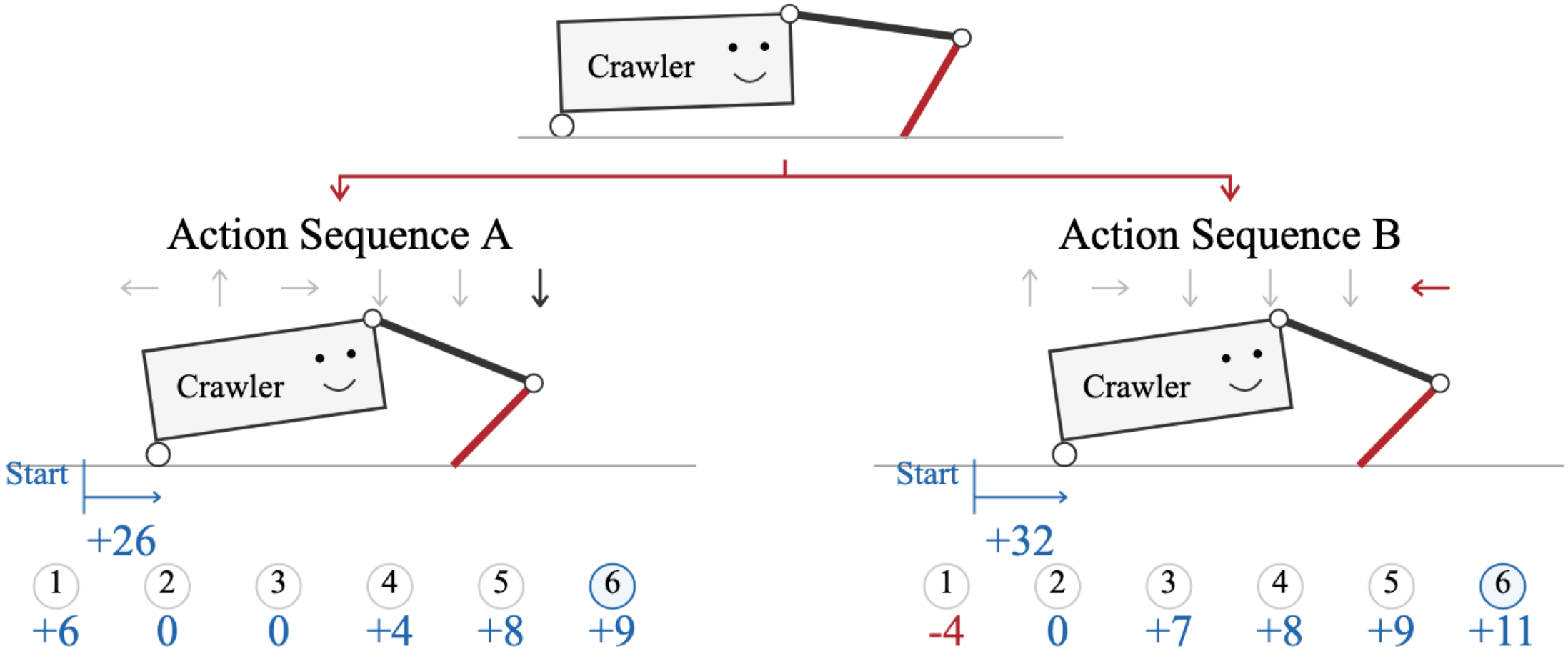
Immediate and Later Rewards

The Crawler Problem



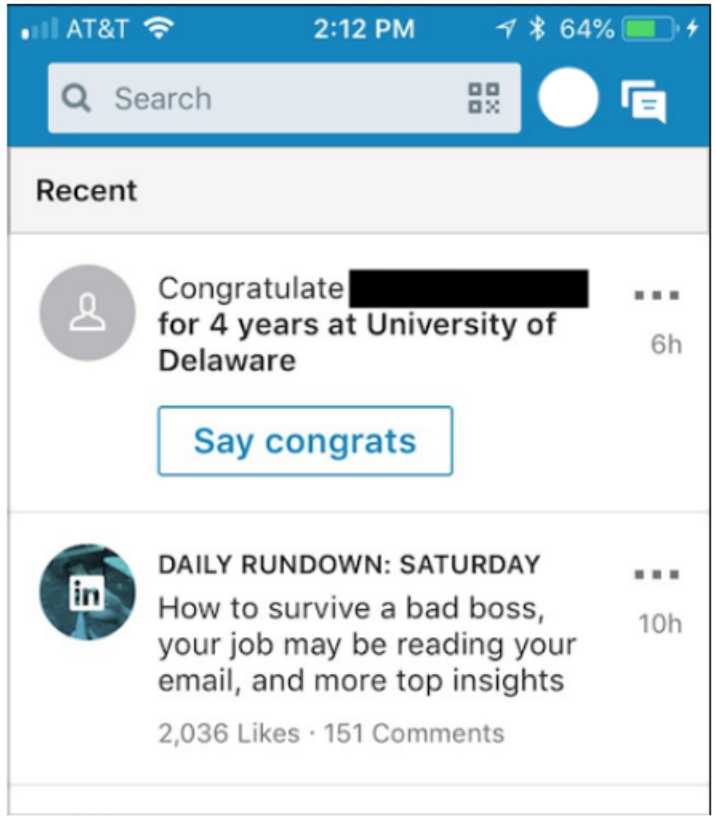
Immediate and Later Rewards

The Crawler Problem



A good choice considers immediate reward and later opportunities

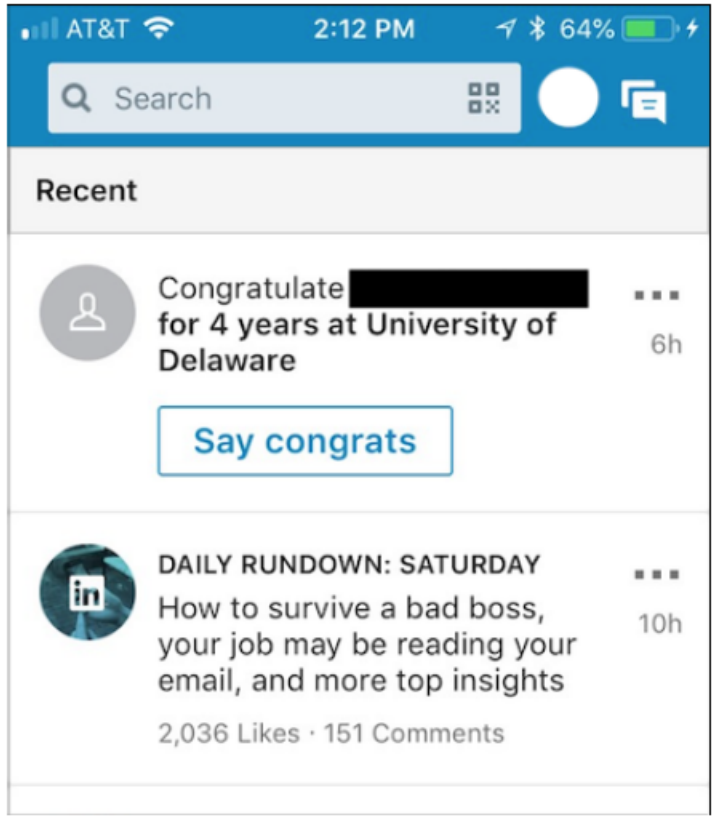
The LinkedIn Problem



Goal: Sustained User Engagement

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

The LinkedIn Problem



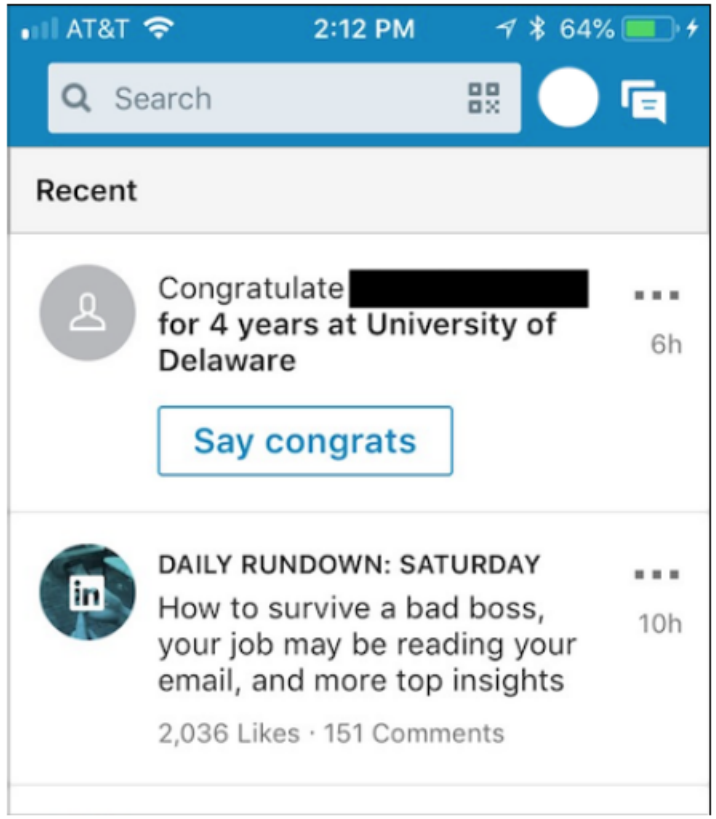
Goal: Sustained User Engagement

A Useful Notification

A Visit

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

The LinkedIn Problem



Goal: Sustained User Engagement

A Useful Notification

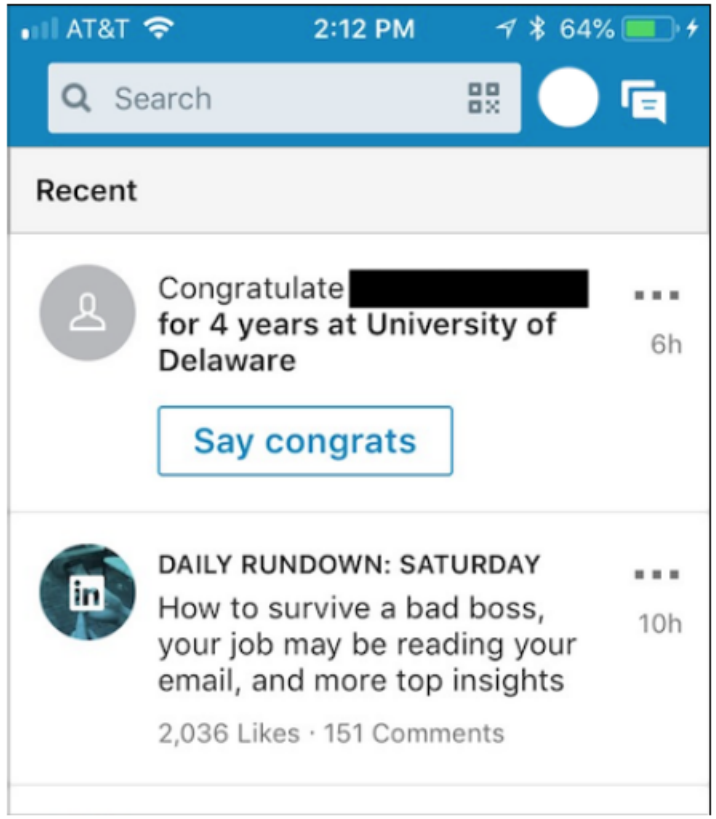
A Visit

Send More Updates

Send Fewer Updates

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

The LinkedIn Problem



Goal: Sustained User Engagement

A Useful Notification

A Visit

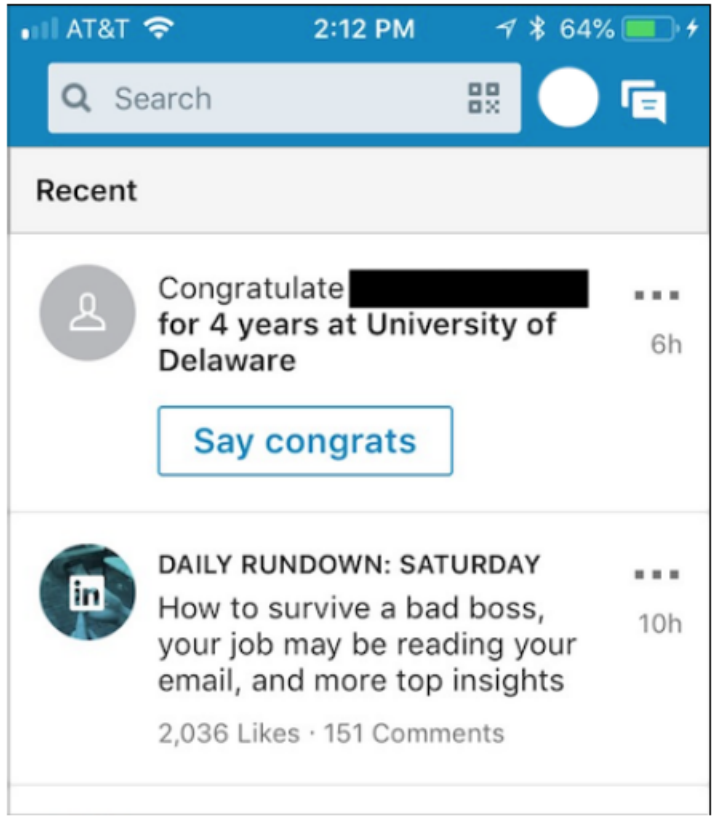
Send More Updates

Send Fewer Updates

What Happens Next?

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

The LinkedIn Problem



Goal: Sustained User Engagement

A Useful Notification

A Visit

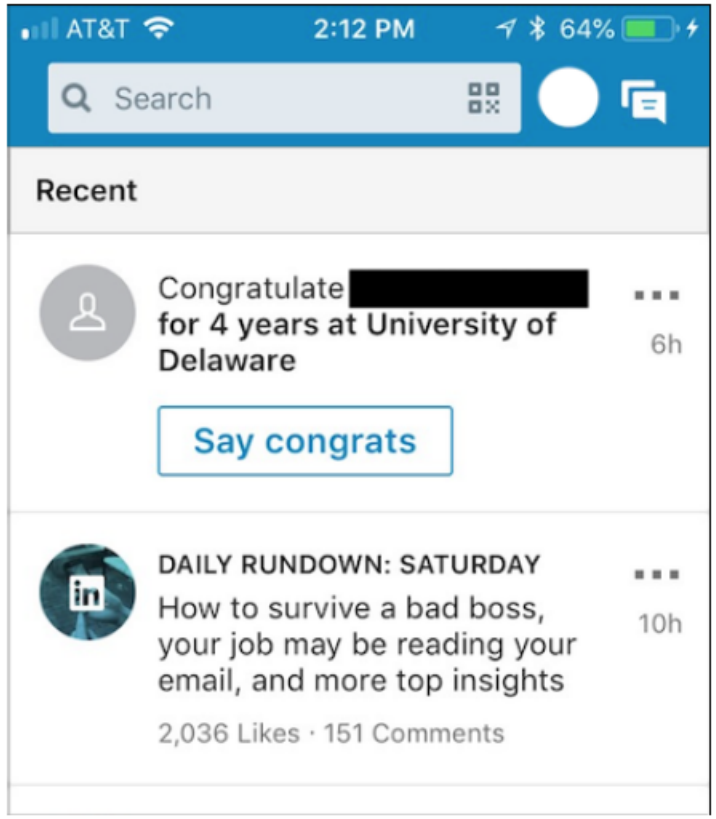
Send More Updates

Send Fewer Updates

Mutes Notifications

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

The LinkedIn Problem



Goal: Sustained User Engagement

A Useful Notification

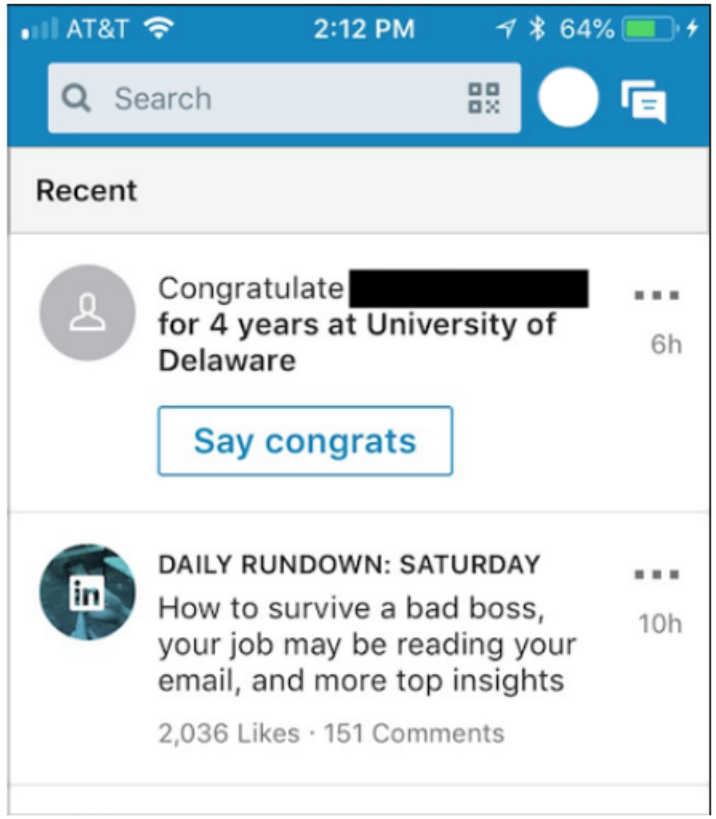
A Visit

Send More Updates → Mutes Notifications

Send Fewer Updates → Keeps Visiting

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

The LinkedIn Problem



Goal: Sustained User Engagement

A Useful Notification

A Visit

Send More Updates → Mutes Notifications

Send Fewer Updates → Keeps Visiting

How Often Should We Notify Each Member?

Yuan, Y., Muralidharan, A., Nandy, P., Cheng, M., & Prabhakar, P. (2022). *Offline Reinforcement Learning for Mobile Notifications.* CIKM '22, pp. 3614–3623. https://arxiv.org/pdf/2202.03867

LinkedIn's Notification Timing

The LinkedIn Problem

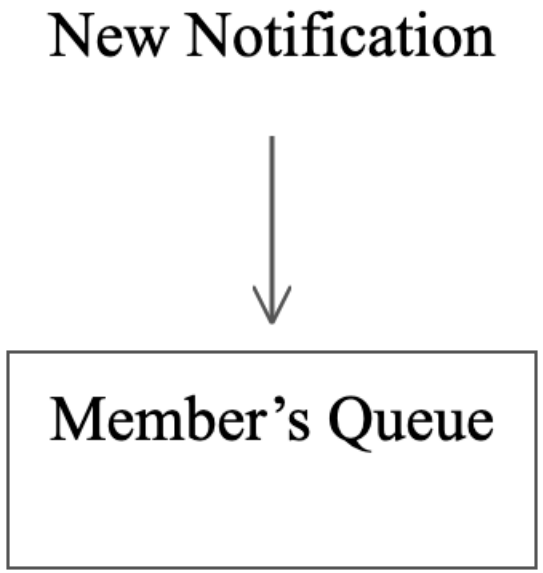
New Notification

Member's Queue

Adapted from Yuan et al. (2022), Figure 2: Notification Spacing System
https://arxiv.org/pdf/2202.03867

LinkedIn’s Notification Timing

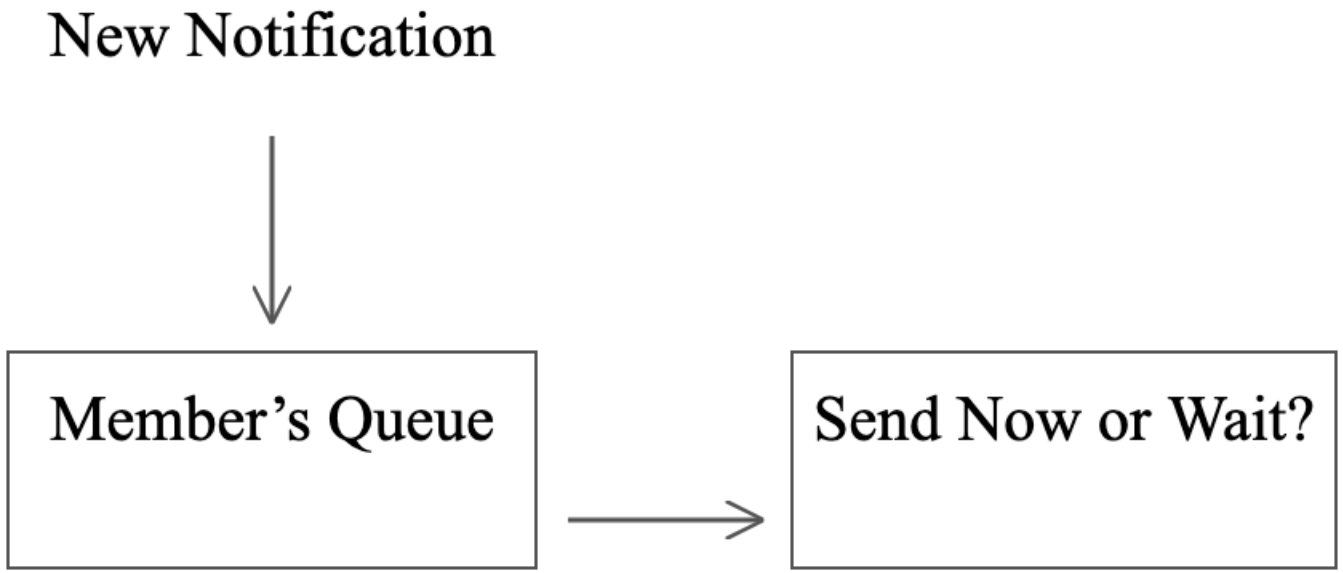
The LinkedIn Problem



Adapted from Yuan et al. (2022), Figure 2: Notification Spacing System
https://arxiv.org/pdf/2202.03867

