# Decision Processes

## CISC 7404 - Decision Making

Steven Morad

University of Macau

# Review

# Markov Processes

# Markov Processes

Decisions must make some change in the world

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

**Markov processes** are a popular way to model the world

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

**Markov processes** are a popular way to model the world

Some things we can model using Markov processes:

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

**Markov processes** are a popular way to model the world

Some things we can model using Markov processes:
- Music

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

**Markov processes** are a popular way to model the world

Some things we can model using Markov processes:
- Music
- DNA sequences

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

**Markov processes** are a popular way to model the world

Some things we can model using Markov processes:
- Music
- DNA sequences
- Cryptography

# Markov Processes

Decisions must make some change in the world

If they make no change, they do not matter, and are not decisions!

Before we look at decision making, let us think about how we model change in the world

**Markov processes** are a popular way to model the world

Some things we can model using Markov processes:
- Music
- DNA sequences
- Cryptography
- History

# Markov Processes

We can model almost anything as a Markov process

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

It is a probabilistic model of dynamical systems that allows us to predict the future

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

It is a probabilistic model of dynamical systems that allows us to predict the future

A Markov process consists of two parts

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

It is a probabilistic model of dynamical systems that allows us to predict the future

A Markov process consists of two parts

The **state space**

$$S$$

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

It is a probabilistic model of dynamical systems that allows us to predict the future

A Markov process consists of two parts

The **state space**

$$S$$

Outcome space describing the state of our system

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

It is a probabilistic model of dynamical systems that allows us to predict the future

A Markov process consists of two parts

The **state space**

$$S$$

Outcome space describing the state of our system

The **state transition function**

$$\mathrm{Tr} : S \mapsto \Delta S$$

# Markov Processes

We can model almost anything as a Markov process

So what is a Markov process?

It is a probabilistic model of dynamical systems that allows us to predict the future

A Markov process consists of two parts

The **state space**

$$S$$

Outcome space describing the state of our system

The **state transition function**

$$\mathrm{Tr} : S \mapsto \Delta S$$

$$\mathrm{Tr}(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t)$$

$$s_{t+1} \sim \mathrm{Tr}(\cdot \mid s_t)$$

# Markov Processes

**Problem:** Predict the weather

# Markov Processes

**Problem:** Predict the weather

$$S = \{\text{rain}, \text{cloud}, \text{sun}\} = \{R, C, S\}$$

# Markov Processes

**Problem:** Predict the weather

$$S = \{\text{rain}, \text{cloud}, \text{sun}\} = \{R, C, S\}$$

$$\text{Tr}(s_{t+1} \mid s_t) = \text{Pr}(s_{t+1} \mid s_t)$$

# Markov Processes

**Problem:** Predict the weather

$$S = \{\text{rain}, \text{cloud}, \text{sun}\} = \{R, C, S\}$$

$$\operatorname{Tr}(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t)$$

$$= \begin{bmatrix} \Pr(C \mid C) & \Pr(R \mid C) & \Pr(S \mid C) \\ \Pr(C \mid R) & \Pr(R \mid R) & \Pr(S \mid R) \\ \Pr(C \mid S) & \Pr(R \mid S) & \Pr(S \mid S) \end{bmatrix}$$

# Markov Processes

**Problem:** Predict the weather

$$S = \{\text{rain}, \text{cloud}, \text{sun}\} = \{R, C, S\}$$

$$\mathrm{Tr}(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t)$$

$$= \begin{bmatrix} \Pr(C \mid C) & \Pr(R \mid C) & \Pr(S \mid C) \\ \Pr(C \mid R) & \Pr(R \mid R) & \Pr(S \mid R) \\ \Pr(C \mid S) & \Pr(R \mid S) & \Pr(S \mid S) \end{bmatrix}$$

$$= \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.5 & 0.3 & 0.2 \\ 0.5 & 0.1 & 0.4 \end{bmatrix}$$

# Markov Processes

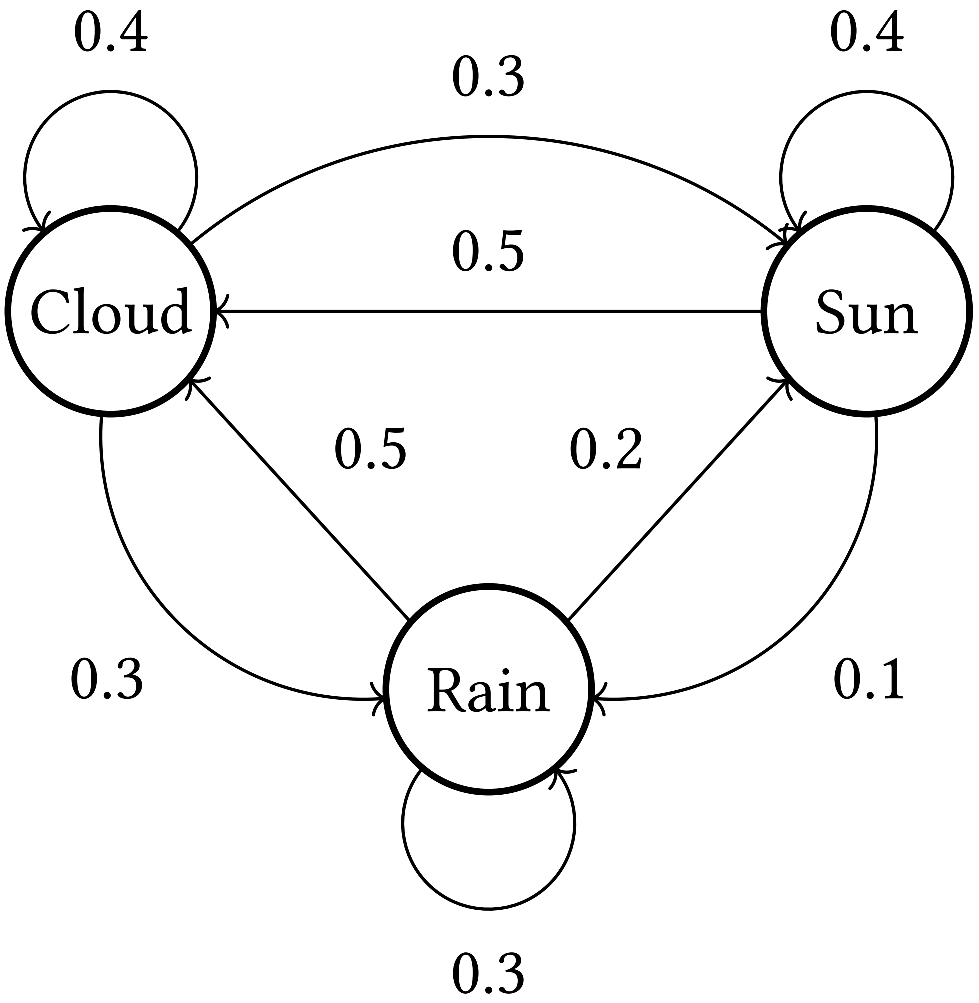
**Problem:** Predict the weather

$$S = \{\text{rain}, \text{cloud}, \text{sun}\} = \{R, C, S\}$$

$$\text{Tr}(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t)$$

$$= \begin{bmatrix} \Pr(C \mid C) & \Pr(R \mid C) & \Pr(S \mid C) \\ \Pr(C \mid R) & \Pr(R \mid R) & \Pr(S \mid R) \\ \Pr(C \mid S) & \Pr(R \mid S) & \Pr(S \mid S) \end{bmatrix}$$

$$= \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.5 & 0.3 & 0.2 \\ 0.5 & 0.1 & 0.4 \end{bmatrix}$$