LinkedIn’s Notification Timing

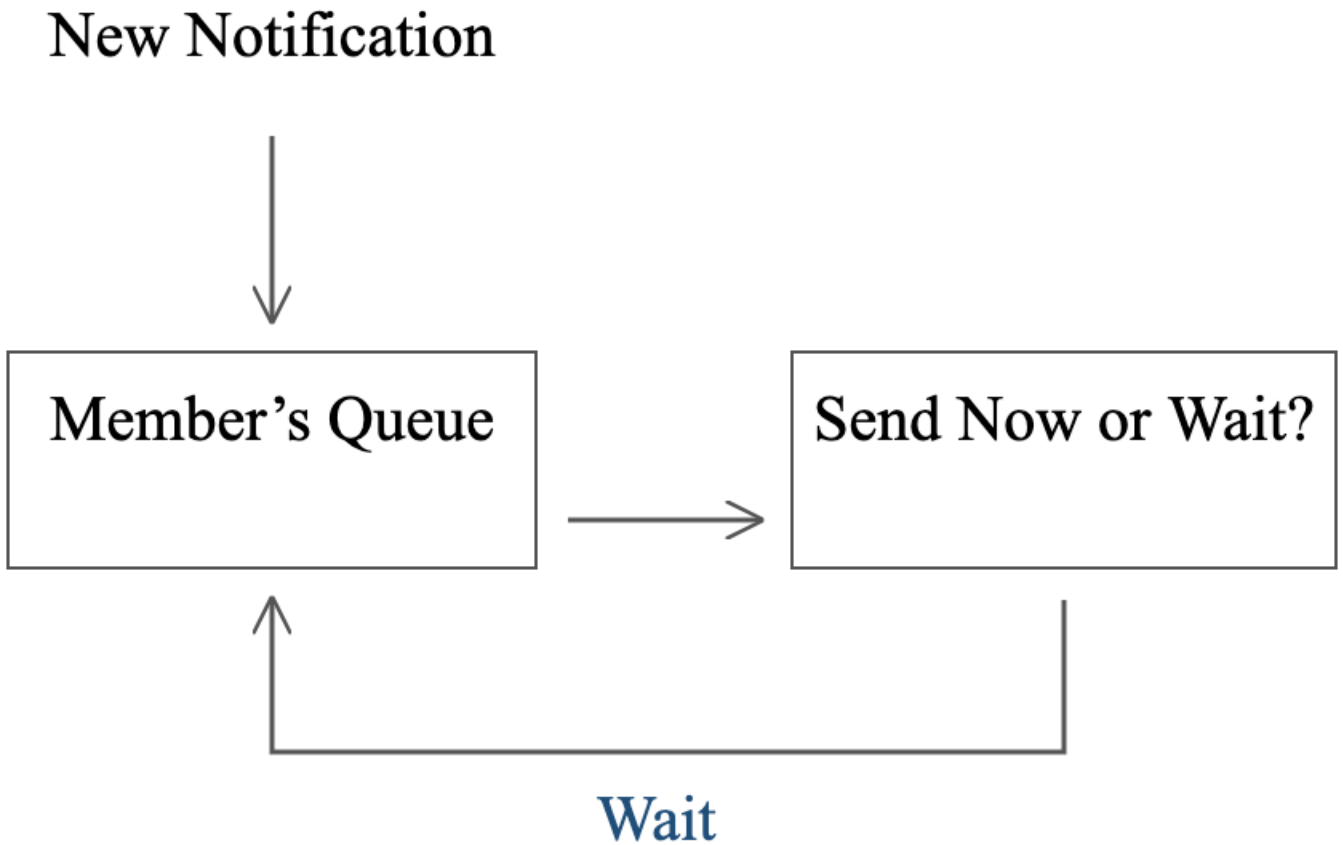
The LinkedIn Problem



Adapted from Yuan et al. (2022), Figure 2: Notification Spacing System
https://arxiv.org/pdf/2202.03867

LinkedIn's Notification Timing

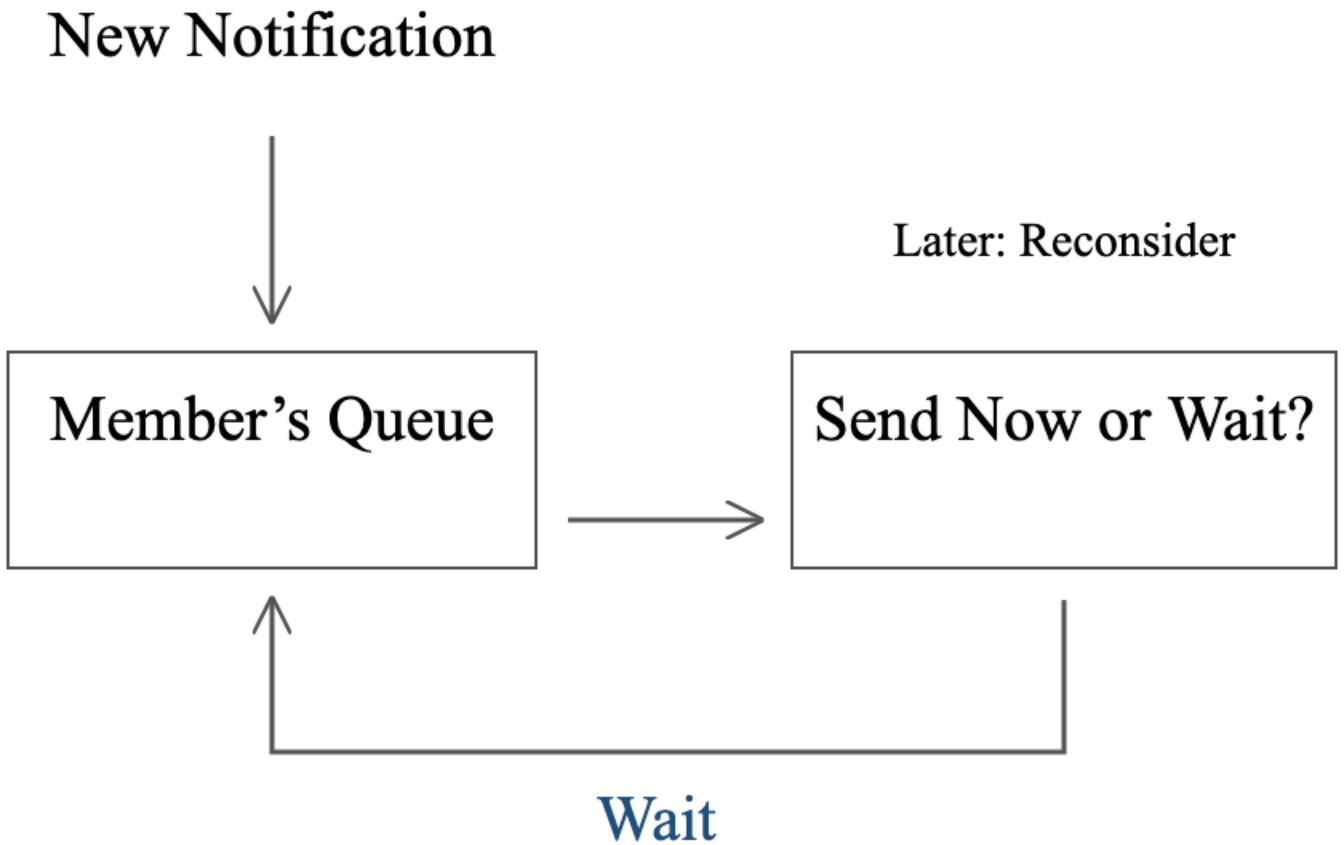
The LinkedIn Problem



Adapted from Yuan et al. (2022), Figure 2: Notification Spacing System
https://arxiv.org/pdf/2202.03867

LinkedIn's Notification Timing

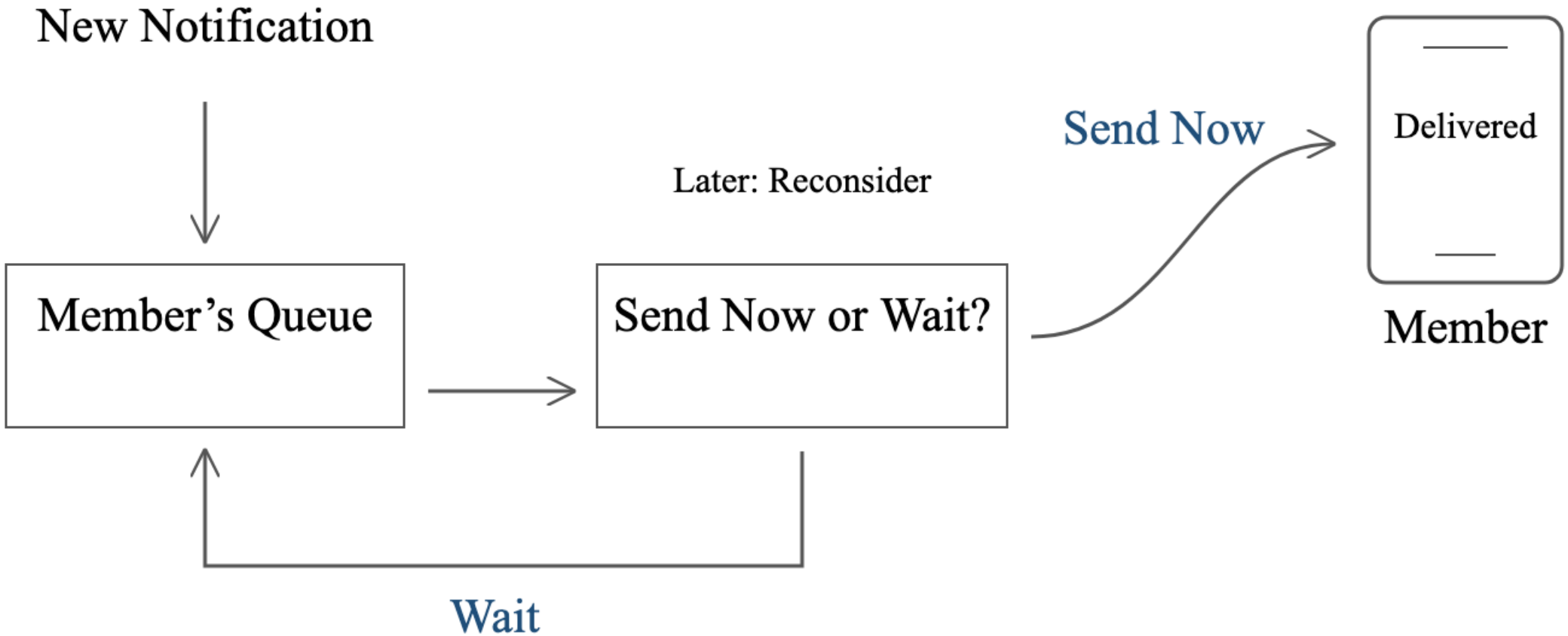
The LinkedIn Problem



Adapted from Yuan et al. (2022), Figure 2: Notification Spacing System
https://arxiv.org/pdf/2202.03867

LinkedIn's Notification Timing

The LinkedIn Problem



Repeated Send / Wait Choices Control Notification Frequency

Adapted from Yuan et al. (2022), Figure 2: Notification Spacing System
https://arxiv.org/pdf/2202.03867

State, Action and Reward

The LinkedIn Problem

The Crawler Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

State, Action and Reward

The LinkedIn Problem

The Crawler Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

State, Action and Reward

The LinkedIn Problem

The Crawler Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Reward

LinkedIn: A Visit

Crawler: Forward Progress

State, Action and Reward

The LinkedIn Problem

The Crawler Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

State, Action and Reward

The LinkedIn Problem

The Crawler Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Learn From Experience to Improve Reward Across a Sequence

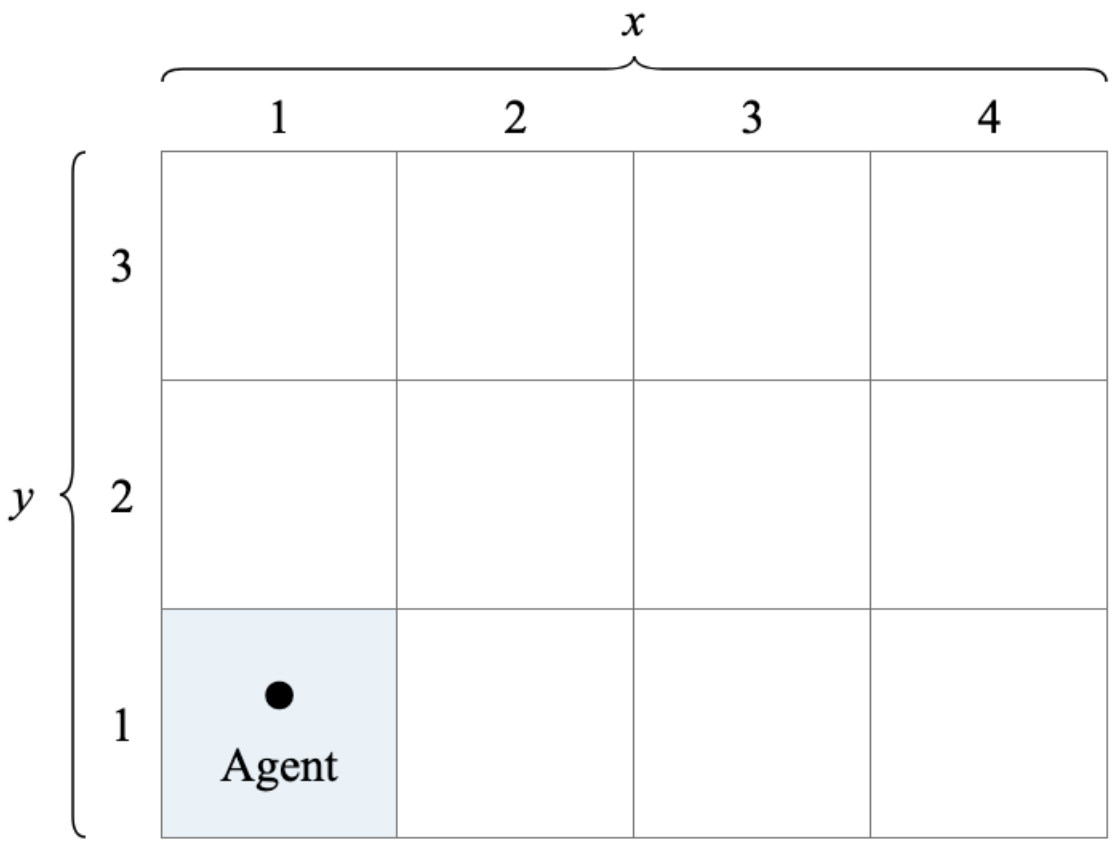
Quiz 1

Scan the QR Code to Open Ed

Answer the Questions

Share Any Questions You Have

The Gridworld Problem

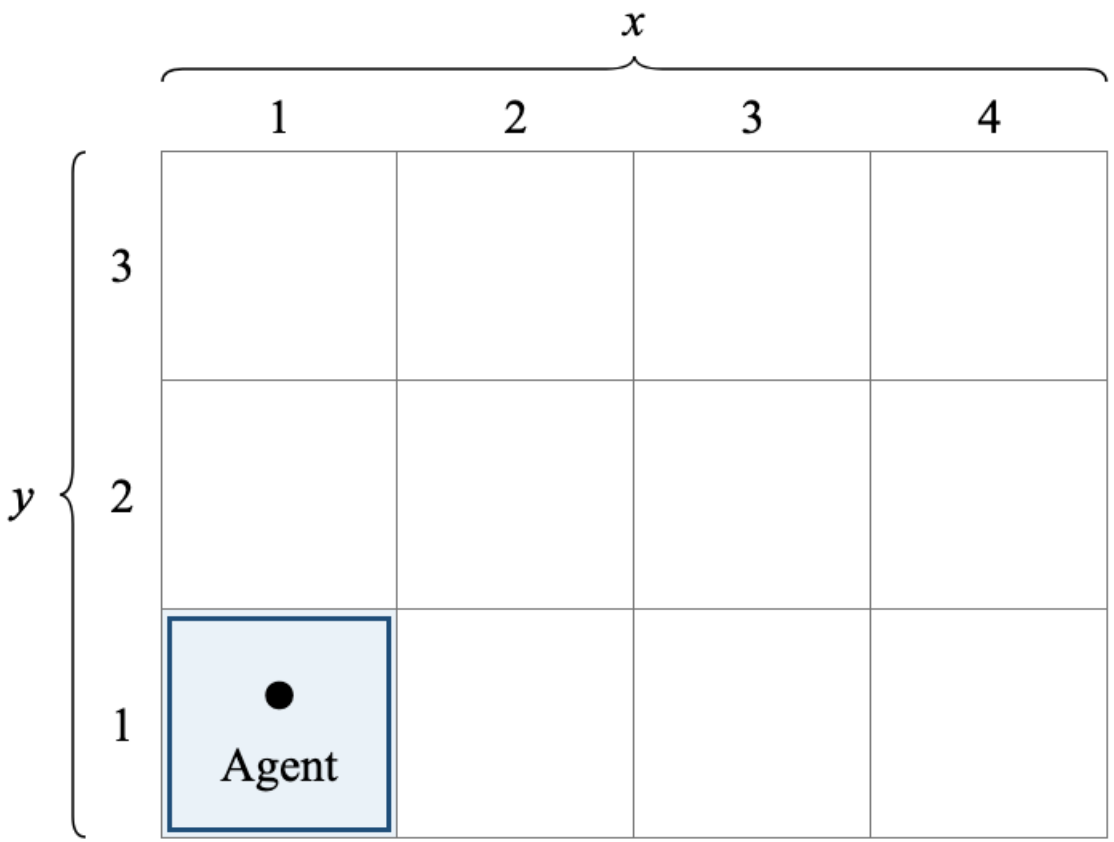


State (x, y)

Agent at (1, 1)

x increases right; y increases up

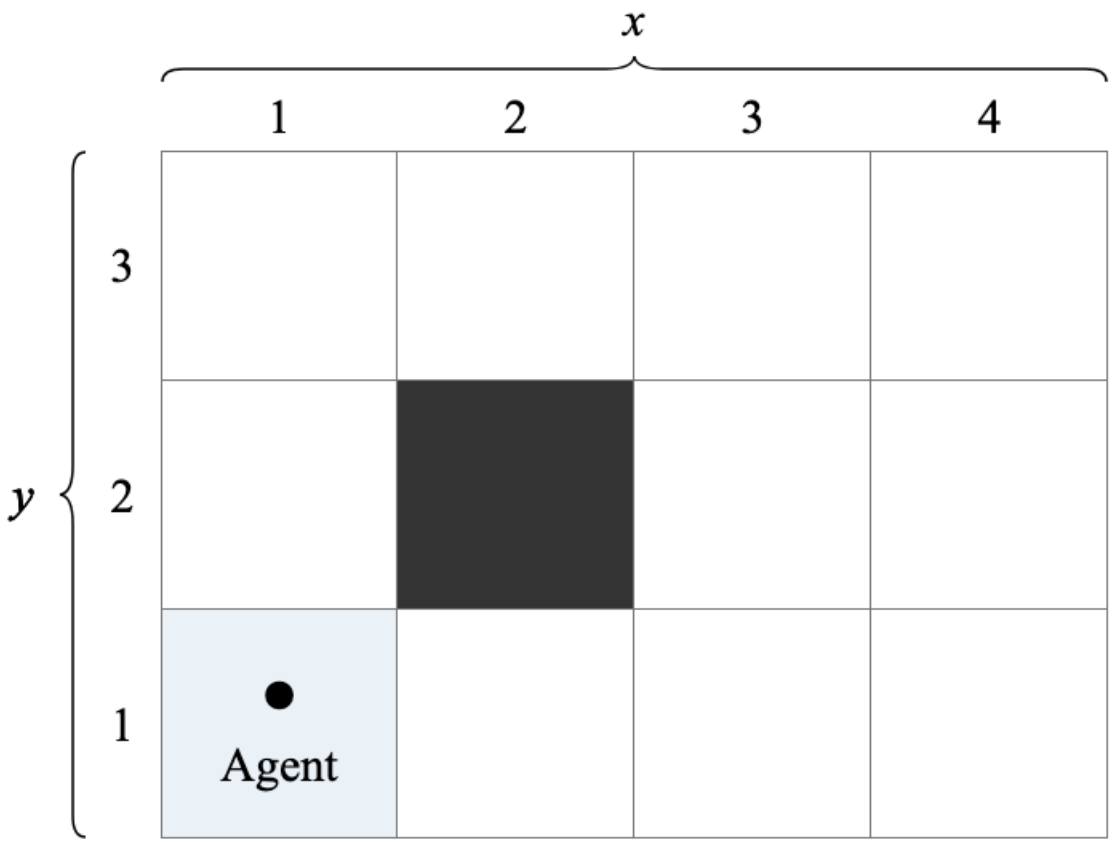
The Gridworld Problem



Regular states

Available actions: North, East, South, West

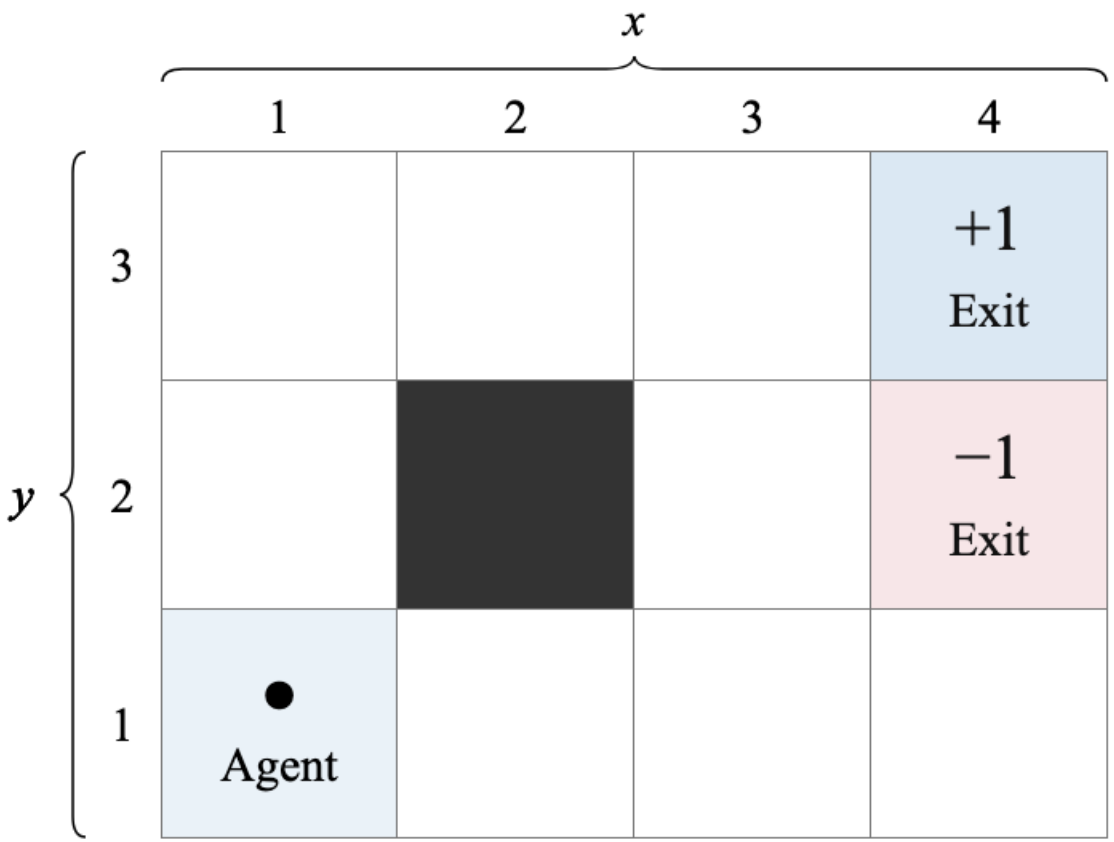
The Gridworld Problem



Wall at (2, 2)

A move into the wall leaves the agent in place

The Gridworld Problem

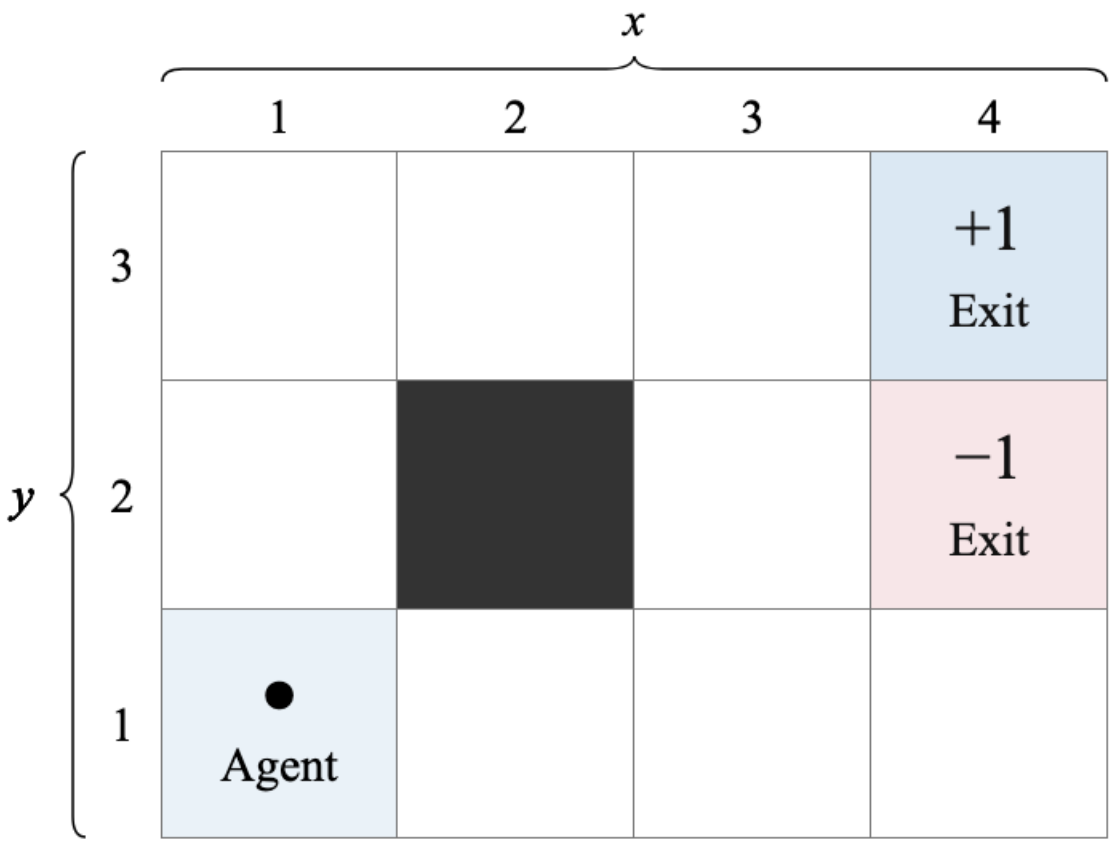


Exit states

Only available action: Exit

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

The Gridworld Problem

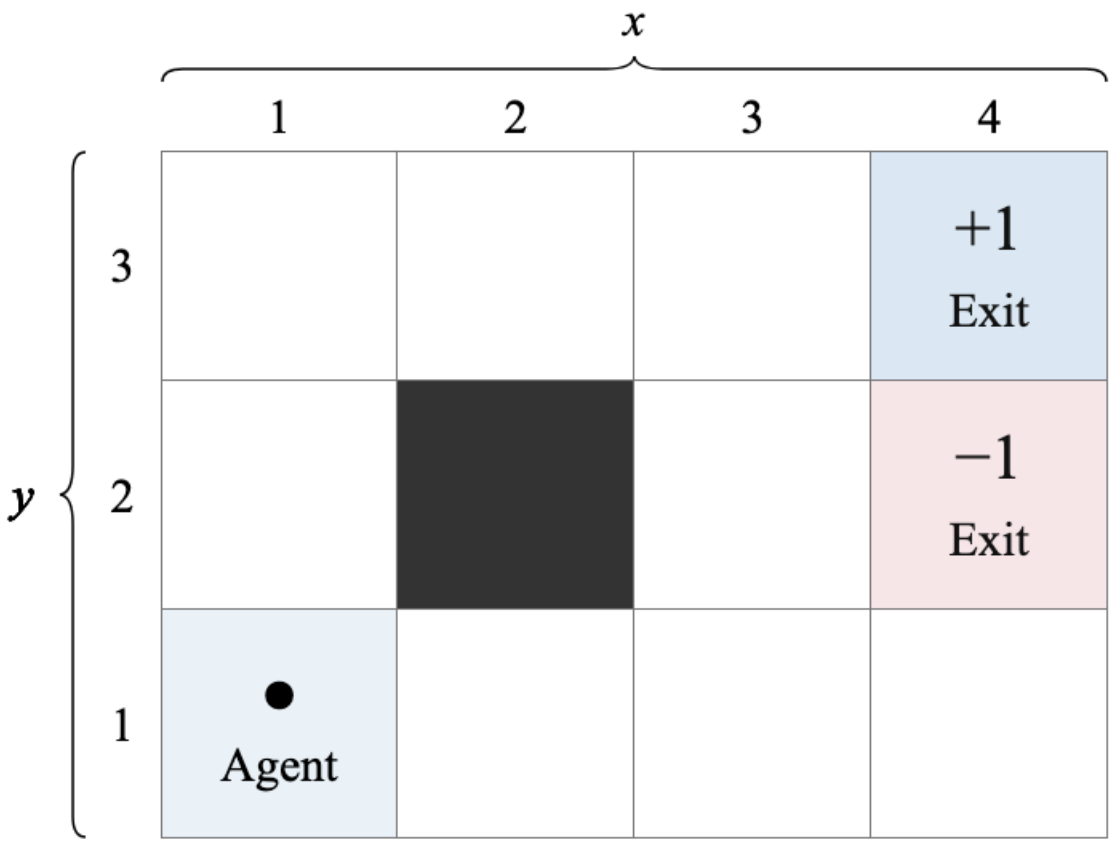


Movement reward

Every compass action pays −0.1

Including blocked moves and moves into exit squares

The Gridworld Problem

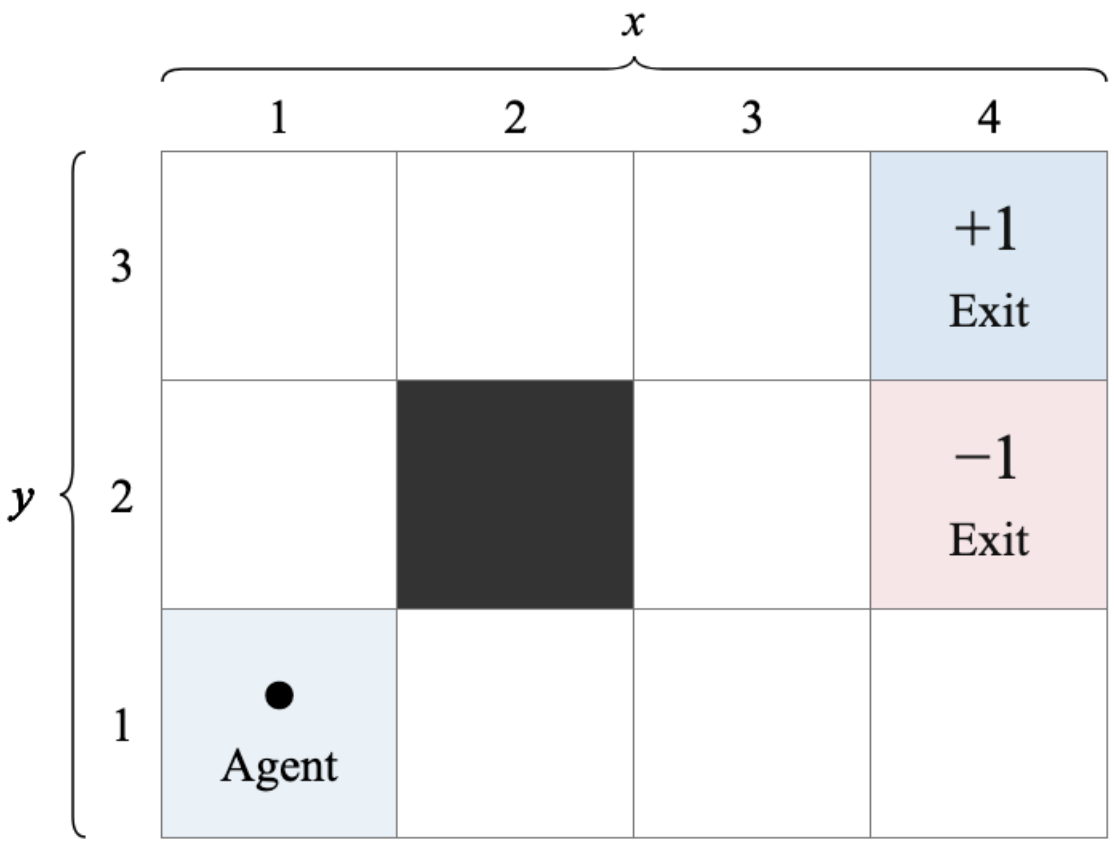


Noisy movement

Intended direction: 0.8

Each sideways direction: 0.1

The Gridworld Problem



Noisy movement

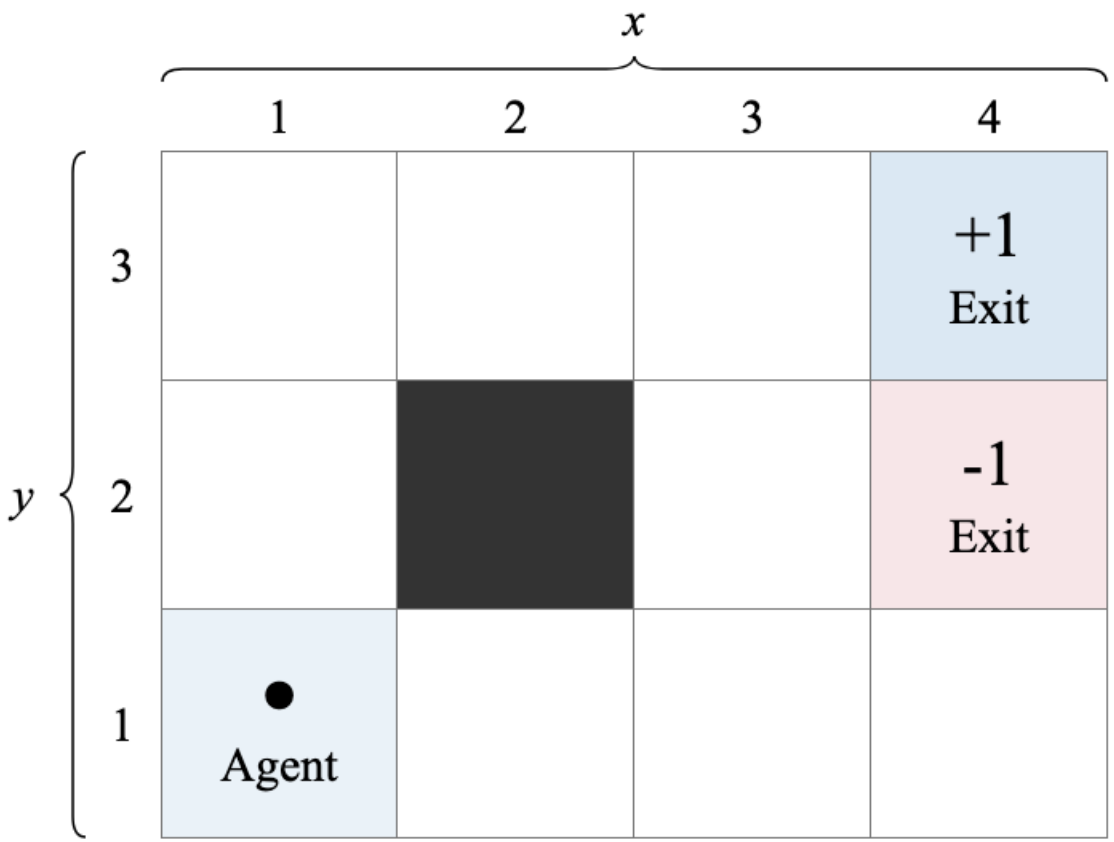
Intended direction: 0.8

Each sideways direction: 0.1

The agent observes its actual square after every action

Deterministic Grid World

The Gridworld Problem



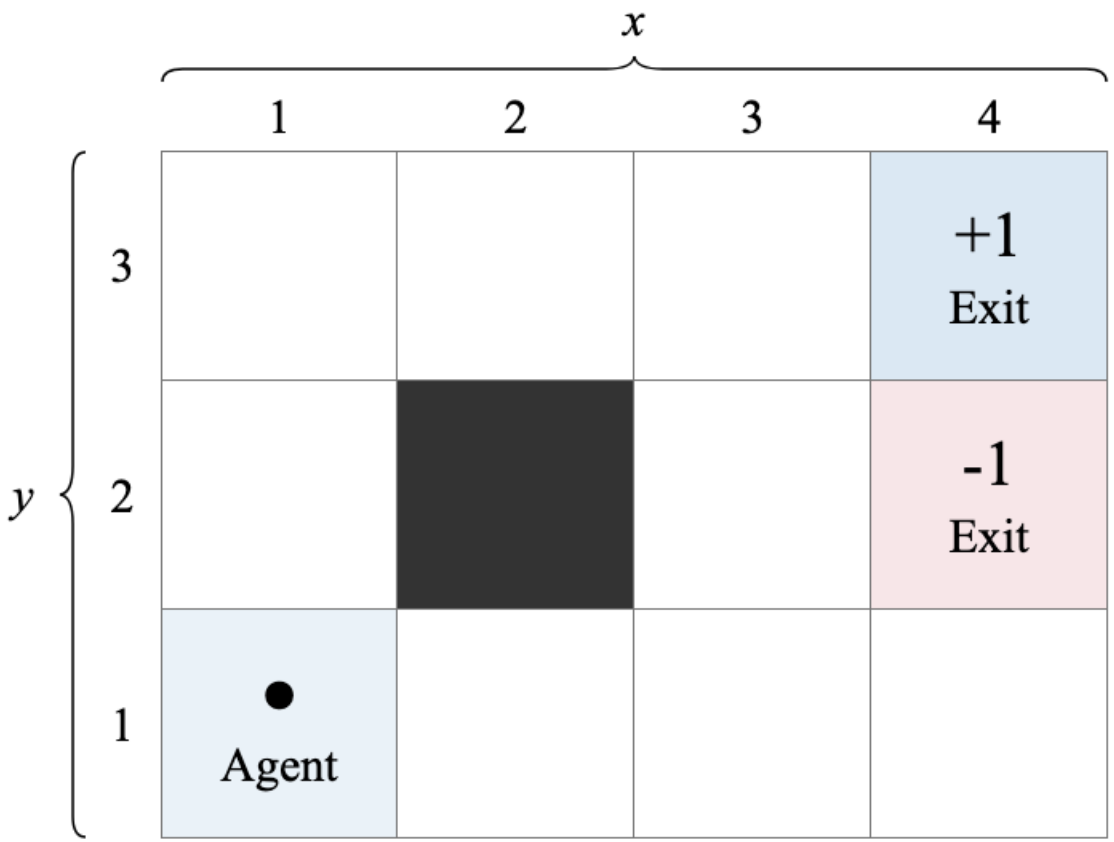
Choose a move

Last reward: 0.0

Total reward: 0.0

Stochastic Grid World

The Gridworld Problem



Click an available action

Intended direction: 0.8

Each sideways direction: 0.1

State, Action and Reward

The LinkedIn Problem | The Crawler Problem | The Gridworld Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Gridworld: ?

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Gridworld: ?

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Gridworld: ?

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Gridworld: ?

State, Action and Reward

The LinkedIn Problem | The Crawler Problem | The Gridworld Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Gridworld: Current Square

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Gridworld: ?

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Gridworld: ?

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Gridworld: ?

State, Action and Reward

The LinkedIn Problem | The Crawler Problem | The Gridworld Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Gridworld: Current Square

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Gridworld: North, East, South, West, or Exit

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Gridworld: ?

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Gridworld: ?

State, Action and Reward

The LinkedIn Problem | The Crawler Problem | The Gridworld Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Gridworld: Current Square

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Gridworld: North, East, South, West, or Exit

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Gridworld: Move: −0.1; Exit: +1 or −1

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Gridworld: ?

State, Action and Reward

The LinkedIn Problem | The Crawler Problem | The Gridworld Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Gridworld: Current Square

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Gridworld: North, East, South, West, or Exit

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Gridworld: Move: −0.1; Exit: +1 or −1

Next State

LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Gridworld: Actual Square, or Episode Ends

State, Action and Reward

The LinkedIn Problem | The Crawler Problem | The Gridworld Problem

State

LinkedIn: Recent Contacts and Visits

Crawler: Current Pose

Gridworld: Current Square

Action

LinkedIn: Send or Wait

Crawler: Move a Joint

Gridworld: North, East, South, West, or Exit

Reward

LinkedIn: A Visit

Crawler: Forward Progress

Gridworld: Move: −0.1; Exit: +1 or −1

Next State

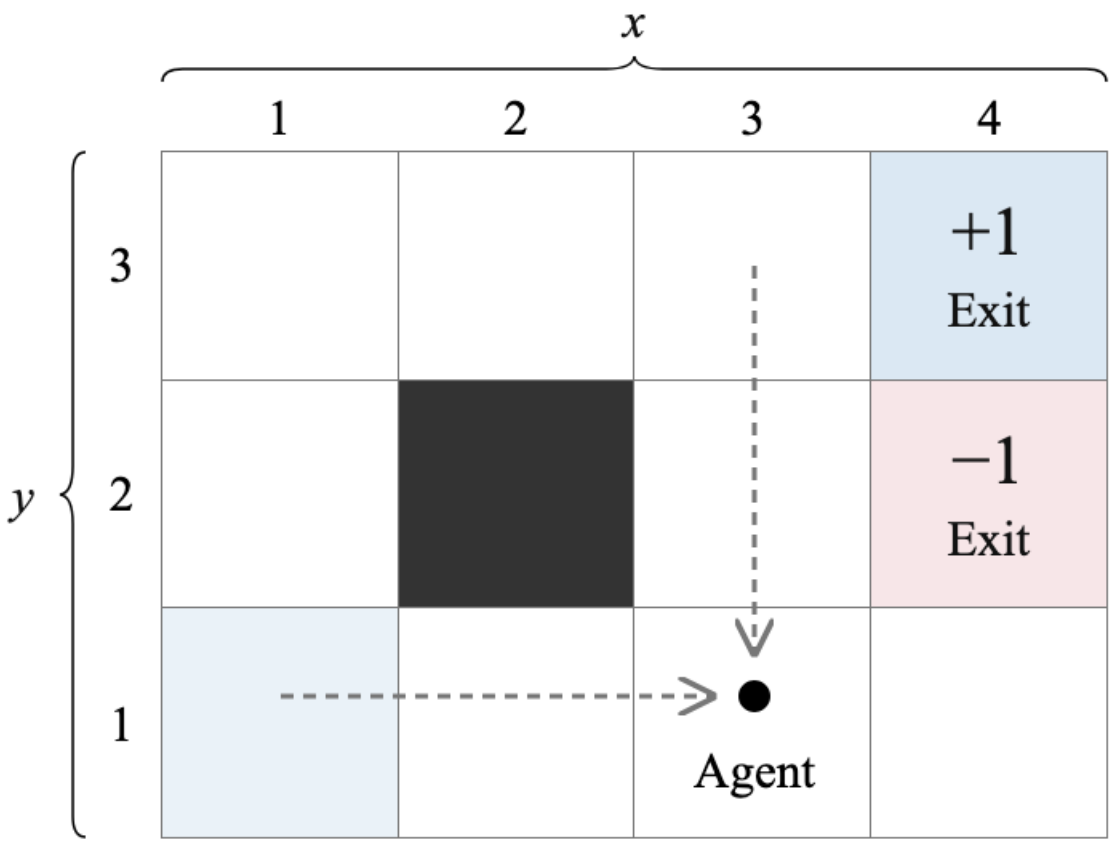
LinkedIn: Updated Contacts and Visits

Crawler: Updated Pose

Gridworld: Actual Square, or Episode Ends

Same Decision Structure; Here the Dynamics Are Known

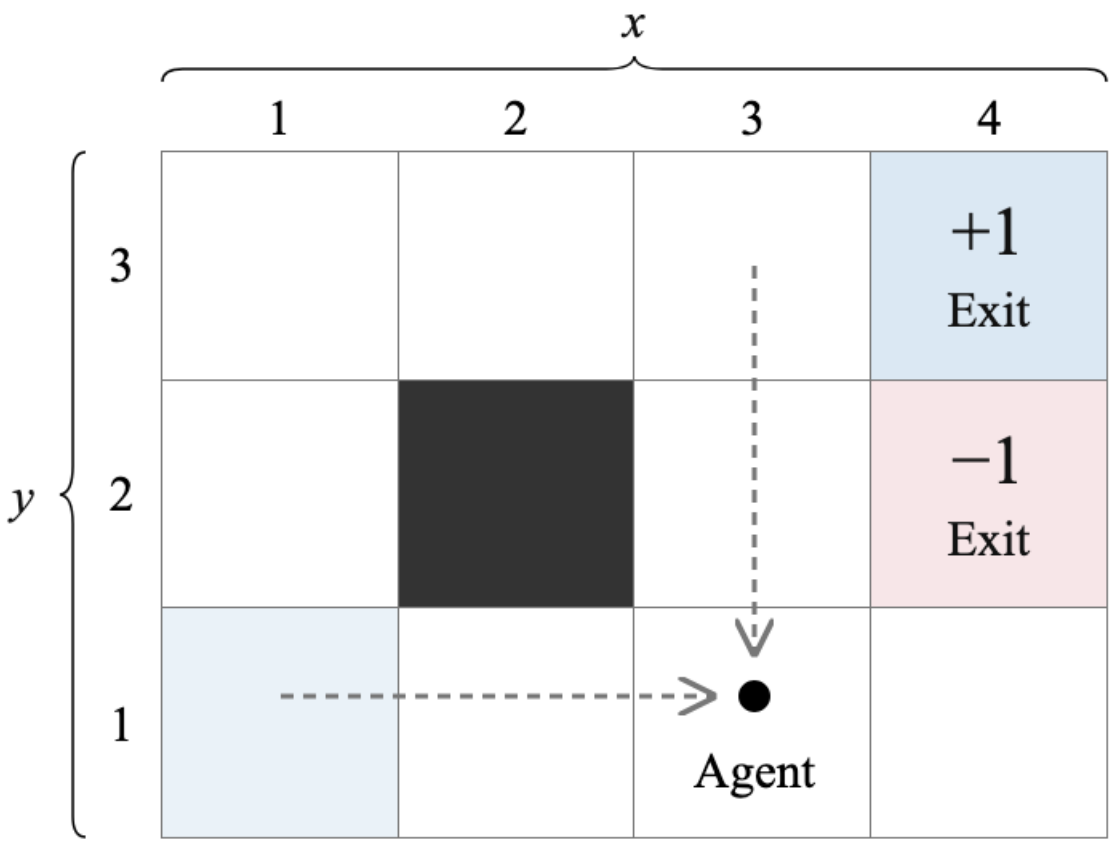
The Transition Model and the Markov Property