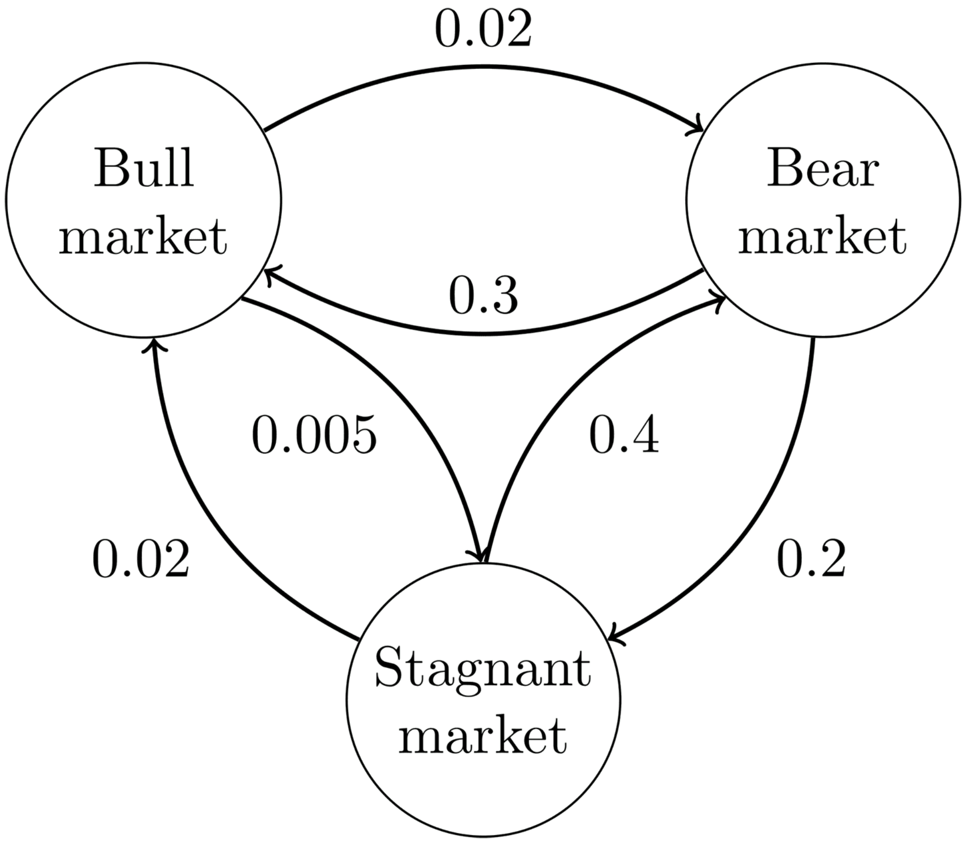


# Markov Processes

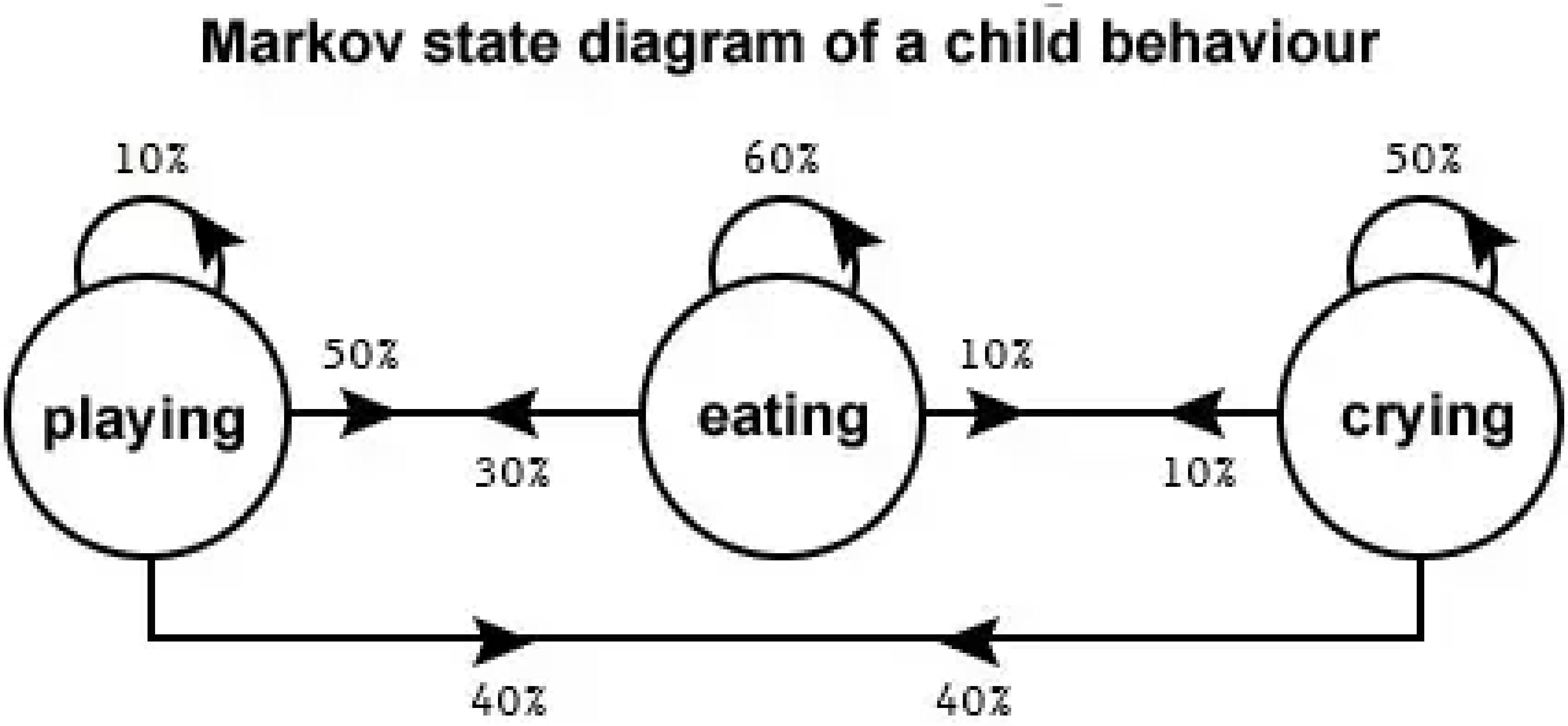
We can model many other systems as Markov processes

# Markov Processes

We can model many other systems as Markov processes



# Markov Processes



# Markov Processes

Why is it called a **Markov** process?

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

$$\Pr(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$$

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

$$\Pr(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$$

**This is a very important condition we must always satisfy**

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

$$\Pr(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$$

**This is a very important condition we must always satisfy**

If we cannot satisfy it, then the process is **not** Markov

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

$$\Pr(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$$

**This is a very important condition we must always satisfy**

If we cannot satisfy it, then the process is **not** Markov

$$\Pr(s_2 = \text{sun} \mid s_1 = \text{rain}, s_0 = \text{sun}) = 0.4$$

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

$$\Pr(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$$

**This is a very important condition we must always satisfy**

If we cannot satisfy it, then the process is **not** Markov

$$\begin{aligned} \Pr(s_2 = \text{sun} \mid s_1 = \text{rain}, s_0 = \text{sun}) &= 0.4 \\ \Pr(s_2 = \text{sun} \mid s_1 = \text{rain}) &= 0.3 \end{aligned}$$

# Markov Processes

Why is it called a **Markov** process?