A: (1, 1) → (2, 1) → (3, 1)

B: (3, 3) → (3, 2) → (3, 1)

The Transition Model and the Markov Property



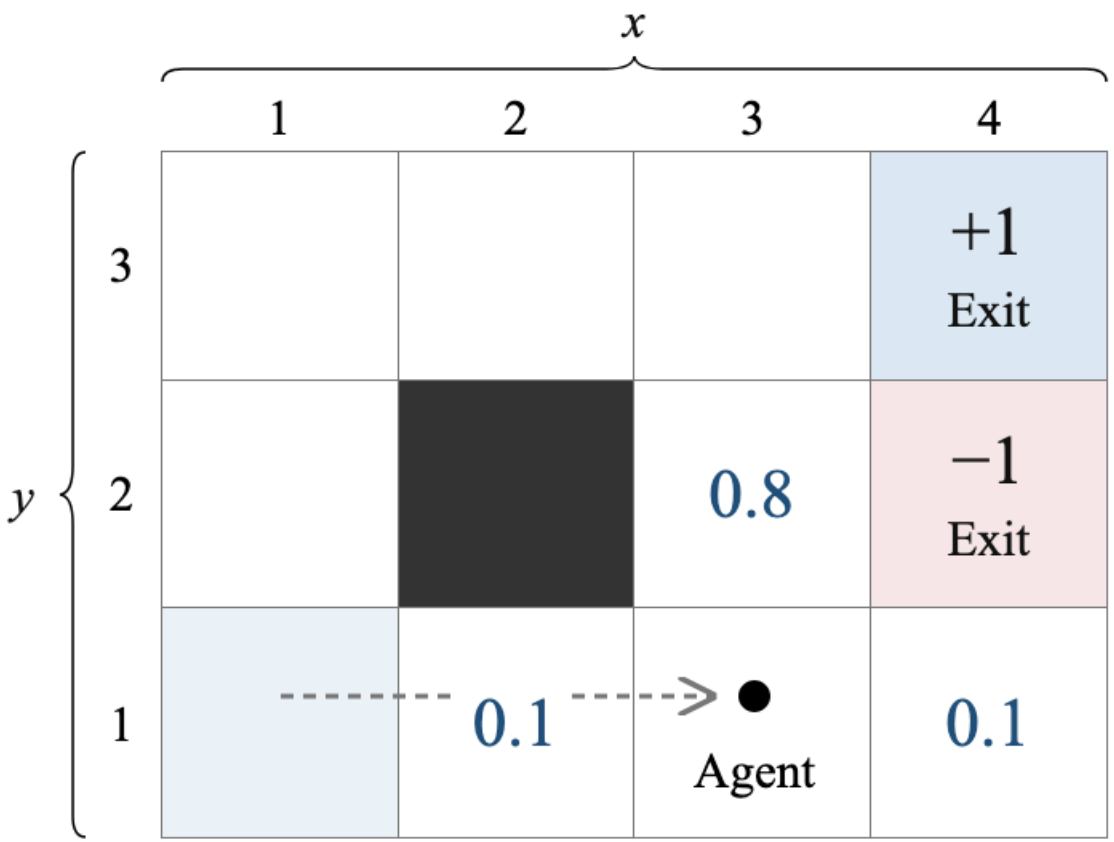
A: (1, 1) → (2, 1) → (3, 1)

B: (3, 3) → (3, 2) → (3, 1)

Same state: (3, 1)

Same action: North

The Transition Model and the Markov Property



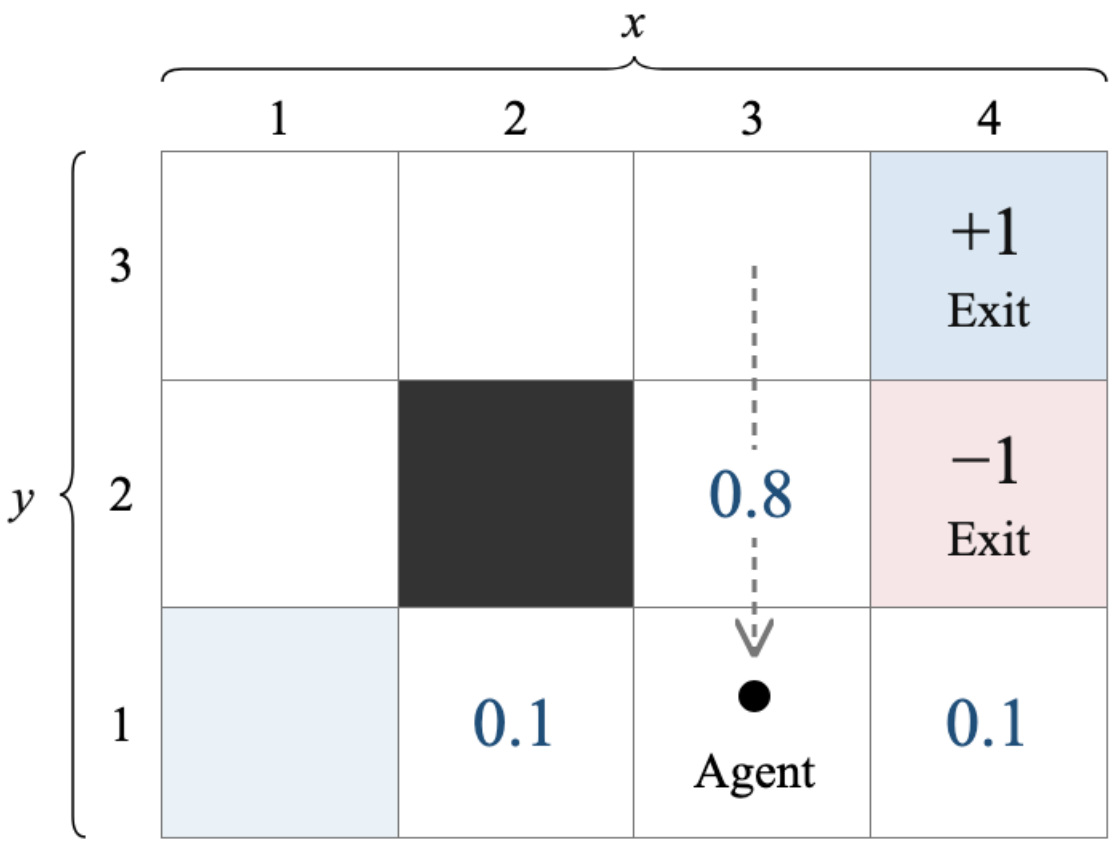
A: (1, 1) → (2, 1) → (3, 1)

B: (3, 3) → (3, 2) → (3, 1)

Same state: (3, 1)

Same action: North

The Transition Model and the Markov Property



A: (1, 1) → (2, 1) → (3, 1)

B: (3, 3) → (3, 2) → (3, 1)

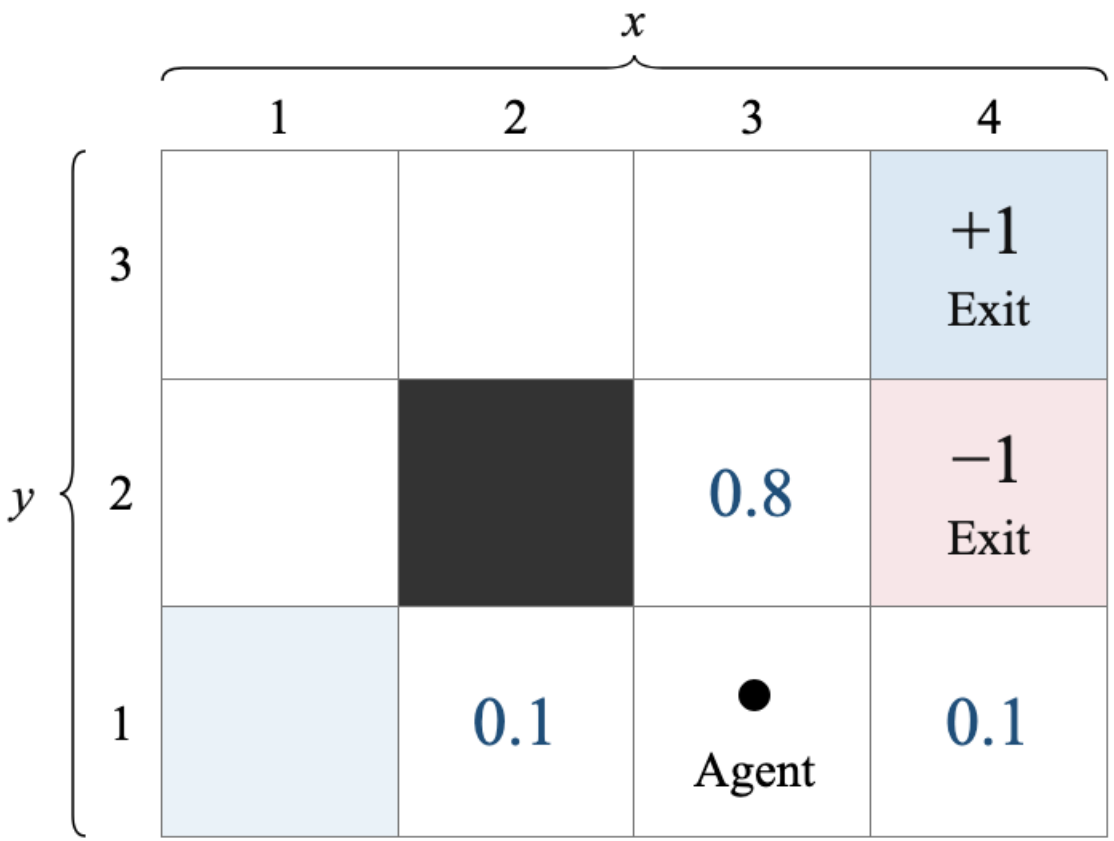
Same state: (3, 1)

Same action: North

Different histories

Same next-state probabilities

The Transition Model and the Markov Property



A: (1, 1) → (2, 1) → (3, 1)

B: (3, 3) → (3, 2) → (3, 1)

Same state: (3, 1)

Same action: North

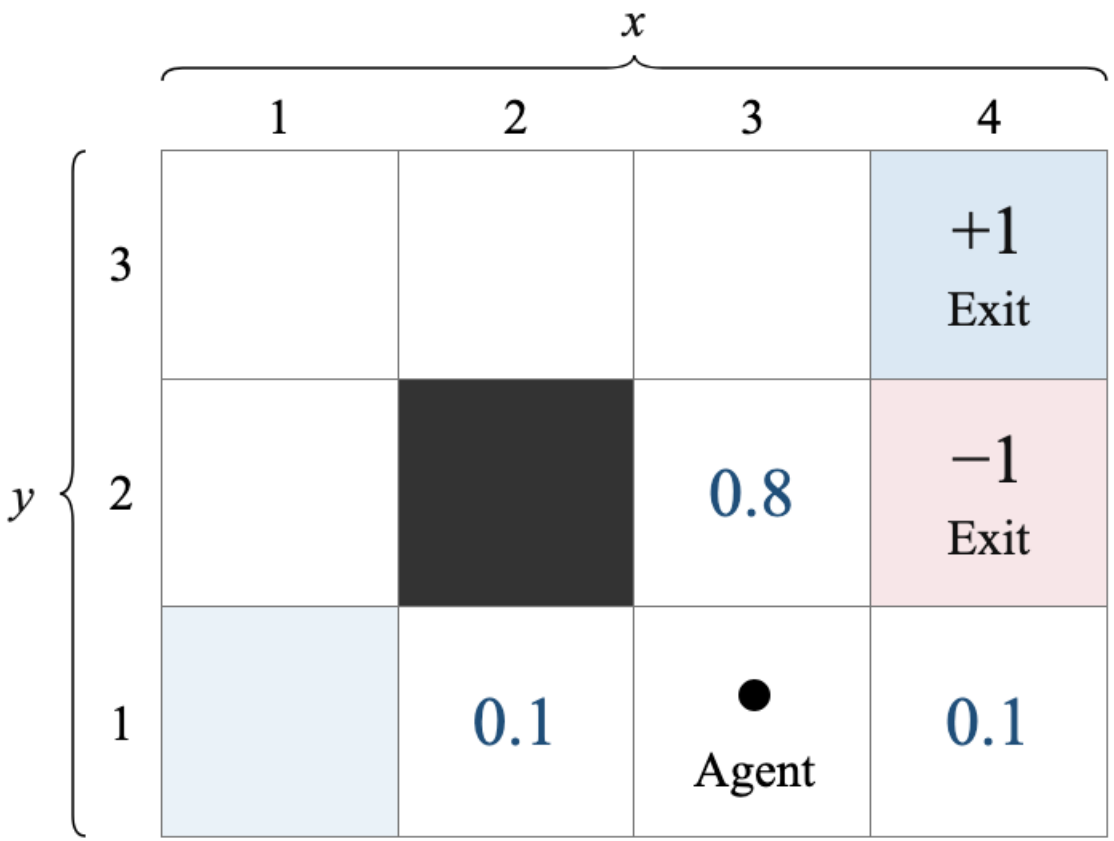
$$T(s, a, s') = P(s' \mid s, a)$$

Current state, action, next state

The current state and action suffice

Earlier history adds nothing

The Transition Model and the Markov Property



A: (1, 1) → (2, 1) → (3, 1)

B: (3, 3) → (3, 2) → (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

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

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

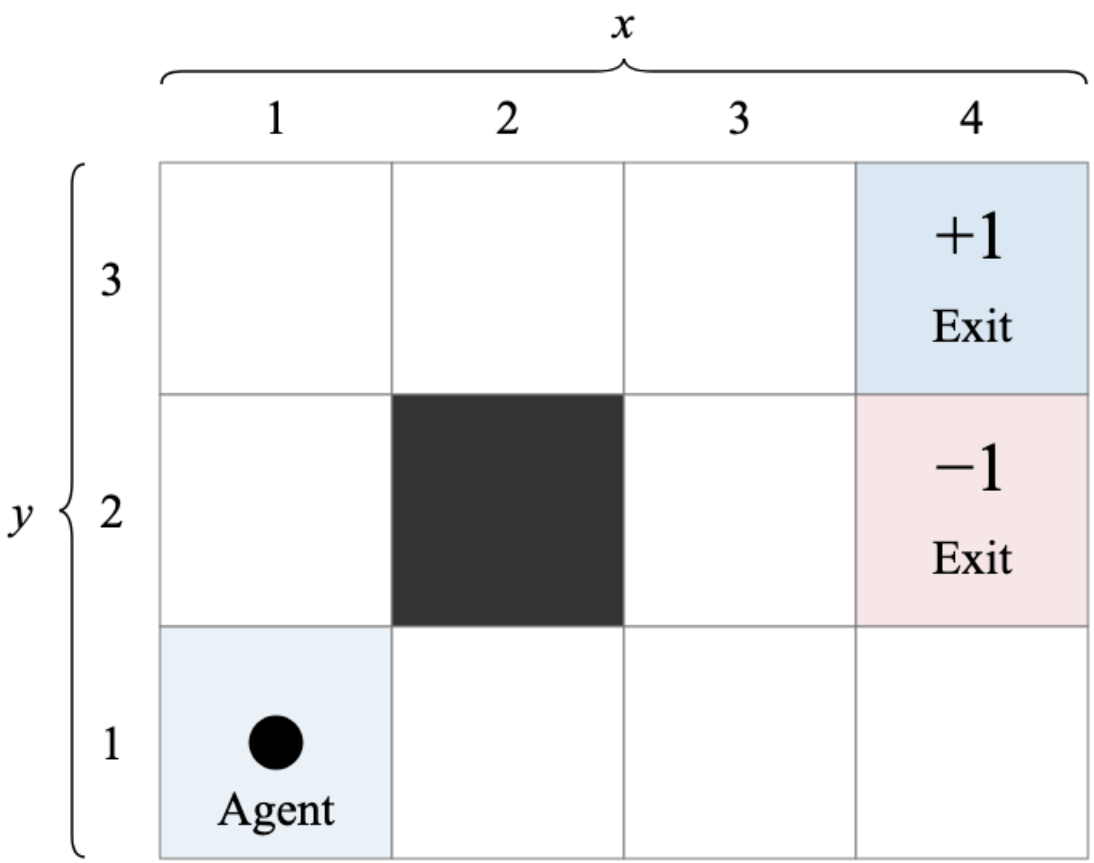
Defining a Markov Decision Process

$s \in S$	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 …

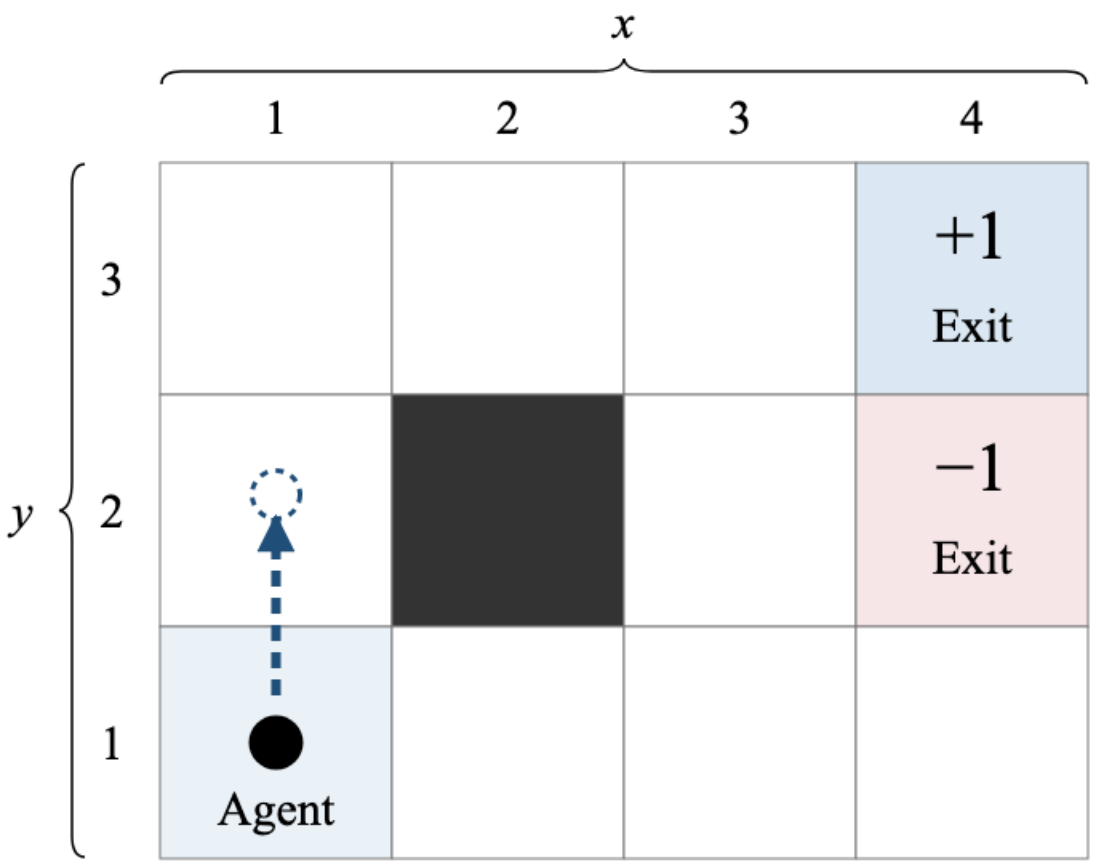


Start at (1, 1)

Follow a fixed list of actions

A Fixed Plan Cannot Respond to a Slip

Plan 1. North 2. North 3. East …



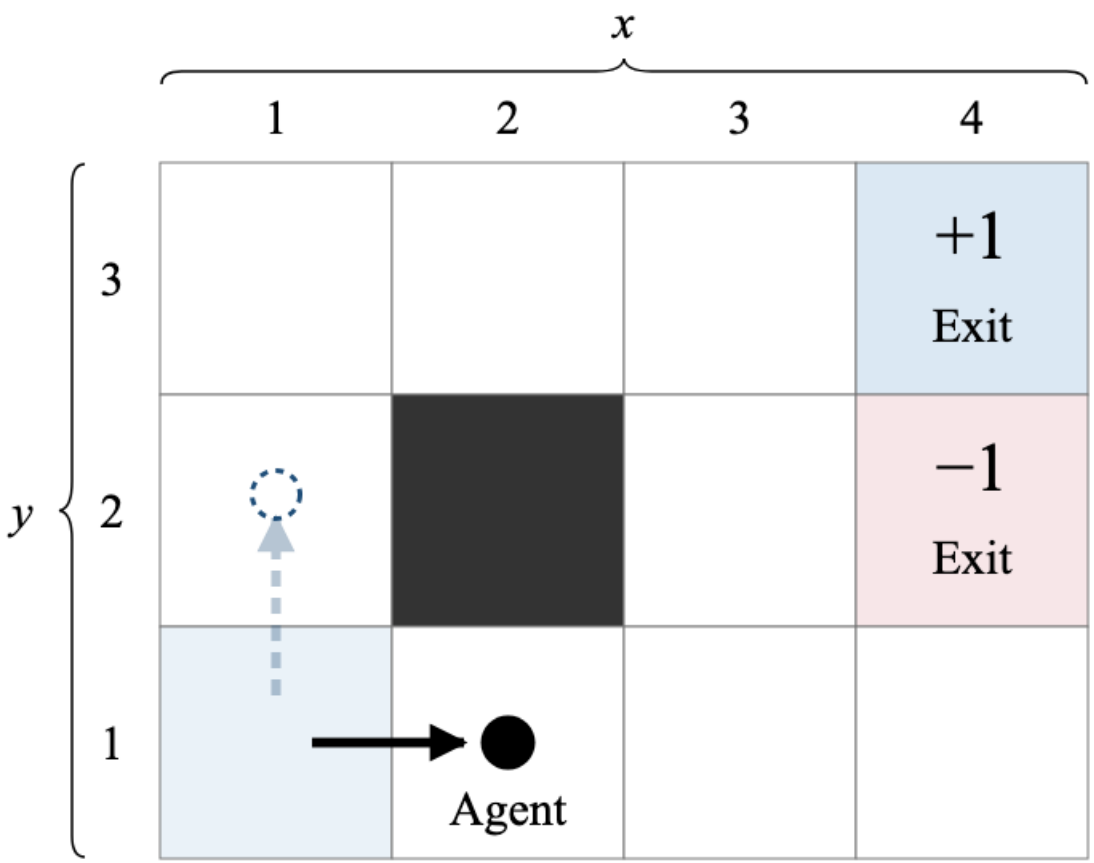
First action: North

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

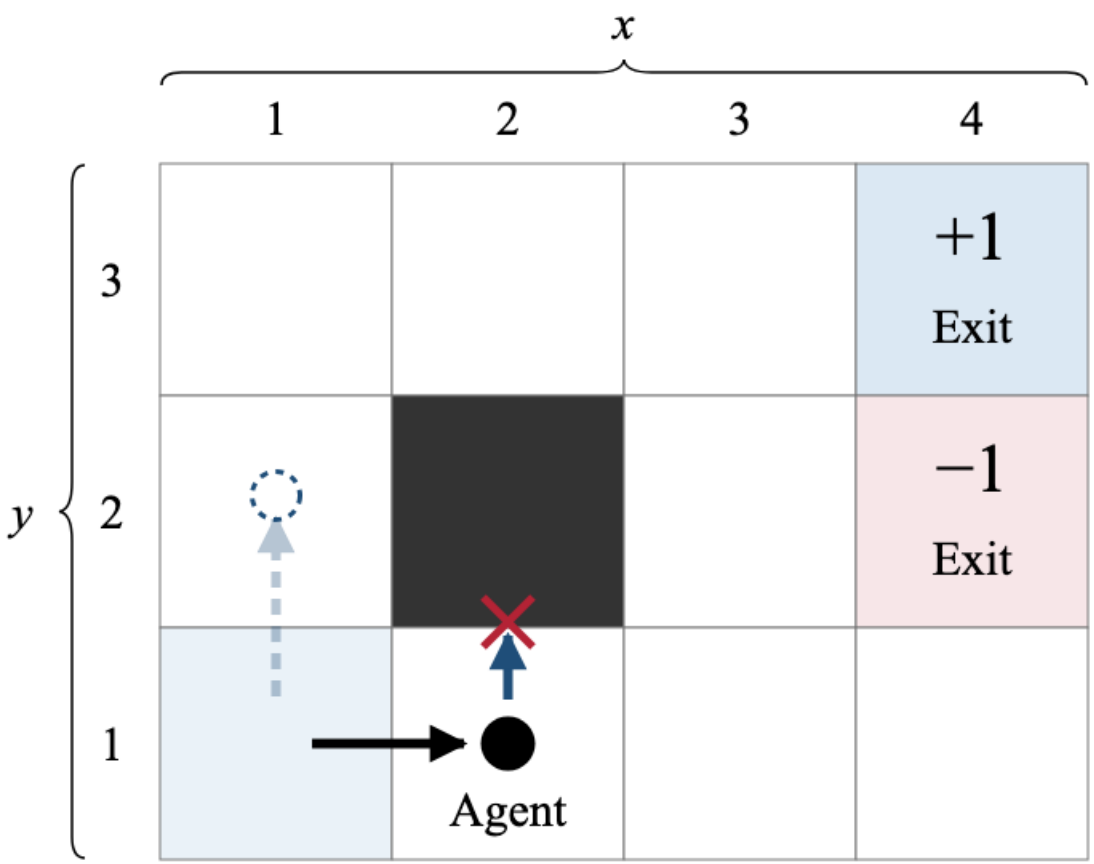
Actual move: East

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 …



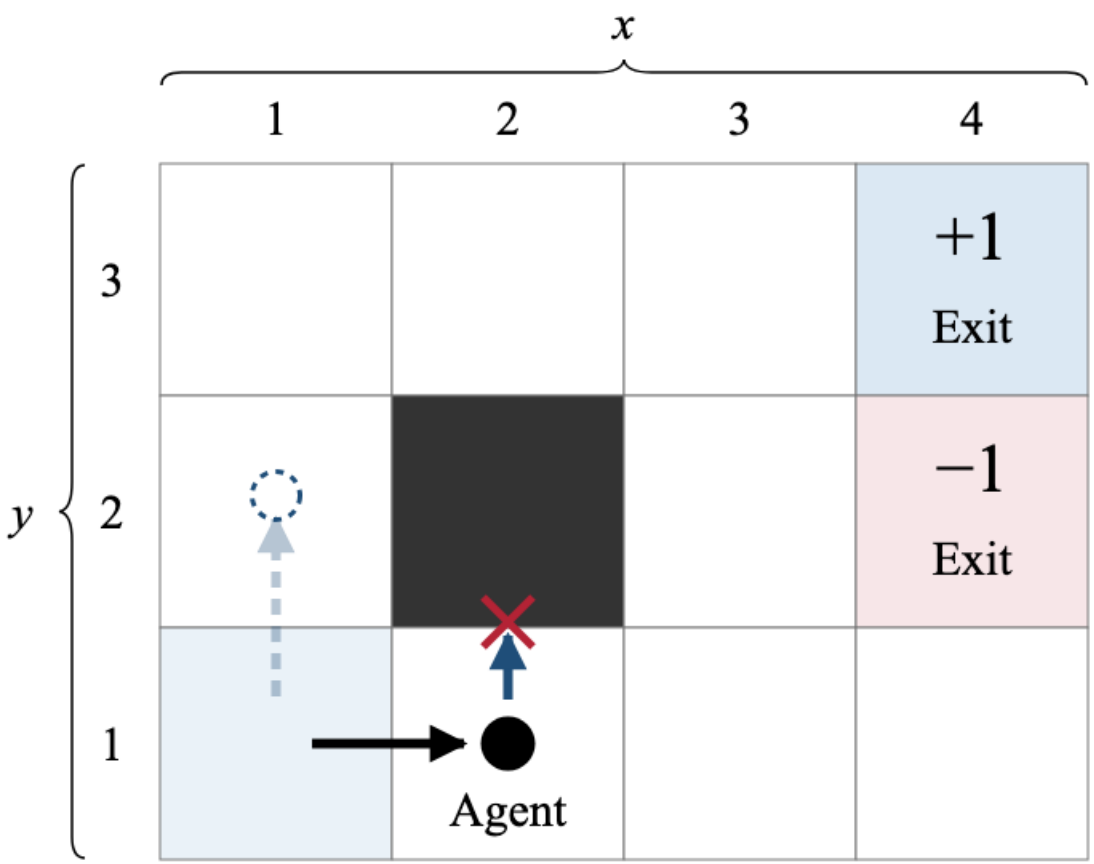
Next on the list: North

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 …

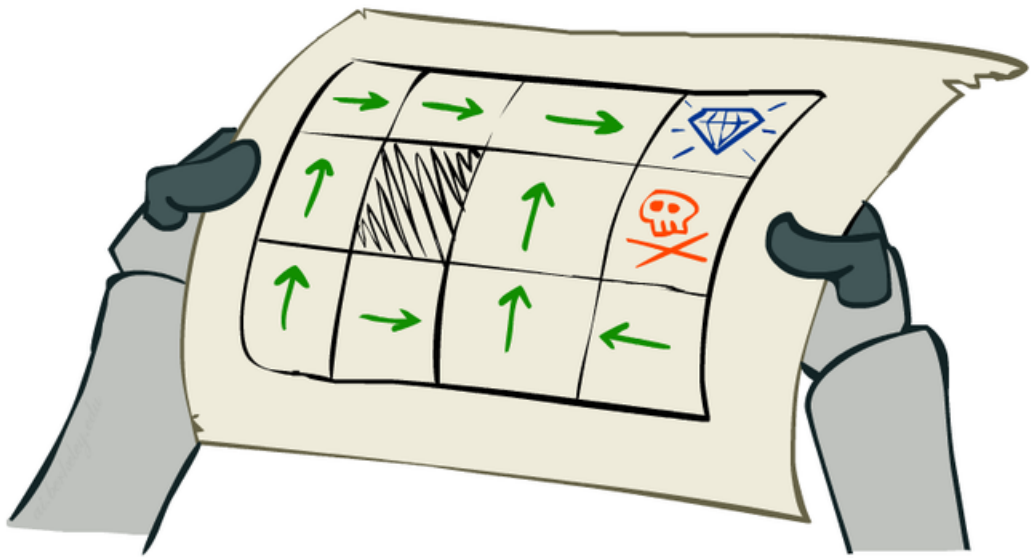


The plan still says North

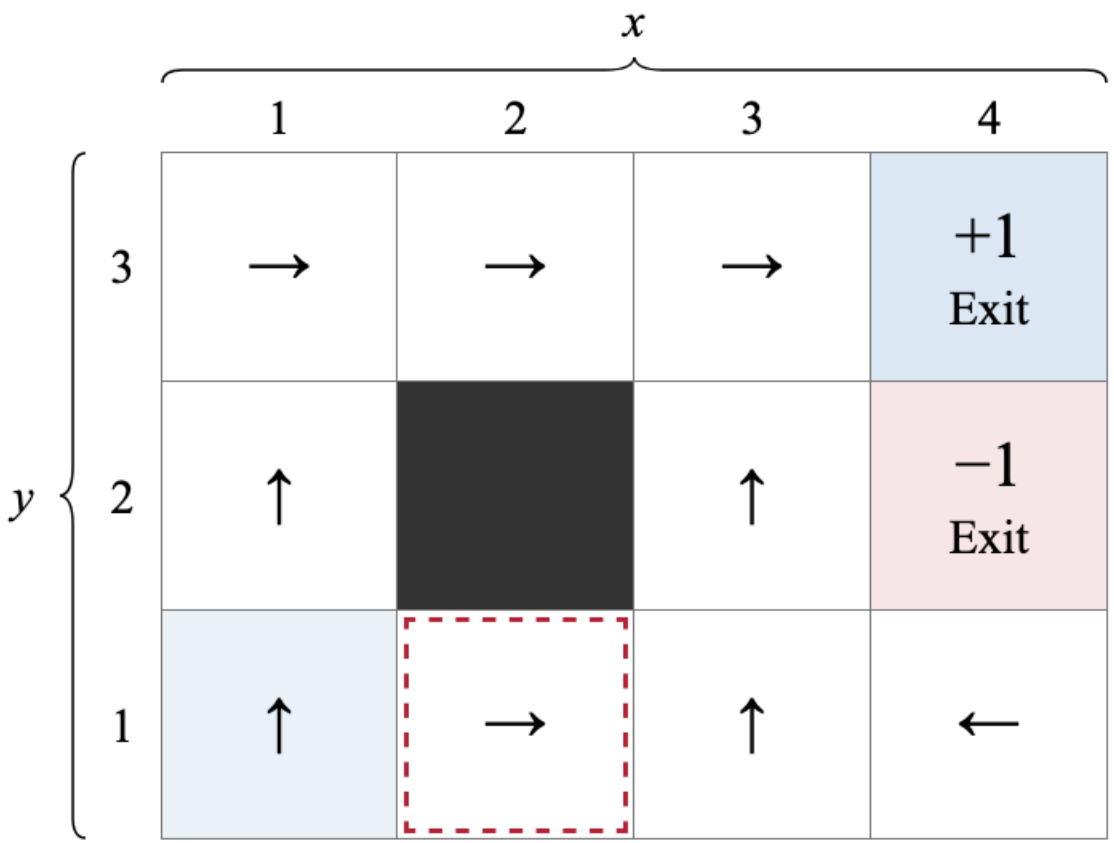
But we are in a different square

We need an action for where we are now

An MDP Solution Is a Policy

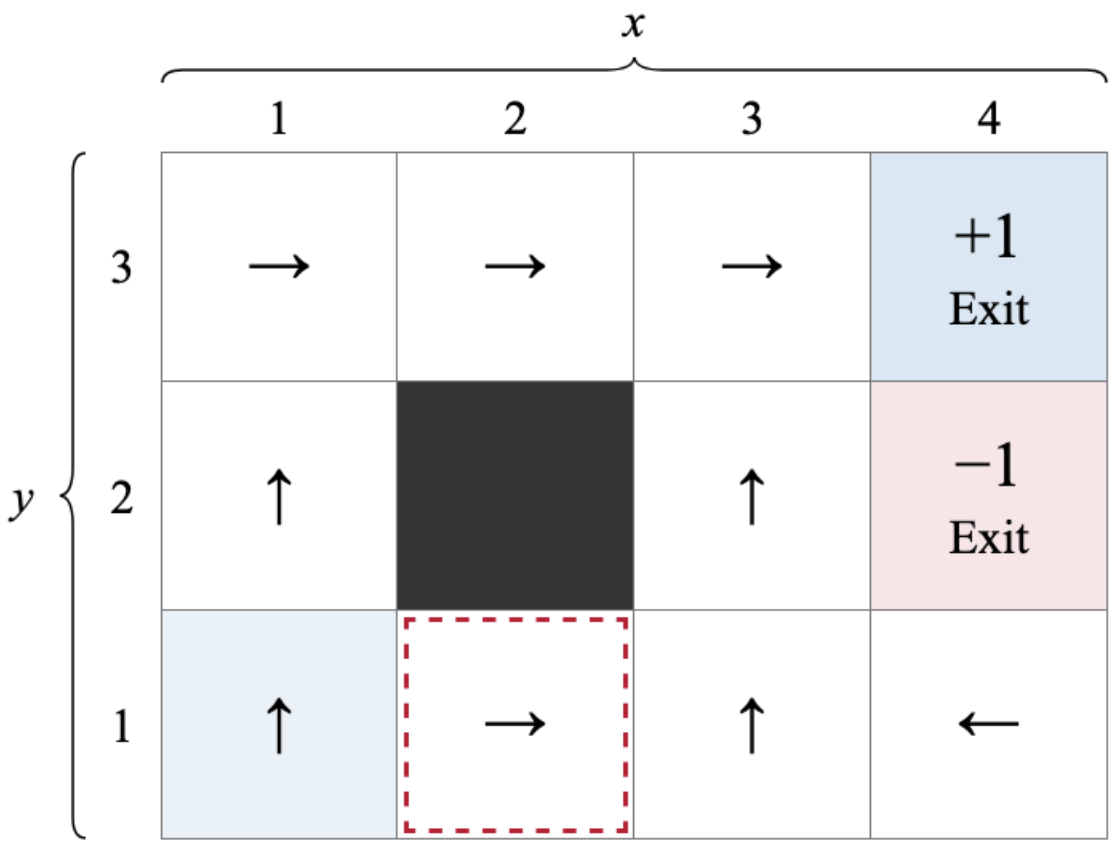


An MDP Solution Is a Policy



A policy π determines what to do for each state

An MDP Solution Is a Policy

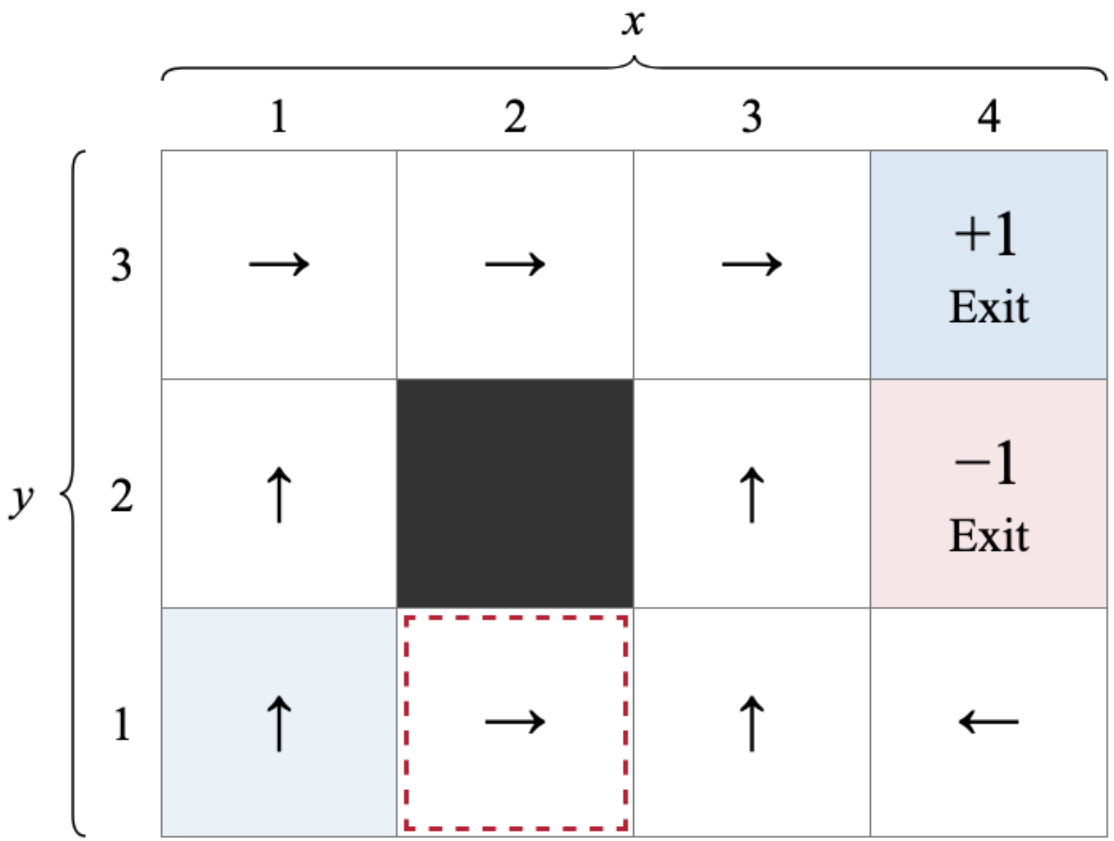


A policy π determines what to do for each state

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)