Markov property: The next state only depends on the current state

$$\Pr(s_{t+1} \mid s_t) = \Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$$

**This is a very important condition we must always satisfy**

If we cannot satisfy it, then the process is **not** Markov

$$\begin{aligned}\Pr(s_2 = \text{sun} \mid s_1 = \text{rain}, s_0 = \text{sun}) &= 0.4 \\ \Pr(s_2 = \text{sun} \mid s_1 = \text{rain}) &= 0.3\end{aligned}$$

$0.3 \neq 0.4, \Pr(s_{t+1} \mid s_t) \neq \Pr, (s_{t+1} \mid s_t, s_{t-1}, ..., s_0)$, **not** Markov

# Markov Processes

We can visualize the Markov property too

# Markov Processes

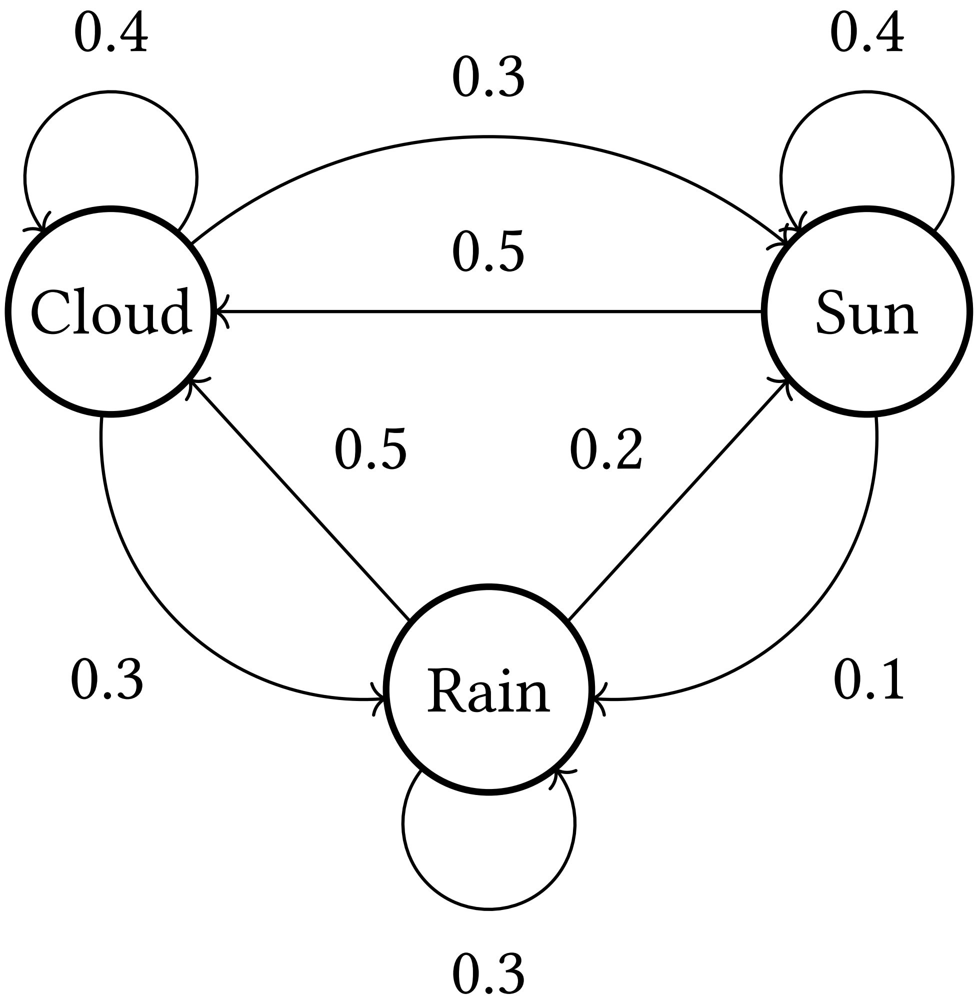
We can visualize the Markov property too

To compute the next node, we only look at the current node

# Markov Processes

We can visualize the Markov property too

To compute the next node, we only look at the current node



# Markov Processes

We can predict the future using Markov processes

# Markov Processes

We can predict the future using Markov processes

Chain transition probabilities together to estimate $s_t$

# Markov Processes

We can predict the future using Markov processes

Chain transition probabilities together to estimate $s_t$

$$\Pr(s_1 \mid s_0 = s)$$

# Markov Processes

We can predict the future using Markov processes

Chain transition probabilities together to estimate $s_t$

$$\Pr(s_1 \mid s_0 = s)$$

$$\Pr(s_2 \mid s_0 = s) = \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

# Markov Processes

We can predict the future using Markov processes

Chain transition probabilities together to estimate $s_t$

$$\Pr(s_1 \mid s_0 = s)$$

$$\Pr(s_2 \mid s_0 = s) = \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Paths from all possible $s_1$ to $s_2$

# Markov Processes

We can predict the future using Markov processes

Chain transition probabilities together to estimate $s_t$

$$\Pr(s_1 \mid s_0 = s)$$

$$\Pr(s_2 \mid s_0 = s) = \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Paths from all possible $s_1$ to $s_2$

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \Pr(s_3 \mid s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

# Markov Processes

We can predict the future using Markov processes

Chain transition probabilities together to estimate $s_t$

$$\Pr(s_1 \mid s_0 = s)$$

$$\Pr(s_2 \mid s_0 = s) = \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Paths from all possible $s_1$ to $s_2$

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \Pr(s_3 \mid s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Can we derive a general form for $P(s_n \mid s_0)$?

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \Pr(s_3 \mid s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \Pr(s_3 \mid s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Product of sum is the sum of products

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \Pr(s_3 \mid s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Product of sum is the sum of products, move the sum outside

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \sum_{s_1 \in S} \Pr(s_3 \mid s_2) \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \Pr(s_3 \mid s_2) \sum_{s_1 \in S} \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Product of sum is the sum of products, move the sum outside

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_2 \in S} \sum_{s_1 \in S} \Pr(s_3 \mid s_2) \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Combine the sums

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_1, s_2 \in S} \Pr(s_3 \mid s_2) \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_1, s_2 \in S} \Pr(s_3 \mid s_2) \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_1, s_2 \in S} \Pr(s_3 \mid s_2) \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Combine the products

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_1, s_2 \in S} \prod_{t=0}^{2} \Pr(s_{t+1} \mid s_t)$$

# Markov Processes

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_1, s_2 \in S} \Pr(s_3 \mid s_2) \Pr(s_2 \mid s_1) \Pr(s_1 \mid s_0 = s)$$

Combine the products

$$\Pr(s_3 \mid s_0 = s) = \sum_{s_1, s_2 \in S} \prod_{t=0}^{2} \Pr(s_{t+1} \mid s_t)$$

Generalize to any timestep $n$

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \ldots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

# Markov Processes

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \ldots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

# Markov Processes

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \ldots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

This expression tells us how the Markov process evolves over time

# Markov Processes

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \dots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

This expression tells us how the Markov process evolves over time

We can predict the future, $n$ timesteps from now

# Markov Processes

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \ldots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

This expression tells us how the Markov process evolves over time

We can predict the future, $n$ timesteps from now

If $s$ is the state of the world, you can predict the future of the world

# Markov Processes

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \ldots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