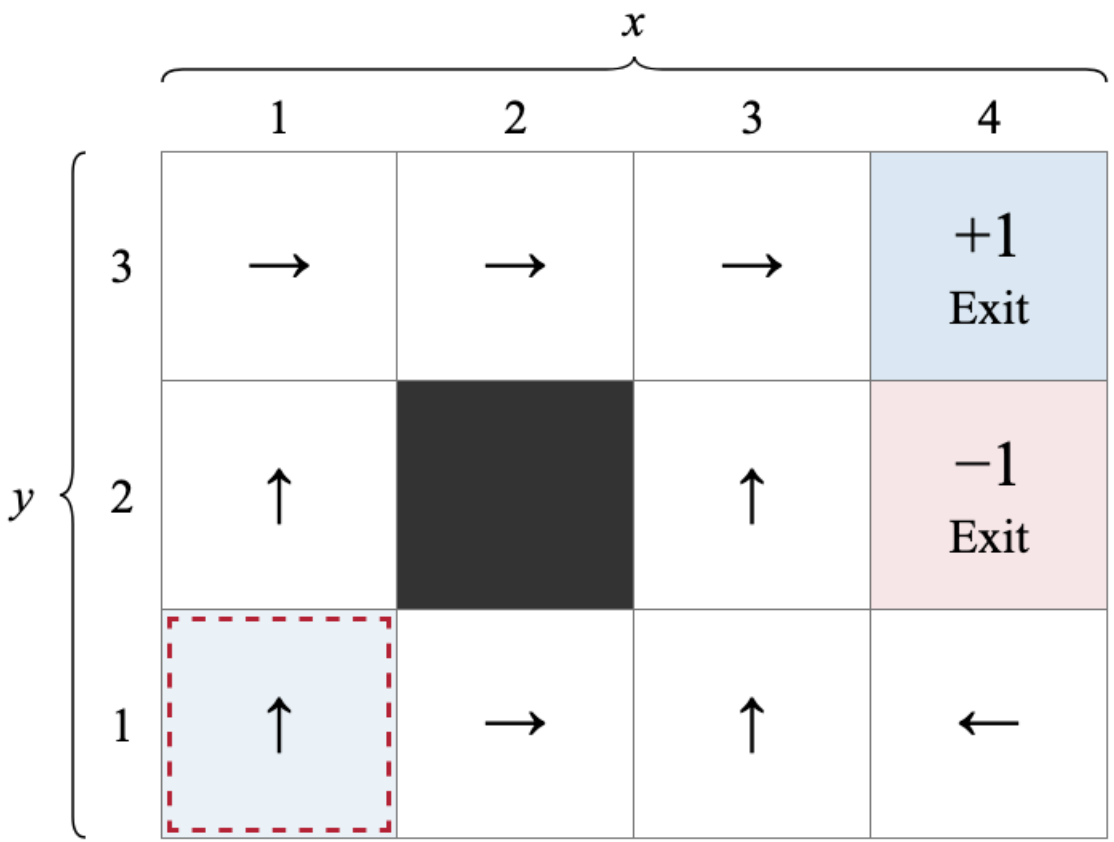
$$\pi(2, 1) = \text{East}$$

Observe the state, then use its action

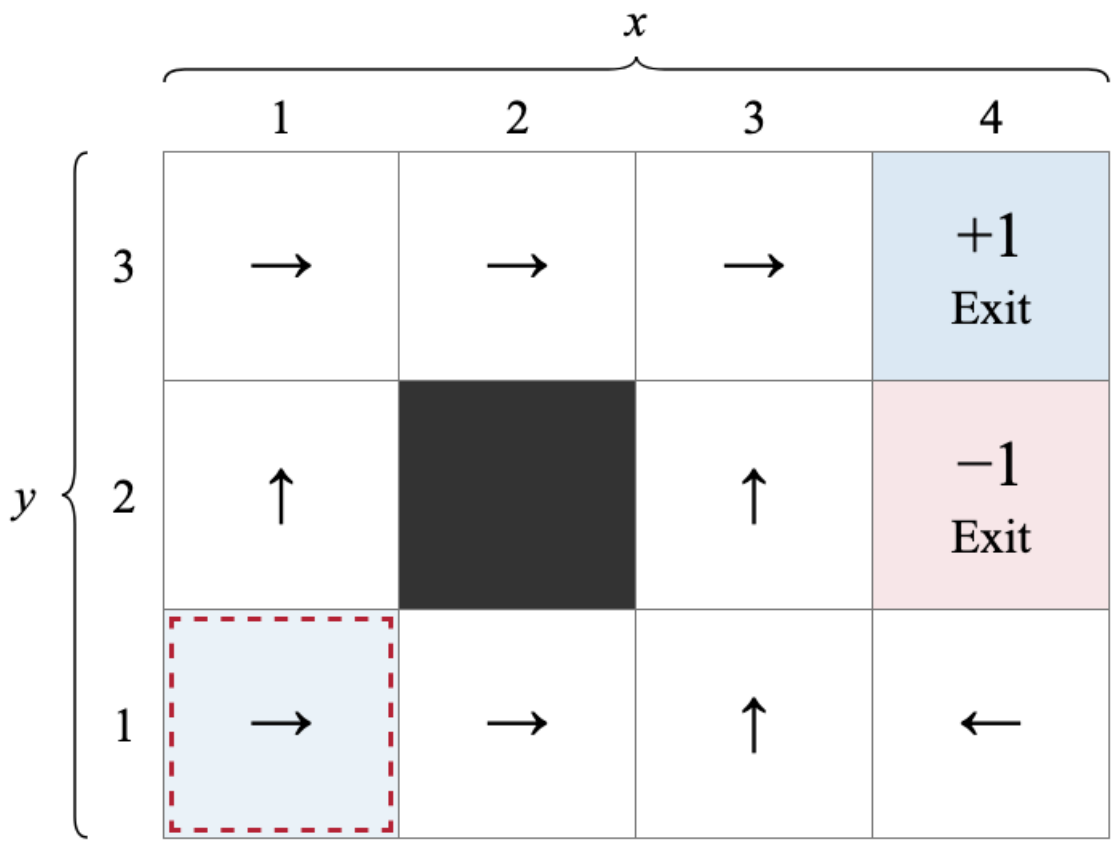
Which Policy Is Better?

Start: (1, 1). Move: −0.1. Exit: +1 or −1.
Intended direction: 0.8; each sideways direction: 0.1

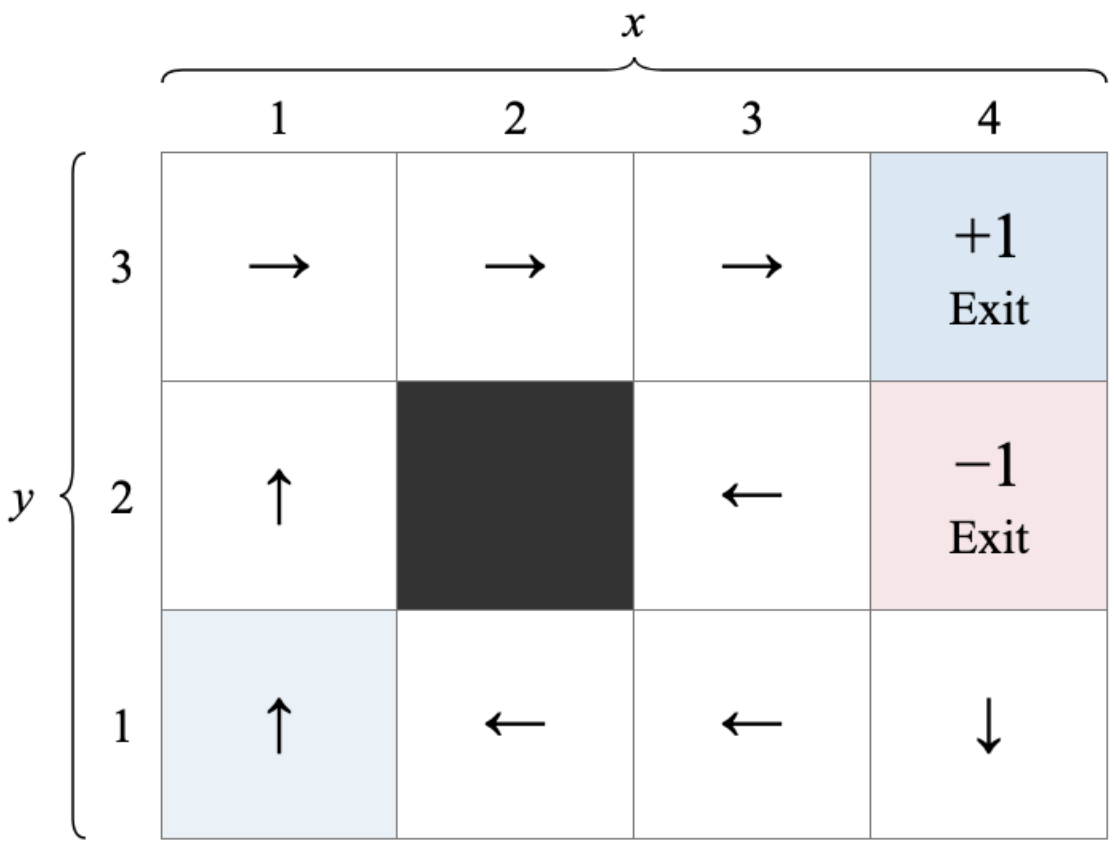
Policy A



Policy B



Living Rewards and the Optimal Policy



Living reward −.01

Reward received on every move

Intended direction: 0.8
Each sideways direction: 0.1
Exit rewards: +1 and −1

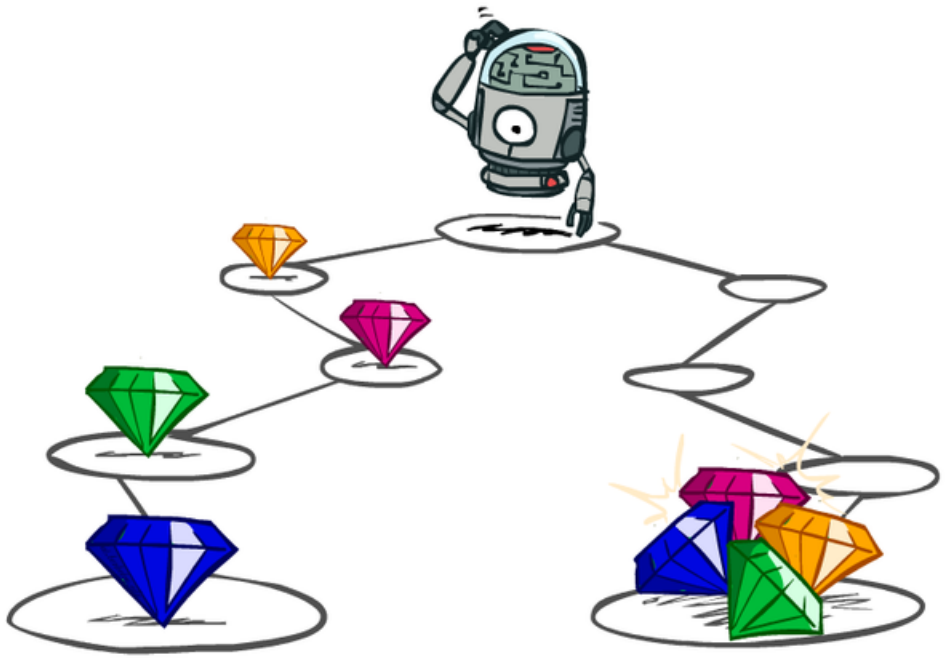
Quiz 2

Scan the QR Code to Open Ed

Answer the Questions

Share Any Questions You Have

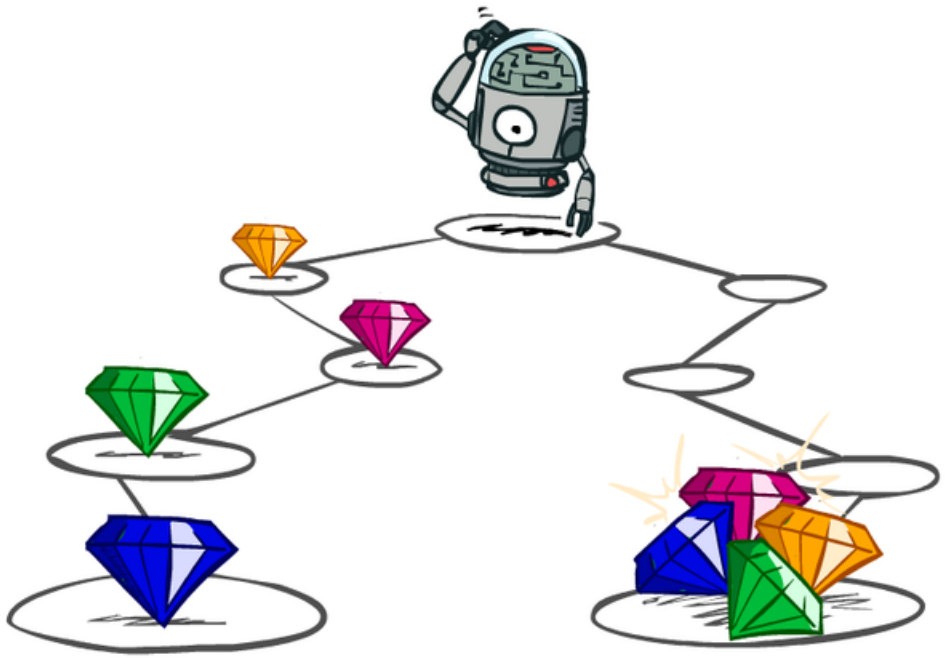
Comparing Reward Sequences



More reward?

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

Comparing Reward Sequences



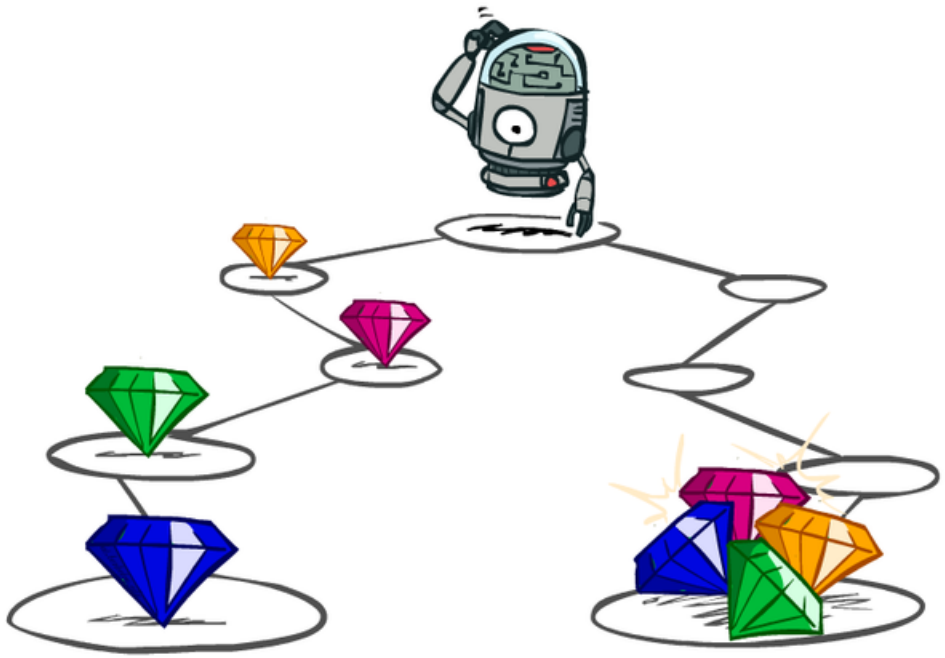
More reward?

$$[1, 2, 2] \quad \text{or} \quad [2, 3, 4]$$

Earlier reward?

$$[0, 0, 1] \quad \text{or} \quad [1, 0, 0]$$

Comparing Reward Sequences



More reward?

$$[1, 2, 2] \quad \text{or} \quad [2, 3, 4]$$

Earlier reward?

$$[0, 0, 1] \quad \text{or} \quad [1, 0, 0]$$

Which sequence is better?

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

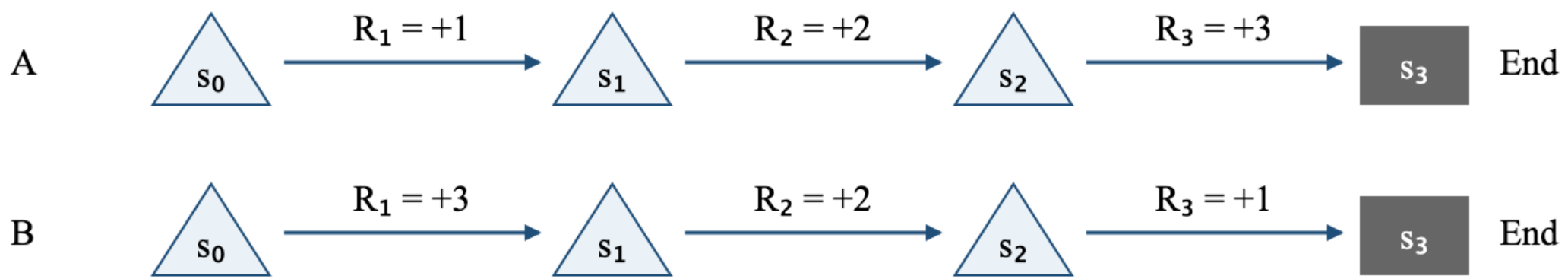
γ^2

Two steps later

Each extra step multiplies the reward's value by γ

$0 \leq \gamma \leq 1$

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

Reward	R_{t+1}	R_{t+2}	R_{t+3}

Discounted Return

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

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

Reward	R_{t+1}	R_{t+2}	R_{t+3}
Weight	1	γ	γ^2

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

Discounted Return

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

Reward	R_{t+1}	R_{t+2}	R_{t+3}
Weight	1	γ	γ^2

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

A smaller γ gives less weight to later rewards

Discounting a Business Decision

Up: cut maintenance now, save 50, then pay 1 per month

Down: upgrade equipment, invest 50, then save 1 per month

+50	-1	-1	-1	. . .	-1	-1	-1	-1
Start				. . .				
-50	+1	+1	+1	. . .	+1	+1	+1	+1

$$G_0(\text{Up}) = 50 - \gamma - \gamma^2 - \cdots - \gamma^{100}$$

$$G_0(\text{Down}) = -50 + \gamma + \gamma^2 + \cdots + \gamma^{100}$$

Illustrative warehouse plans; payoffs in $1,000s; 100 later months

Quiz 3

Scan the QR Code to Open Ed

Answer the Questions

Share Any Questions You Have

Decisions and Possible Returns

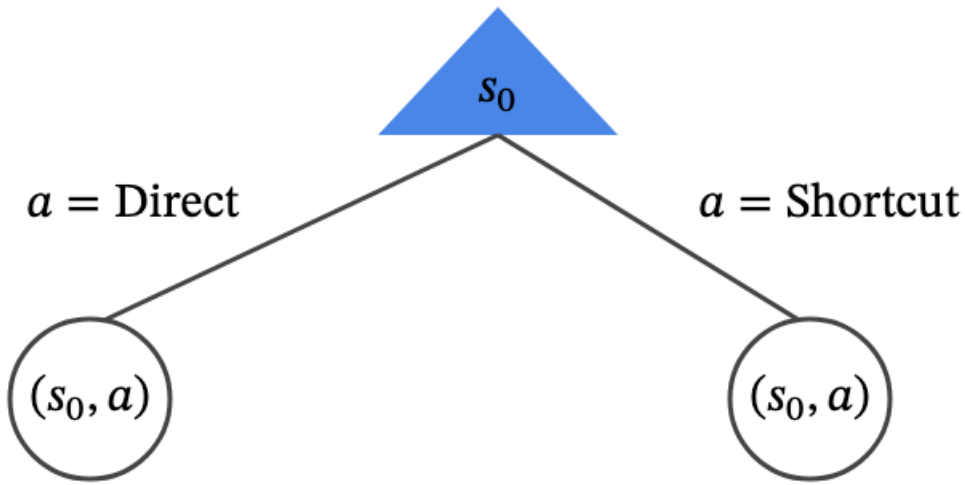
A Courier Chooses Direct or Shortcut

Triangle: state s

Action: a

Decisions and Possible Returns

A Courier Chooses Direct or Shortcut



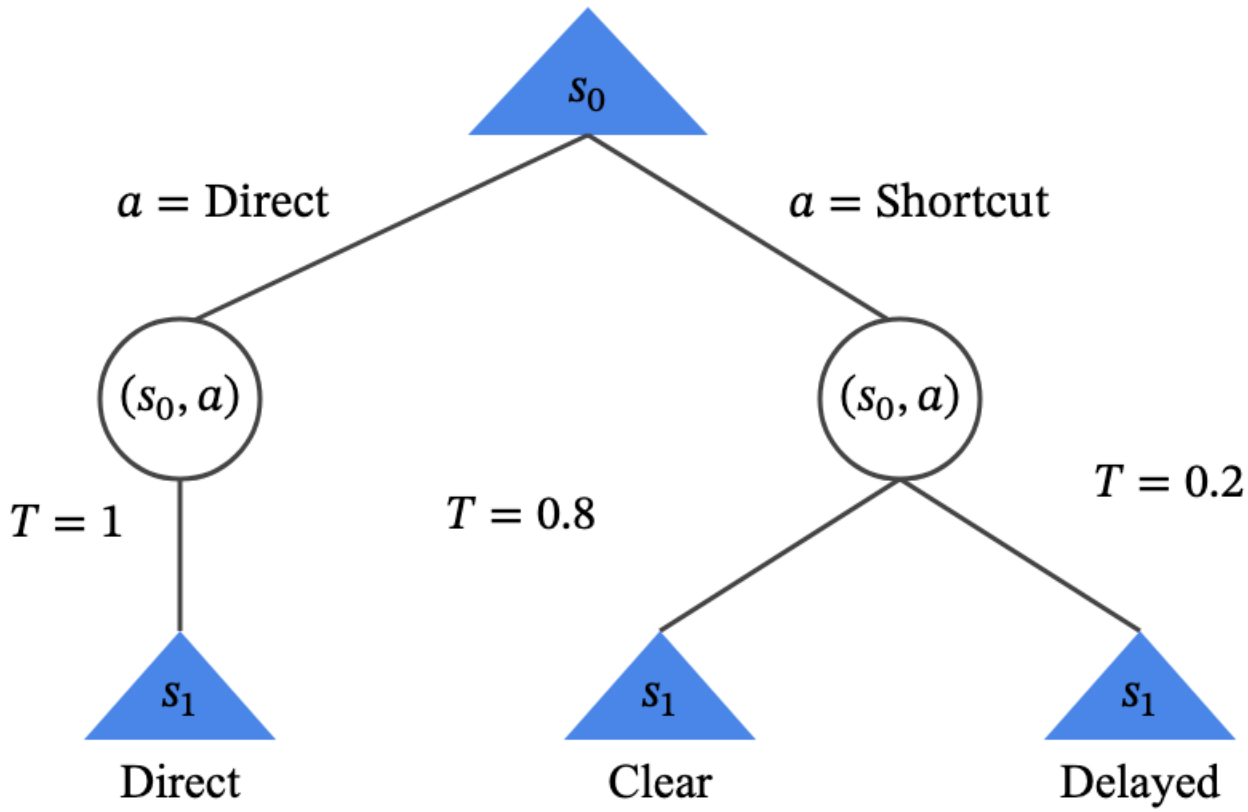
Triangle: state s

Action: a

Circle: state–action pair (s, a)

Decisions and Possible Returns

A Courier Chooses Direct or Shortcut



Triangle: state s

Action: a

Circle: state–action pair (s, a)

Branches: next states

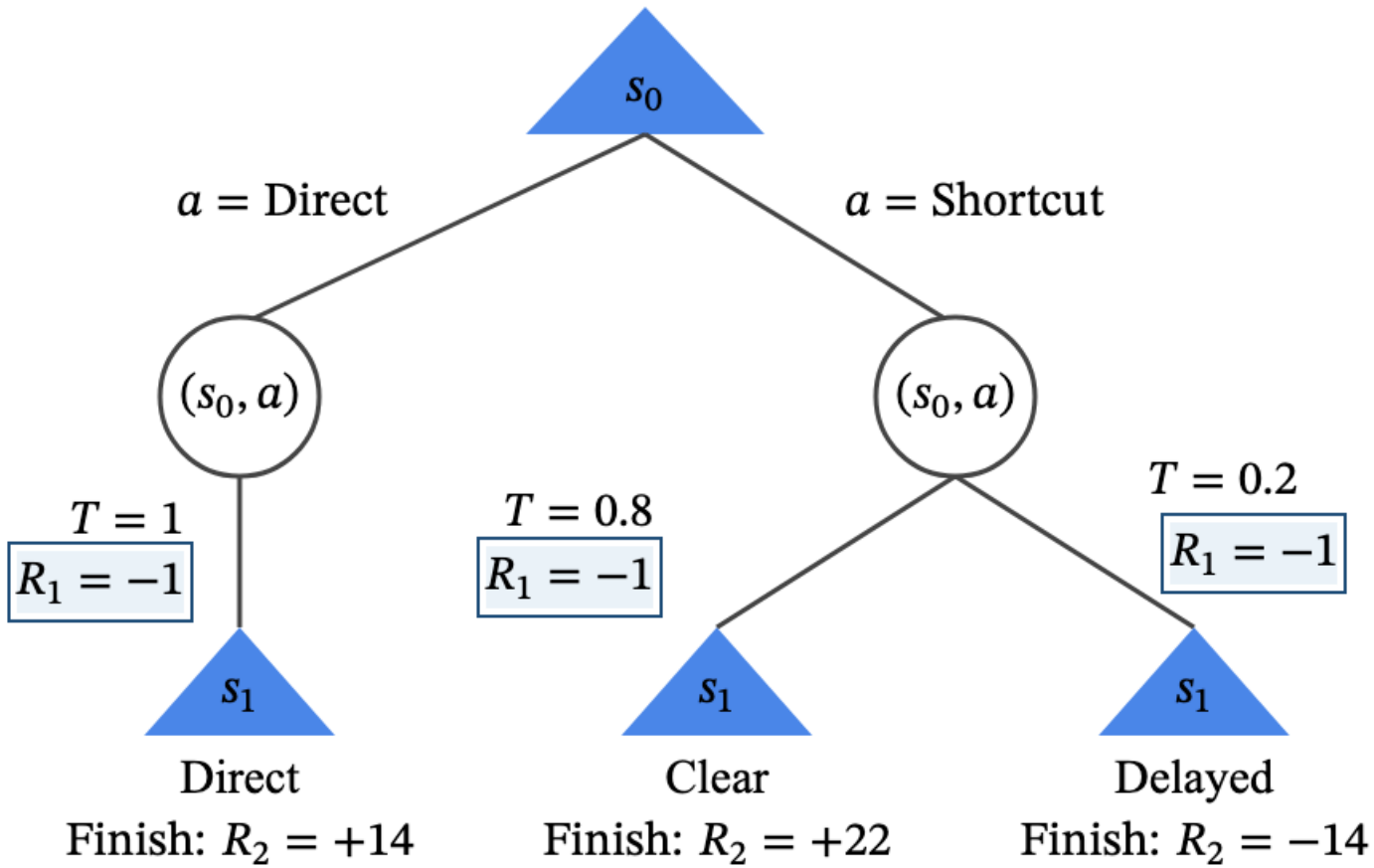
T: transition probability

Decisions and Possible Returns

A Courier Chooses Direct or Shortcut

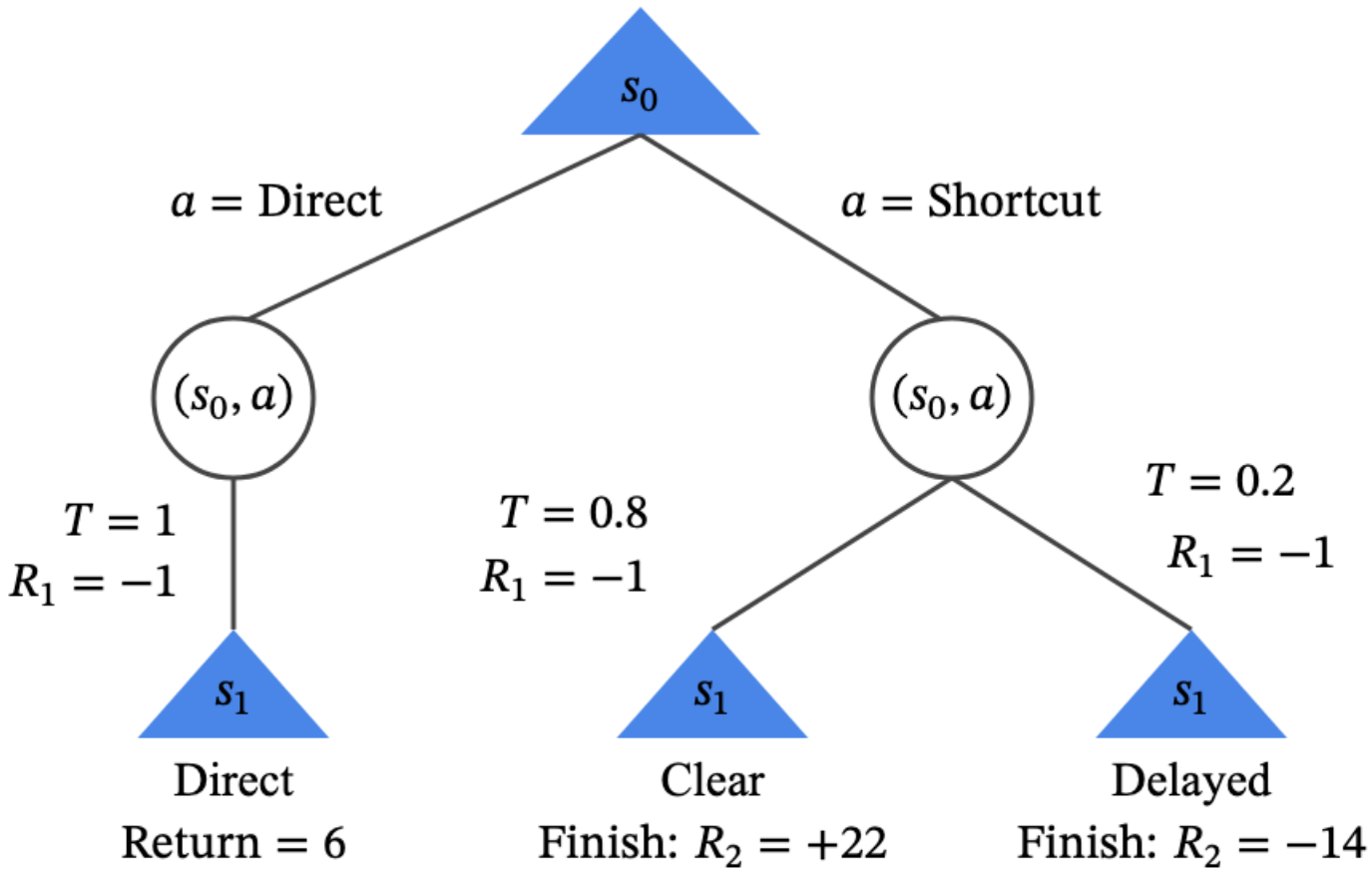
$$G_0 = R_1 + \gamma R_2$$

$$\gamma = 0.5$$



Decisions and Possible Returns

A Courier Chooses Direct or Shortcut



$$G_0 = R_1 + \gamma R_2$$

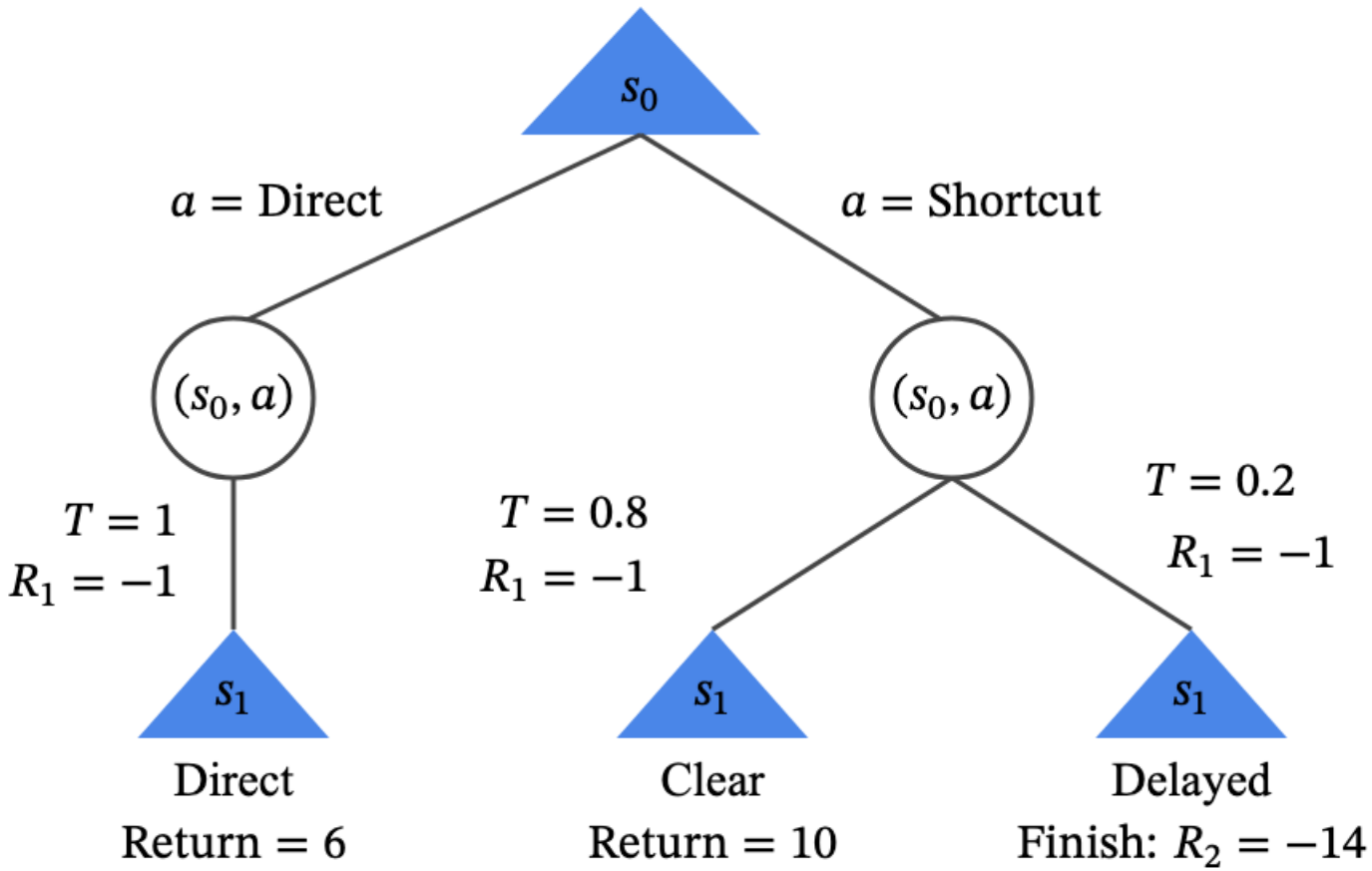
$$\gamma = 0.5$$

Direct

$$G_0 = -1 + 0.5(14) = 6$$

Decisions and Possible Returns

A Courier Chooses Direct or Shortcut



$$G_0 = R_1 + \gamma R_2$$

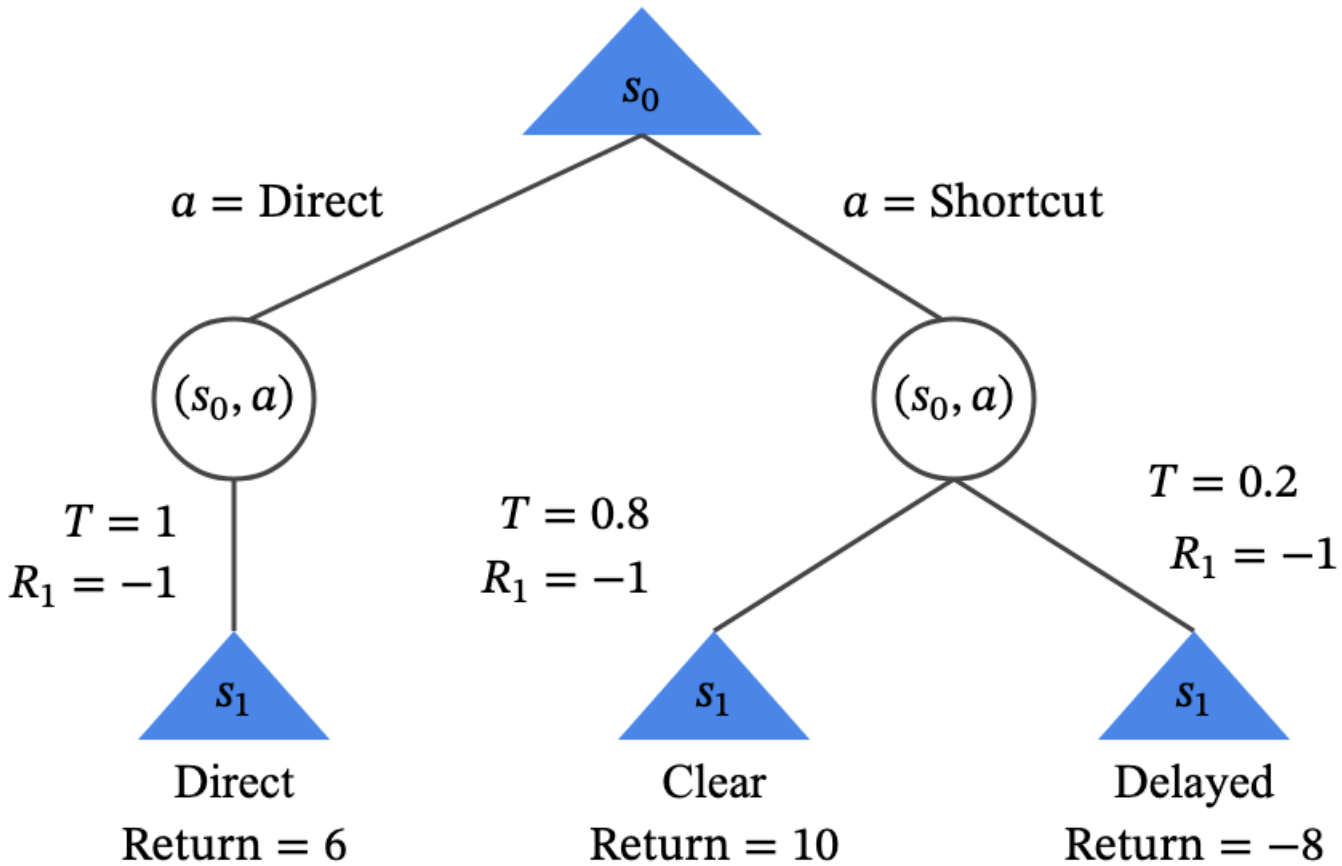
$$\gamma = 0.5$$

Shortcut, clear

$$G_0 = -1 + 0.5(22) = 10$$

Decisions and Possible Returns

A Courier Chooses Direct or Shortcut



$$G_0 = R_1 + \gamma R_2$$

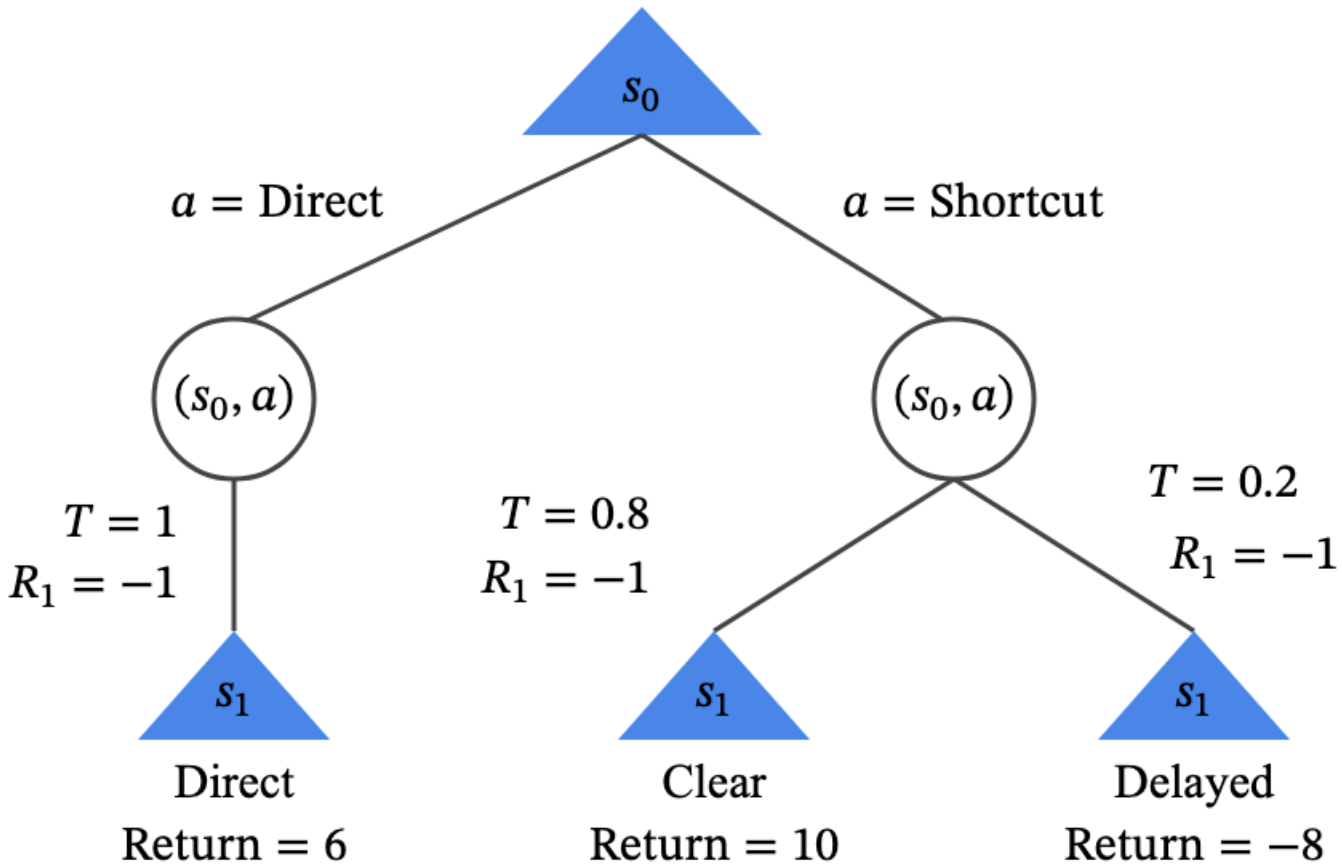
$$\gamma = 0.5$$

Shortcut, delayed

$$G_0 = -1 + 0.5(-14) = -8$$

Decisions and Possible Returns

A Courier Chooses Direct or Shortcut



$$G_0 = R_1 + \gamma R_2$$

$$\gamma = 0.5$$

Starting at s_0, what return should we expect if we choose Shortcut?