This expression tells us how the Markov process evolves over time

We can predict the future, $n$ timesteps from now

If $s$ is the state of the world, you can predict the future of the world

If $s$ represents someone's mind, you can predict their future thoughts

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

However, many processes we like to model eventually end

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

However, many processes we like to model eventually end

- Dying in a video game

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

However, many processes we like to model eventually end

- Dying in a video game
- Completing a task

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

However, many processes we like to model eventually end

- Dying in a video game
- Completing a task
- Running out of money

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

However, many processes we like to model eventually end

- Dying in a video game
- Completing a task
- Running out of money

**Question:** How can we model a Markov process that ends?

# Markov Processes

We can predict a state $s_n$, but a Markov process never ends

The future infinite, there is always a next state

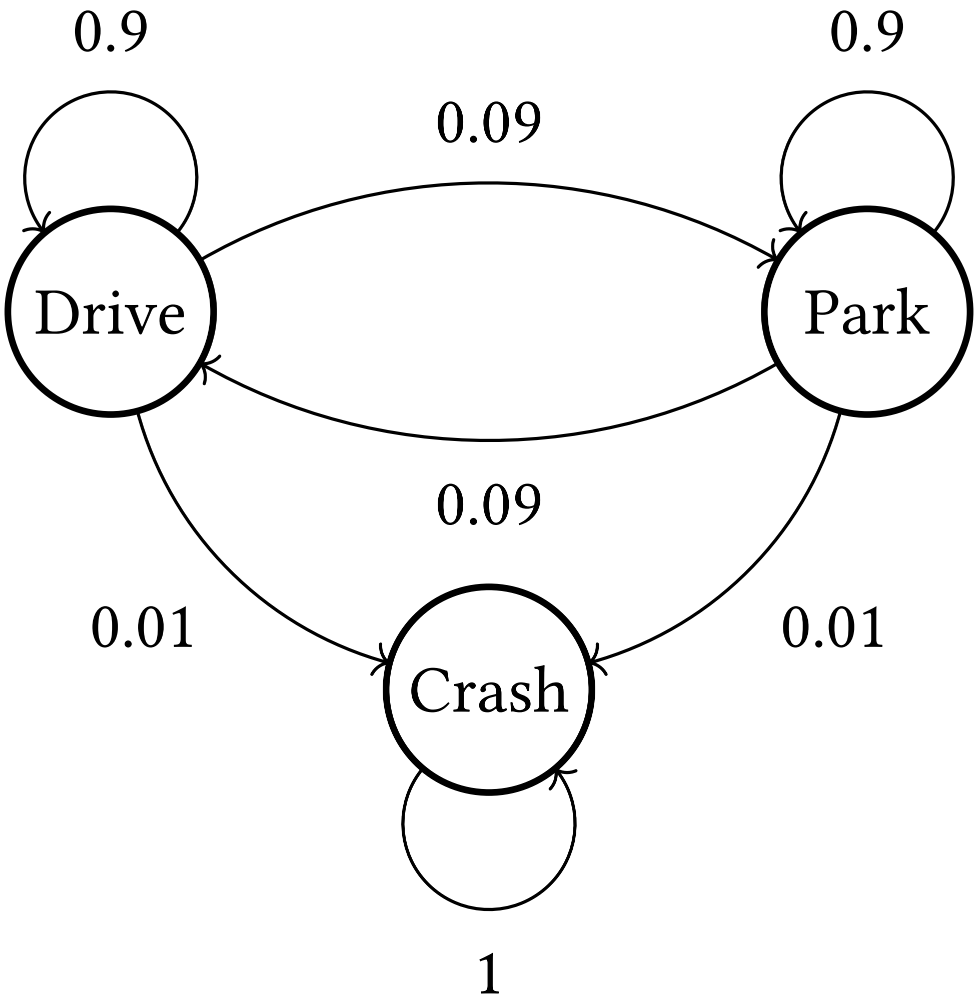
However, many processes we like to model eventually end

- Dying in a video game
- Completing a task
- Running out of money

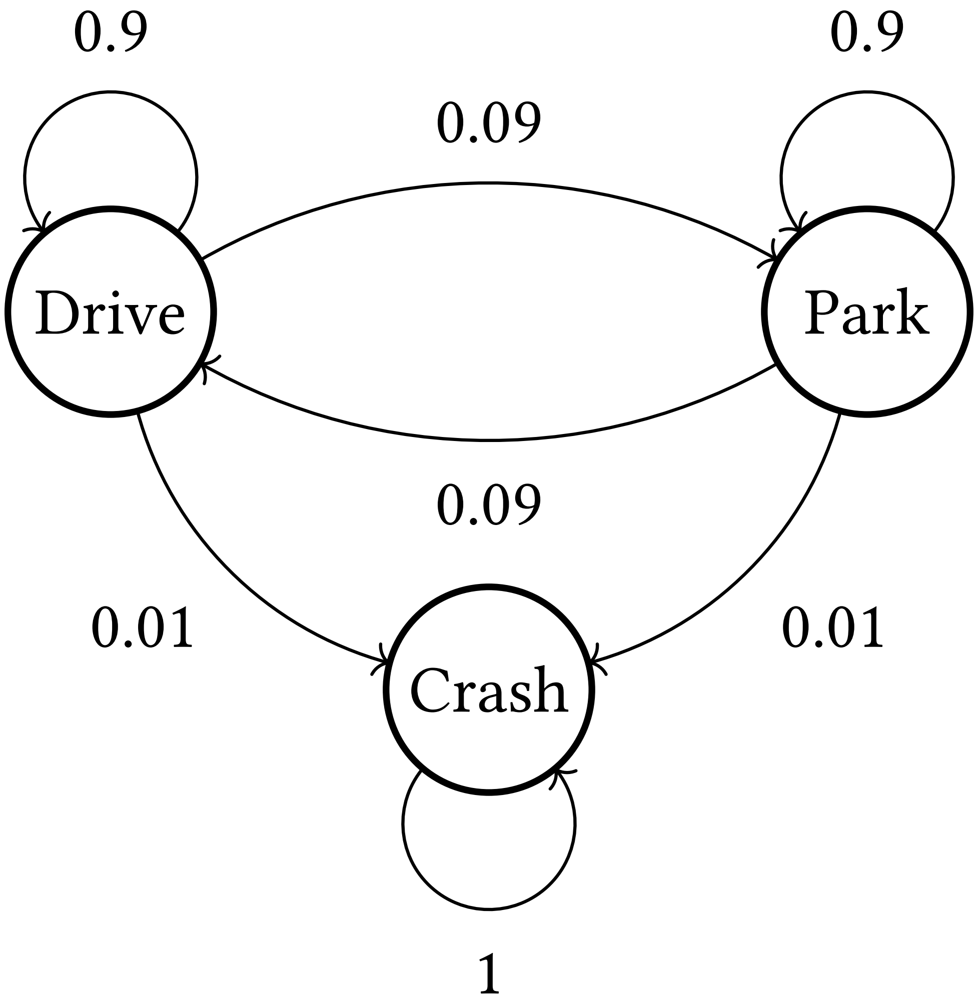
**Question:** How can we model a Markov process that ends?