Expectimax: Average, Then Choose

At the circle, average the possible returns G_0

$$\text{Shortcut: } 0.8(10) + 0.2(-8) = 6.4$$

$$\text{Direct: } 1(6) = 6$$

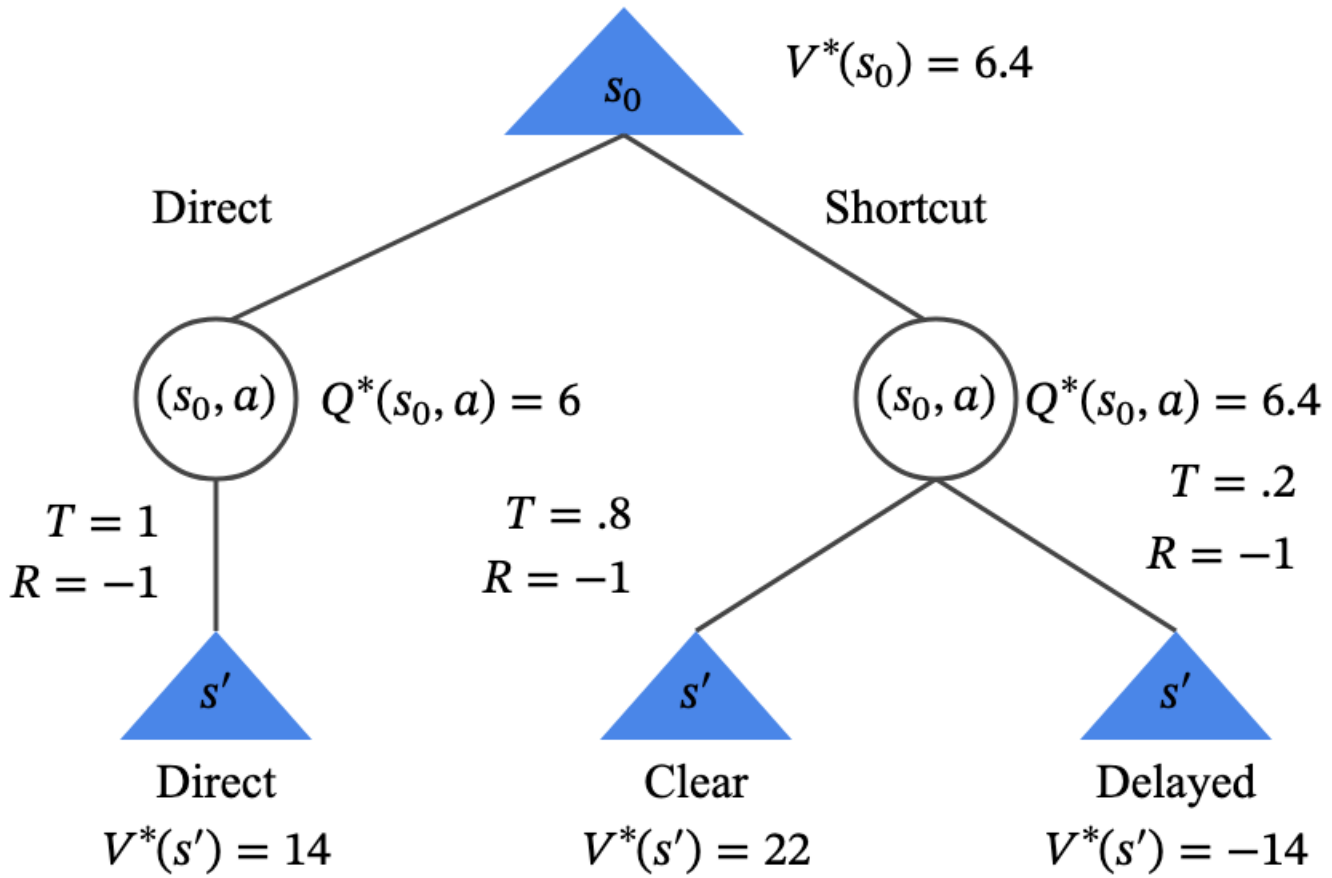
At the triangle, choose the higher expected return

$$\max(6, 6.4) = 6.4$$

Choose Shortcut under these assumptions

The Bellman Equation

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



Q^*: action value

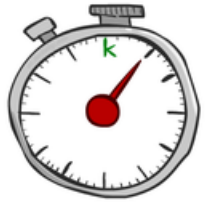
V^*: state value

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

Star means optimal

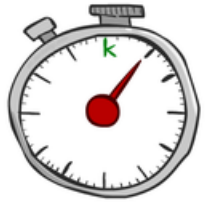
$\gamma = 0.5$

Time-Limited Values



- Define $V_k(s)$ as the optimal value of state s
  if the game ends in k more steps (k more rewards)

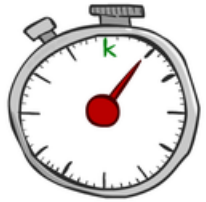
Time-Limited Values



- Define $V_k(s)$ as the optimal value of state s
  if the game ends in k more steps (k more rewards)

- Equivalently, look ahead k steps with expectimax

Time-Limited Values



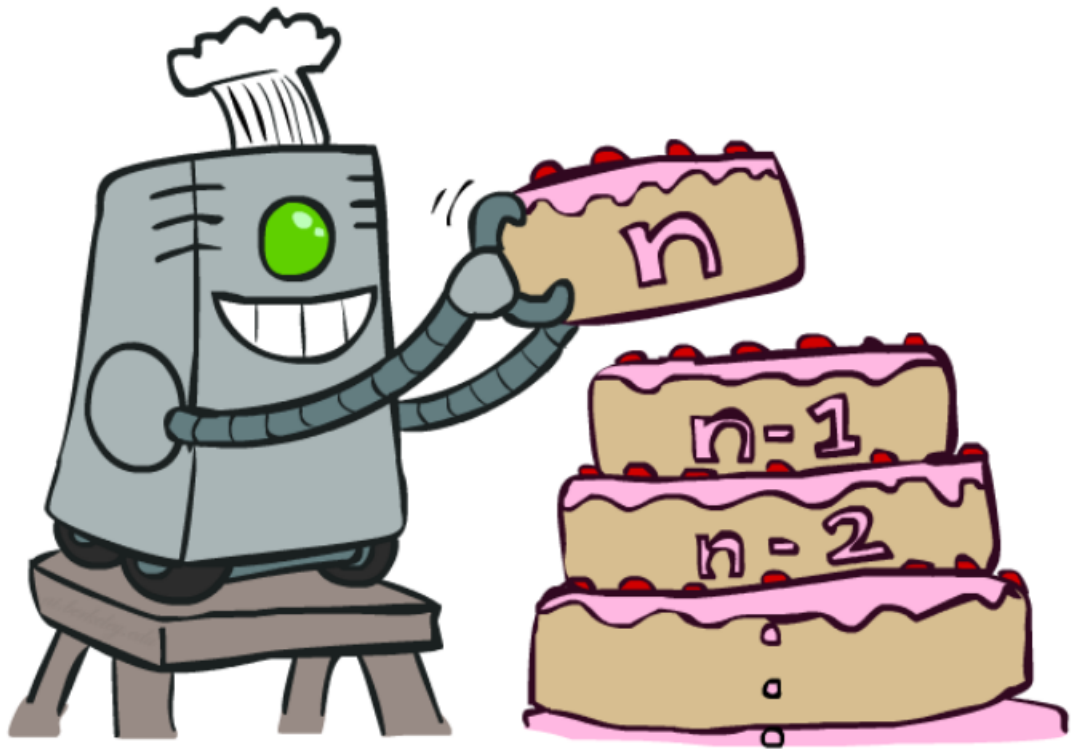
- Define $V_k(s)$ as the optimal value of state s
  if the game ends in k more steps (k more rewards)

- Equivalently, look ahead k steps with expectimax

$$V_0(s) = 0$$

No steps left, no more rewards

Value Iteration



$$V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow \cdots$$

Each sweep uses the previous values

Value Iteration: Propagating Rewards

For this calculation: movement reward 0, $\gamma = 0.99$

Start with $V_0(s) = 0$ for every state

One sweep: exit values become $+1$ and -1

Second sweep, at (3, 3), choose East:

$$0 + 0.99\,[0.8(1) + 0.1(0) + 0.1(0)] = 0.792$$

Read the previous values everywhere before writing the new ones

Intended direction probability 0.8; each sideways direction 0.1

Value Iteration: Key Ideas

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

- Use a known model: transition probabilities T and rewards R
- Start at $V_0 = 0$; use V_k throughout the sweep to calculate V_{k+1}
- Average over outcomes, then choose the largest expected return
- Repeat until values change very little; choose actions using those values

Planning with a Known Model

For continuing tasks with bounded rewards: $0 \leq \gamma < 1$

Quiz 4

Scan the QR Code to Open Ed

Answer the Questions

Share Any Questions You Have

The Two-Armed Bandit

Each play returns to the same decision state; $\gamma = 0.9$

Blue	Red
\$1 every time	\$2 with probability 0.75; otherwise \$0
Expected reward: \$1	Expected reward: \$1.50

$$V_{k+1}(s) = \max\{1 + 0.9V_k(s),\ 1.5 + 0.9V_k(s)\}$$

Known odds: use value iteration

Unknown odds: learn from observed rewards

Value Iteration Needs a Model

Next-State Probabilities and Rewards

The Gridworld Problem

Move North

80% Intended Direction

10% Each Side · Rewards Given

Model Supplied

Value Iteration Needs a Model

Next-State Probabilities and Rewards

The Gridworld Problem	The Crawler Problem	The LinkedIn Problem
Move North	Move a Joint	Send or Wait
80% Intended Direction	Distance Moved: ?	Visit Probability: ?
10% Each Side · Rewards Given	Movement Reward Must Be Learned	Later Responses Are Also Unknown
Model Supplied	Model Not Supplied	Model Not Supplied

Value Iteration Needs a Model

Next-State Probabilities and Rewards

The Gridworld Problem	The Crawler Problem	The LinkedIn Problem
Move North	Move a Joint	Send or Wait
80% Intended Direction	Distance Moved: ?	Visit Probability: ?
10% Each Side · Rewards Given	Movement Reward Must Be Learned	Later Responses Are Also Unknown
Model Supplied	Model Not Supplied	Model Not Supplied

Missing Model Inputs → Cannot Compute the Backup Yet

If We Learn a Model, We Can Use Value Iteration

Learn from Experience

Current Situation → Action → Reward and Next Situation

The Gridworld Problem	The Crawler Problem	The LinkedIn Problem
Move North	Move a Joint	Send or Wait

Learn from Experience

Current Situation → Action → Reward and Next Situation

The Gridworld Problem	The Crawler Problem	The LinkedIn Problem
Move North	Move a Joint	Send or Wait
↓	↓	↓
Next Square	New Pose	Member Response
and Reward	and Distance Moved	and Updated Activity History

Learn from Experience

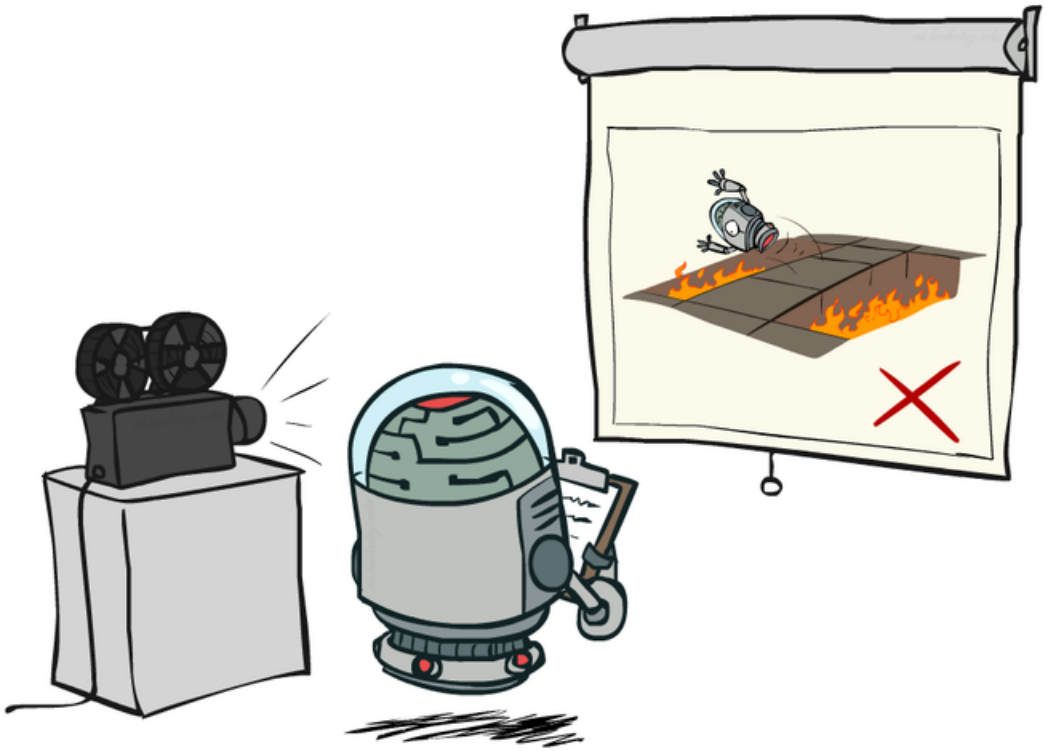
Current Situation → Action → Reward and Next Situation

The Gridworld Problem	The Crawler Problem	The LinkedIn Problem
Move North	Move a Joint	Send or Wait
↓	↓	↓
Next Square	New Pose	Member Response
and Reward	and Distance Moved	and Updated Activity History

Learn State Values from Experience

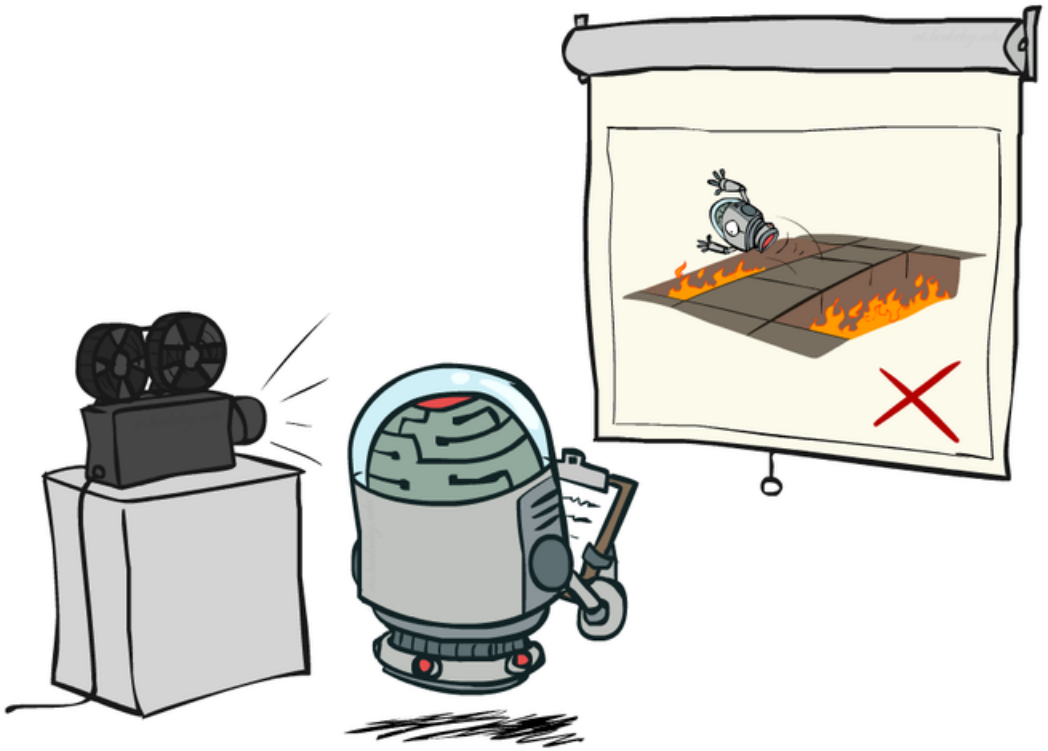
How Much Reward Does a Fixed Policy Earn?

Passive Reinforcement Learning



A Fixed Policy Chooses the Actions

Passive Reinforcement Learning



A Fixed Policy Chooses the Actions

Learn How Good That Policy Is

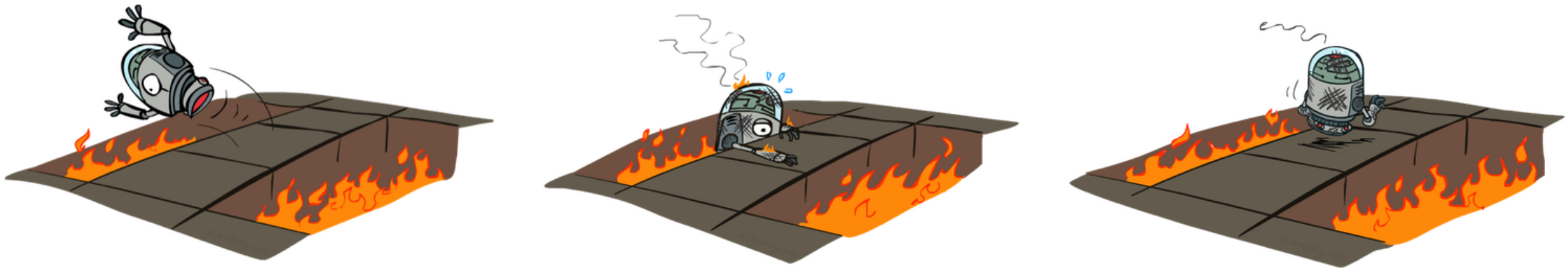
Learning About a Poor Policy

What If the Given Policy Makes Poor Choices?

Learning About a Poor Policy

What If the Given Policy Makes Poor Choices?

Learning About a Poor Policy



What If the Given Policy Makes Poor Choices?

Learn Its Value While Keeping Its Actions Fixed

Learning a Fixed Policy's Values

Given Policy: B → East → C → South → E → Exit

Moves Earn −2; Exit at E Earns −4

Each Number Estimates the Return from Following This Policy

Learn the Values, Keep the Policy Fixed

Passive Learning in Gridworld

The Same Gridworld, with a Supplied Policy

Follow the Arrows and Observe the Outcomes

Repeated Experience Refines the State-Value Estimates

Learn the Values, Keep the Policy Fixed

Manual Learning in Gridworld

Start with Zero Estimates

Choose Actions with the Keyboard

Watch the Estimates Change with Experience

Changing Our Actions Changes What We Estimate

Evaluating the Bridge Policy

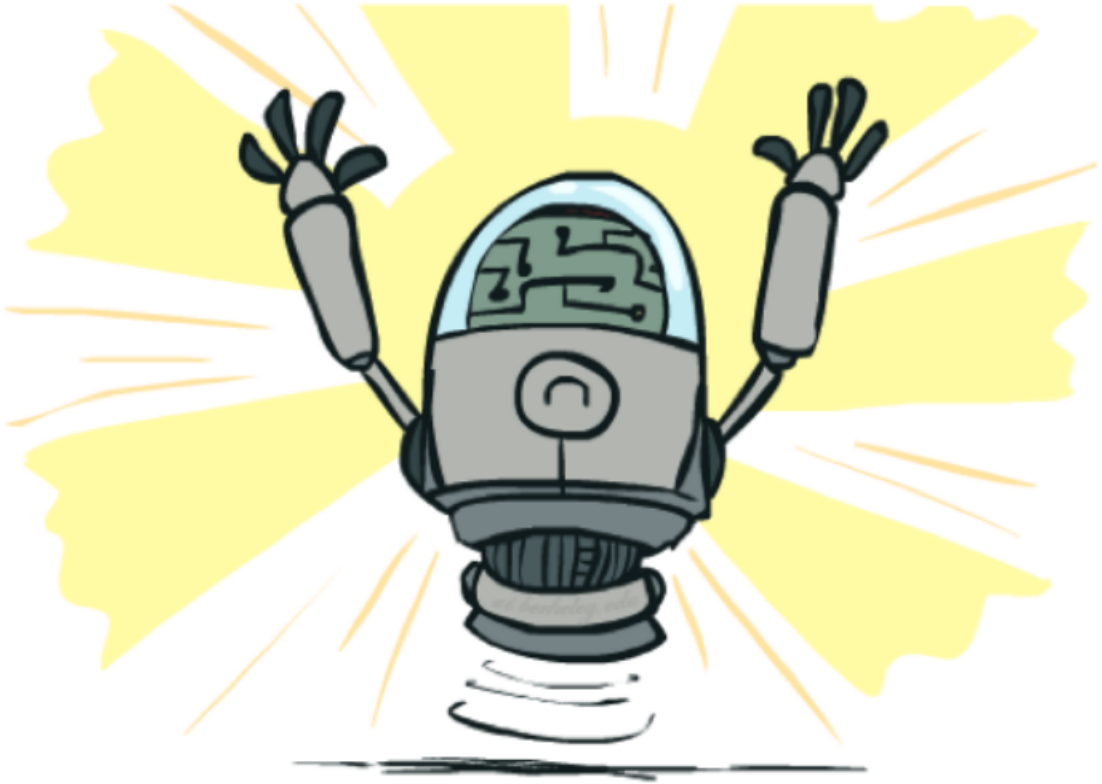
Given Policy: Move East Along the Bridge

Moves Can Slip Into a Pit

How Much Reward Does This Policy Earn?

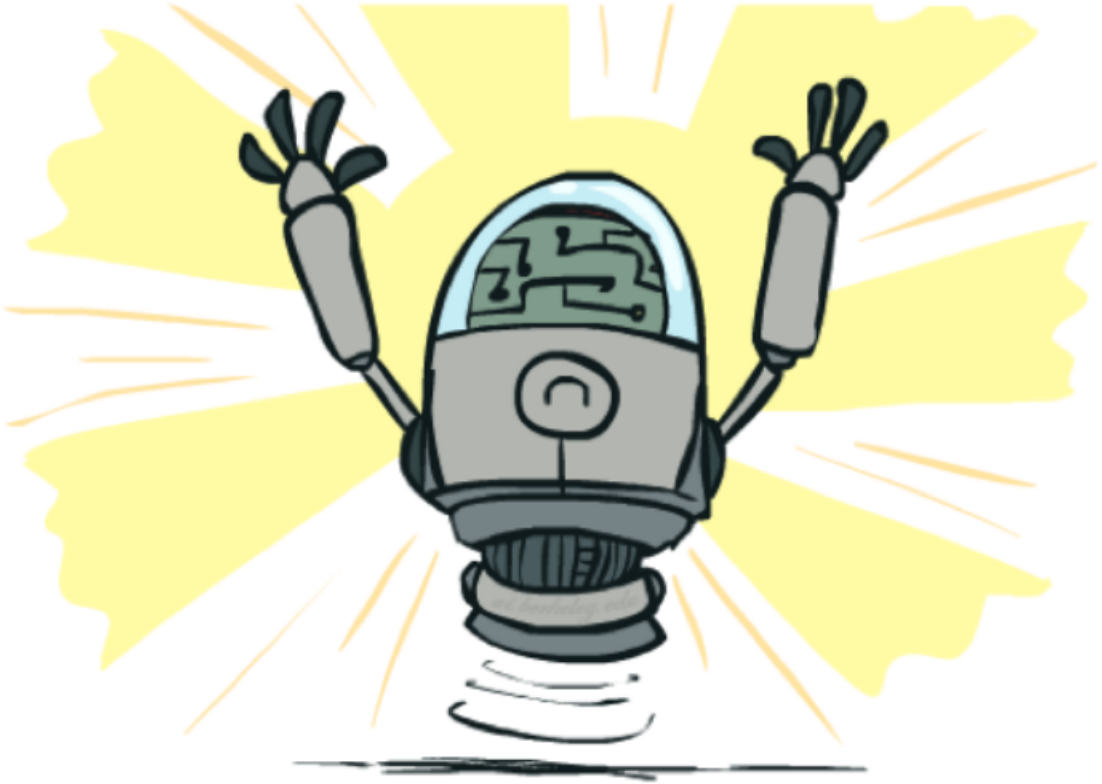
Learn the Values, Keep the Policy Fixed

Next: Active Reinforcement Learning



Today: Evaluate a Given Policy

Next: Active Reinforcement Learning



Today: Evaluate a Given Policy

Next Time: Learn Better Actions

Quiz 5

Scan the QR Code to Open Ed

Answer the Questions

Share Any Questions You Have

Exit Ticket

Scan the QR Code to Open Ed

Supervised, Unsupervised, or RL?

Explain Why for Your Project