**Answer:** We create a **terminal state** that we can enter but cannot leave

# Markov Processes

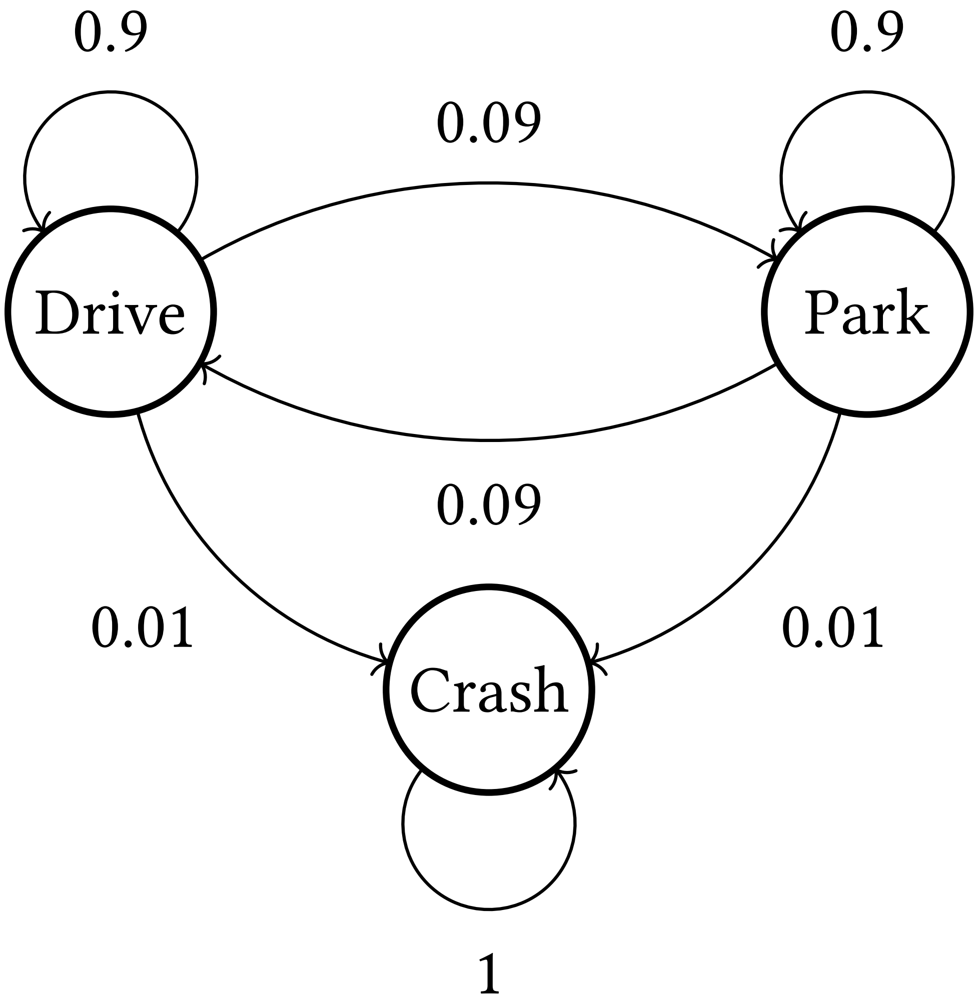


# Markov Processes




Upon reaching a terminal state, we get stuck

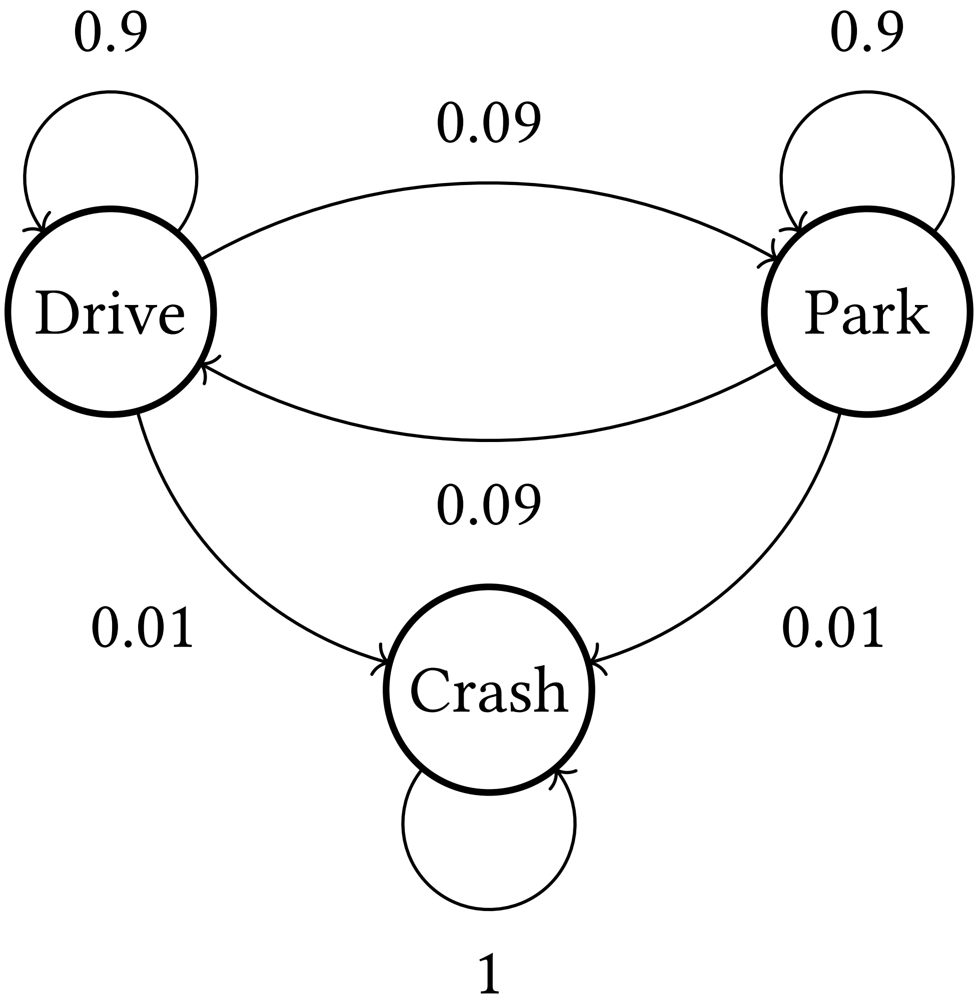
# Markov Processes




Upon reaching a terminal state, we get stuck

Once we crash our car, we cannot drive or park any more

# Markov Processes




Upon reaching a terminal state, we get stuck

Once we crash our car, we cannot drive or park any more

The only transition from a terminal state is back to itself

$$\Pr(s_{t+1} = \text{term} \mid s_t = \text{term}) = 1$$
$$\Pr(s_{t+1} = \text{not term} \mid s_t = \text{term}) = 0$$

# Exercise

# Exercise

Design an Markov process about a problem you care about

# Exercise

Design an Markov process about a problem you care about

- 4 states

# Exercise

Design an Markov process about a problem you care about

- 4 states
- State transition function $\mathrm{Tr} = \Pr(s_{t+1} \mid s_t)$ for all $s_t, s_{t+1} \in S$

# Exercise

Design an Markov process about a problem you care about

- 4 states
- State transition function $\mathrm{Tr} = \Pr(s_{t+1} \mid s_t)$ for all $s_t, s_{t+1} \in S$
- Create a terminal state

# Exercise

Design an Markov process about a problem you care about

- 4 states
- State transition function $\mathrm{Tr} = \Pr(s_{t+1} \mid s_t)$ for all $s_t, s_{t+1} \in S$
- Create a terminal state
- Given a starting state $s_0$, what will your state distribution be for $s_2$?

# Exercise

Design an Markov process about a problem you care about

- 4 states
- State transition function $\mathrm{Tr} = \Pr(s_{t+1} \mid s_t)$ for all $s_t, s_{t+1} \in S$
- Create a terminal state
- Given a starting state $s_0$, what will your state distribution be for $s_2$?

$$\Pr(s_n \mid s_0) = \sum_{s_1, s_2, \ldots s_{n-1} \in S} \prod_{t=0}^{n-1} \Pr(s_{t+1} \mid s_t)$$

# Markov Control Processes

# Markov Control Processes

Markov processes model complex evolving processes

# Markov Control Processes

Markov processes model complex evolving processes

But this course is on decision making, how can we model decision making in a Markov process?

We can't

# Markov Control Processes

Markov processes model complex evolving processes

But this course is on decision making, how can we model decision making in a Markov process?

We can't

Markov processes follow the state transition function $\mathrm{Tr}$

# Markov Control Processes

Markov processes model complex evolving processes

But this course is on decision making, how can we model decision making in a Markov process?

We can't

Markov processes follow the state transition function $\mathrm{Tr}$

The future of the system is already determined

# Markov Control Processes

Markov processes model complex evolving processes

But this course is on decision making, how can we model decision making in a Markov process?

We can't

Markov processes follow the state transition function $\mathrm{Tr}$

The future of the system is already determined

There is no room for decisions to change the fate of the system

# Markov Control Processes

Markov processes model complex evolving processes

But this course is on decision making, how can we model decision making in a Markov process?

We can't

Markov processes follow the state transition function $\mathrm{Tr}$

The future of the system is already determined

There is no room for decisions to change the fate of the system

We will modify the Markov process for decision making

# Markov Control Processes

Markov processes model complex evolving processes

But this course is on decision making, how can we model decision making in a Markov process?

We can't

Markov processes follow the state transition function $\mathrm{Tr}$

The future of the system is already determined

There is no room for decisions to change the fate of the system

We will modify the Markov process for decision making

The point of decision making is to choose our fate

# Markov Control Processes

A Markov process models the predetermined evolution of some system

# Markov Control Processes

A Markov process models the predetermined evolution of some system

We call this system the **environment**, because we cannot control it

# Markov Control Processes

A Markov process models the predetermined evolution of some system

We call this system the **environment**, because we cannot control it

The **agent** lives in the environment

# Markov Control Processes

A Markov process models the predetermined evolution of some system

We call this system the **environment**, because we cannot control it

The **agent** lives in the environment

The agent makes decisions

# Markov Control Processes

A Markov process models the predetermined evolution of some system

We call this system the **environment**, because we cannot control it

The **agent** lives in the environment

The agent makes decisions

The agent changes the environment with its decisions

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

The action space $A$ defines what our agent can do

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

The action space $A$ defines what our agent can do rrr

Markov process

$$(S, \mathrm{Tr})$$

$$\mathrm{Tr} : S \mapsto \Delta S$$

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

The action space $A$ defines what our agent can do rrr

| Markov process | Markov control process |
| --- | --- |
| $(S, \mathrm{Tr})$ | $(S, A, \mathrm{Tr})$ |
| $\mathrm{Tr} : S \mapsto \Delta S$ | $\mathrm{Tr} : S \times A \mapsto \Delta S$ |

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

The action space $A$ defines what our agent can do rrr

| Markov process | Markov control process |
|---|---|
| $(S, \mathrm{Tr})$ | $(S, A, \mathrm{Tr})$ |
| $\mathrm{Tr} : S \mapsto \Delta S$ | $\mathrm{Tr} : S \times A \mapsto \Delta S$ |

In a Markov process, the future follows a predefined evolution

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

The action space $A$ defines what our agent can do rrr

| Markov process | Markov control process |
|---|---|
| $(S, \mathrm{Tr})$ | $(S, A, \mathrm{Tr})$ |
| $\mathrm{Tr} : S \mapsto \Delta S$ | $\mathrm{Tr} : S \times A \mapsto \Delta S$ |

In a Markov process, the future follows a predefined evolution

In a Markov control process, we can control the evolution!

# Markov Control Processes

The agent takes **actions** $a \in A$ that change the environment

The action space $A$ defines what our agent can do rrr

| Markov process | Markov control process |
| --- | --- |
| $(S, \mathrm{Tr})$ | $(S, A, \mathrm{Tr})$ |
| $\mathrm{Tr} : S \mapsto \Delta S$ | $\mathrm{Tr} : S \times A \mapsto \Delta S$ |

In a Markov process, the future follows a predefined evolution

In a Markov control process, we can control the evolution!

Let us see an example

# Markov Control Processes

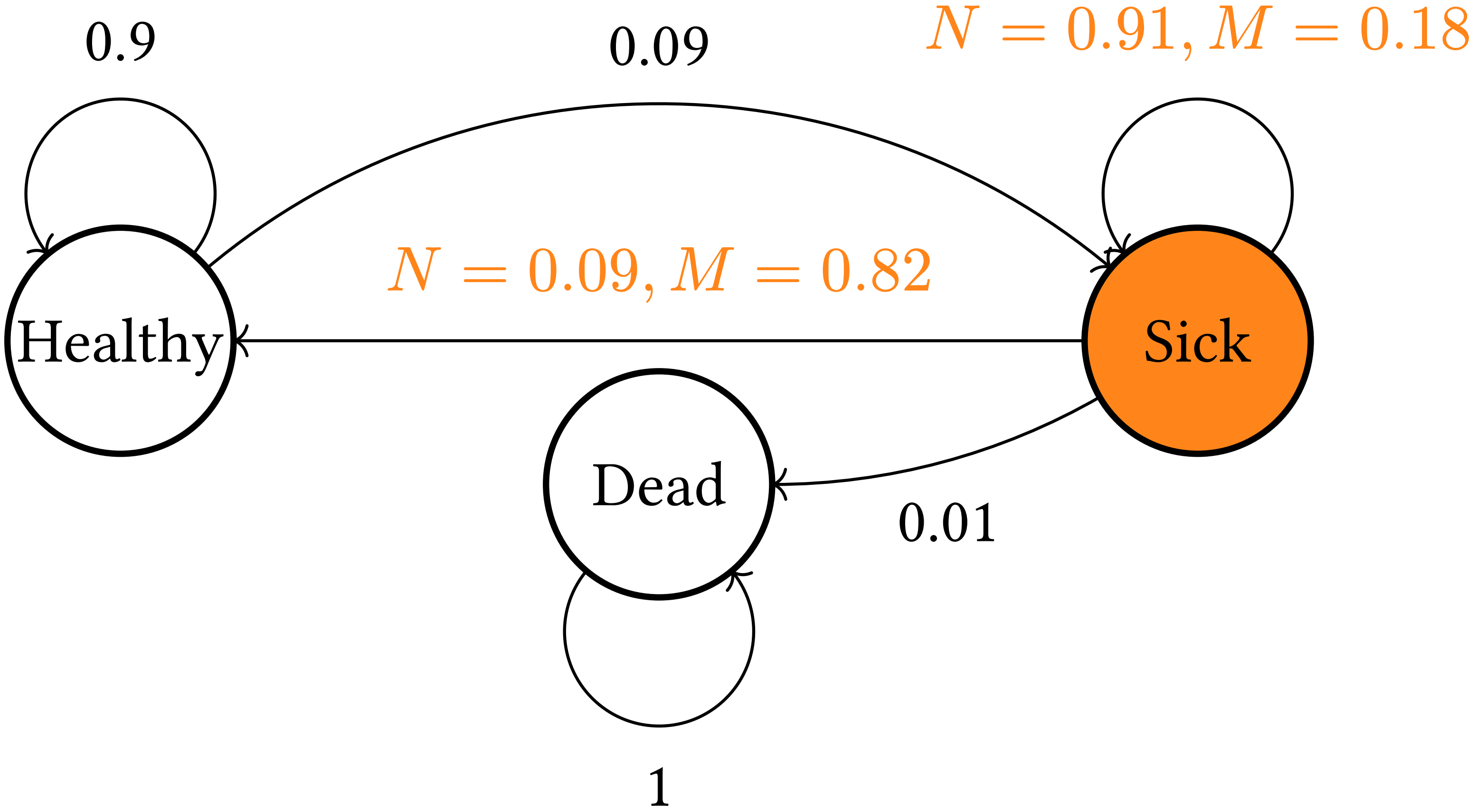
$$S = \{\text{Healthy}, \text{Sick}, \text{Dead}\}$$

# Markov Control Processes

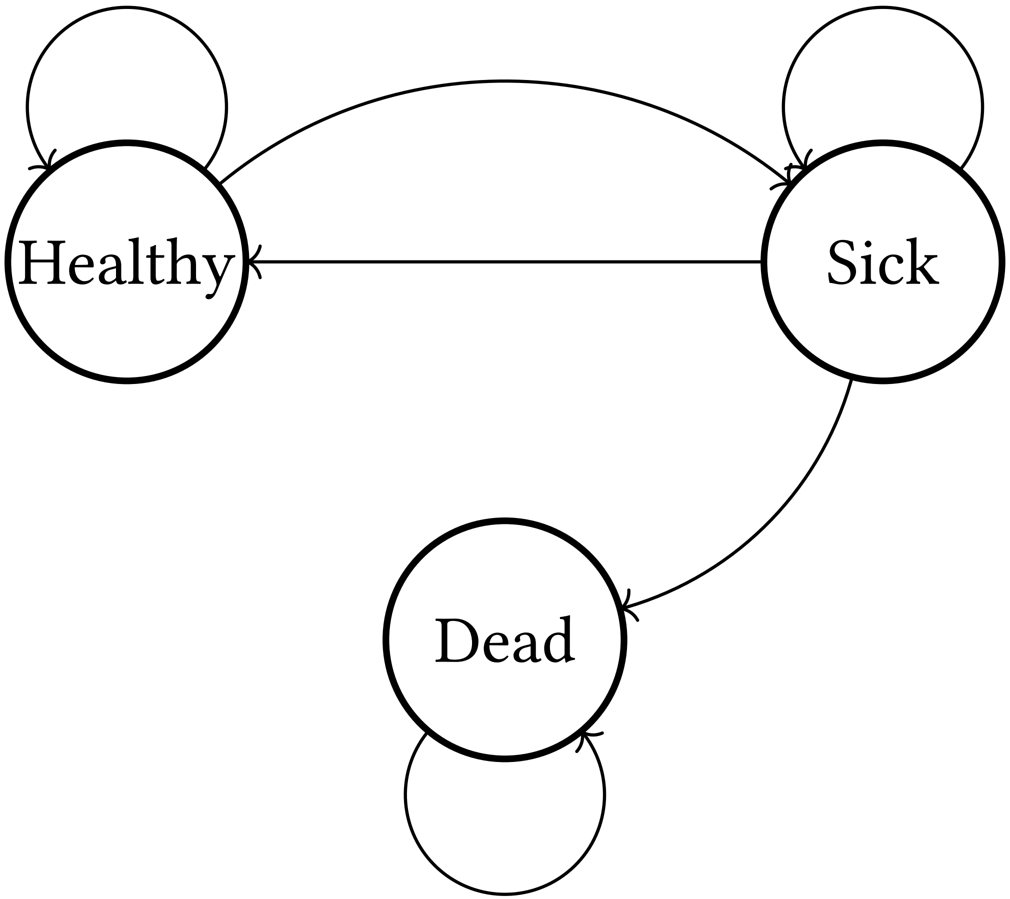
$$S = \{\text{Healthy}, \text{Sick}, \text{Dead}\} \qquad A = \{\text{Nothing}, \text{Medicine}\} = \{N, M\}$$

# Markov Control Processes

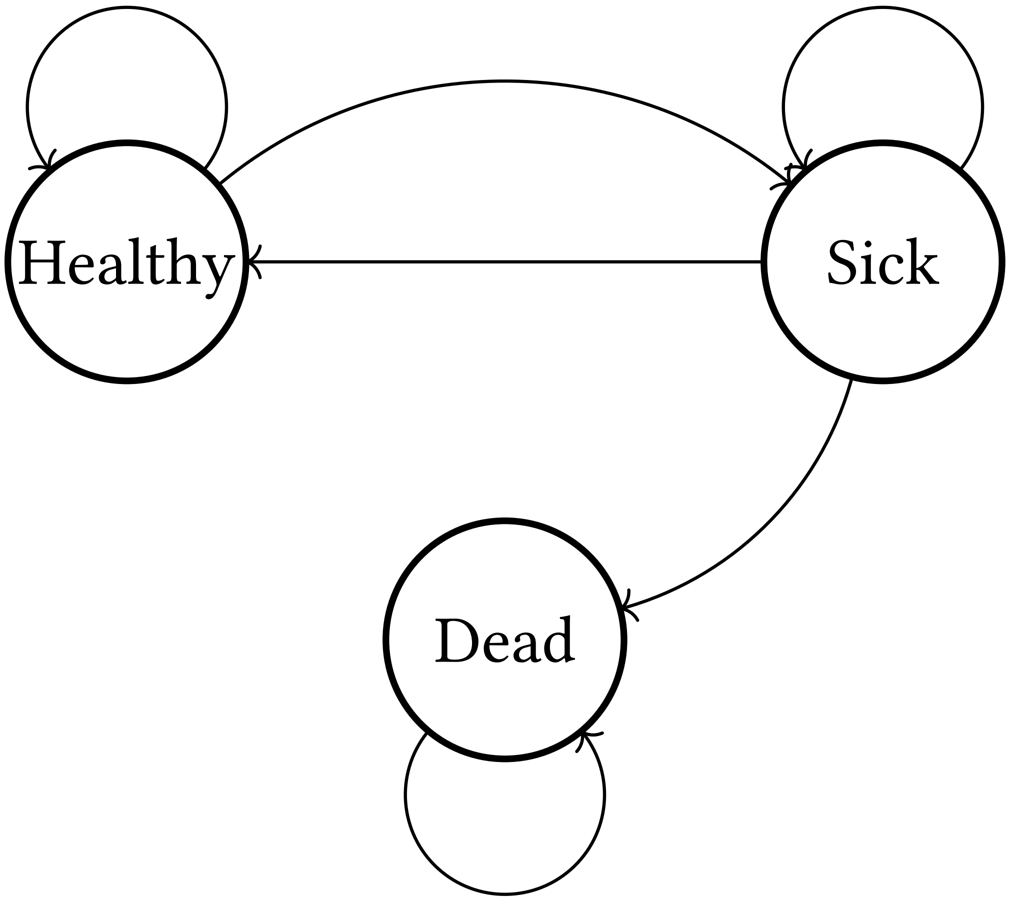
$$S = \{\text{Healthy}, \text{Sick}, \text{Dead}\} \qquad A = \{\text{Nothing}, \text{Medicine}\} = \{N, M\}$$



# Markov Control Processes

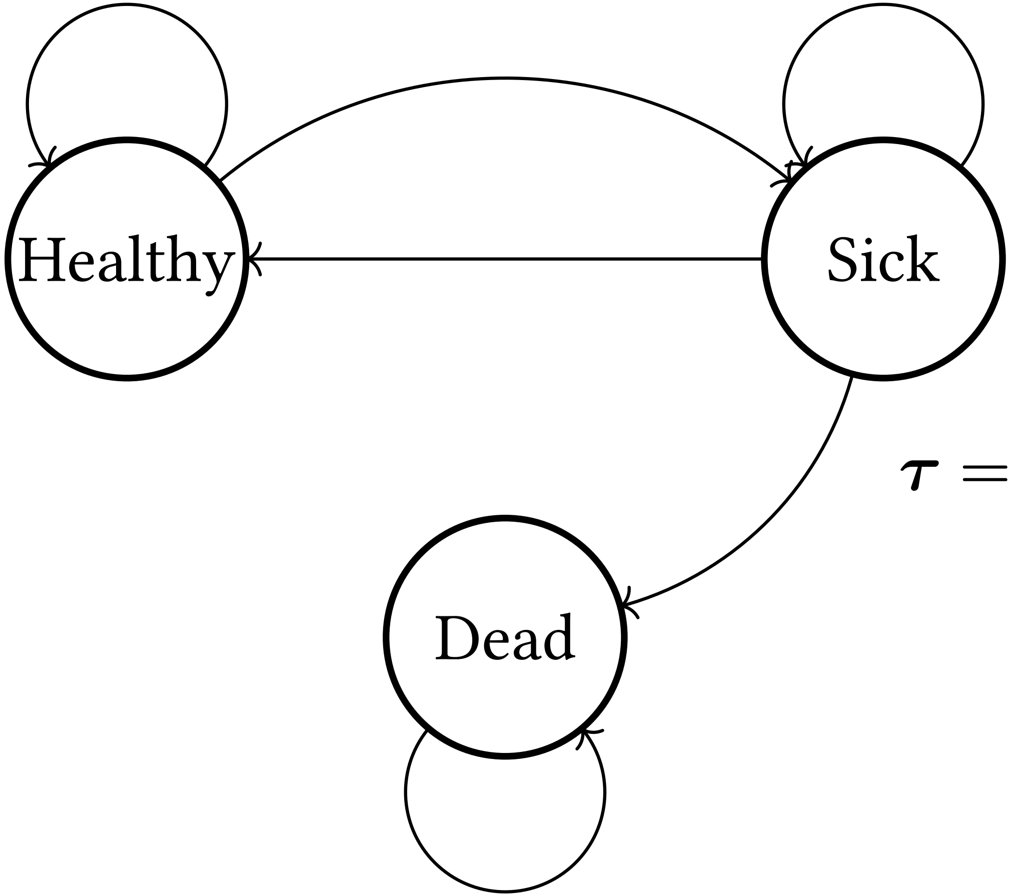


# Markov Control Processes




The **trajectory** contains the states and actions until a terminal state

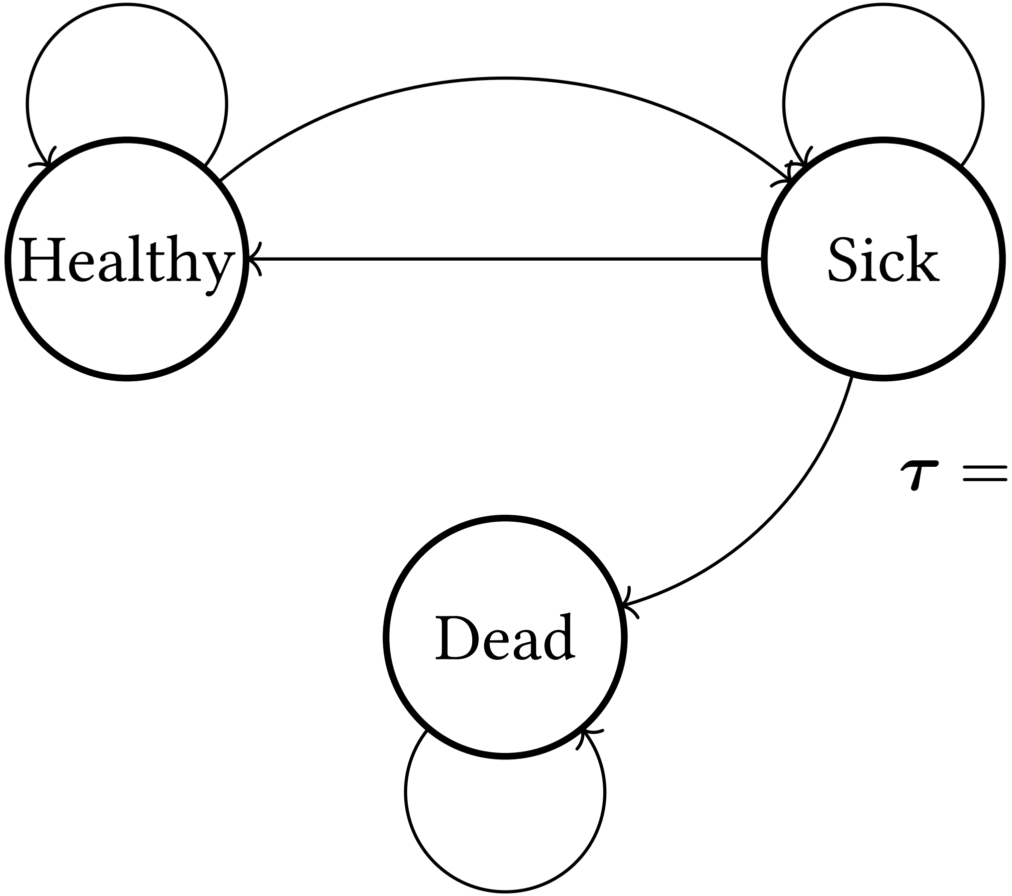
# Markov Control Processes




The **trajectory** contains the states and actions until a terminal state

$$\boldsymbol{\tau} = \begin{bmatrix} s_0 & a_0 \\ s_1 & a_1 \\ s_2 & a_2 \\ \vdots & \vdots \\ s_{n-1} & a_{n-1} \\ s_n & \emptyset \end{bmatrix}$$

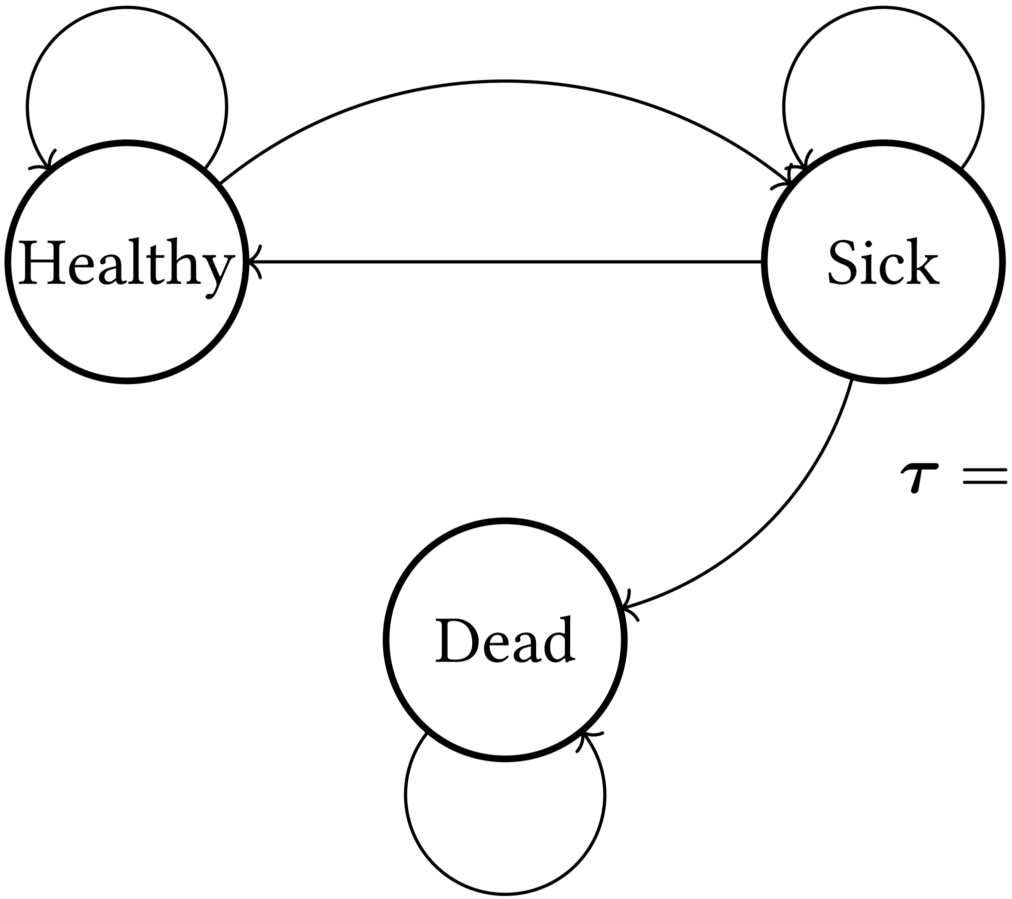
# Markov Control Processes




The **trajectory** contains the states and actions until a terminal state

$$\boldsymbol{\tau} = \begin{bmatrix} s_0 & a_0 \\ s_1 & a_1 \\ s_2 & a_2 \\ \vdots & \vdots \\ s_{n-1} & a_{n-1} \\ s_n & \emptyset \end{bmatrix} = \begin{bmatrix} \text{Healthy} & \text{Nothing} \\ \text{Sick} & \text{Nothing} \\ \text{Sick} & \text{Medicine} \\ \vdots & \vdots \\ \text{Sick} & \text{Nothing} \\ \text{Dead} & \emptyset \end{bmatrix}$$

# Markov Control Processes




The **trajectory** contains the states and actions until a terminal state

$$\boldsymbol{\tau} = \begin{bmatrix} s_0 & a_0 \\ s_1 & a_1 \\ s_2 & a_2 \\ \vdots & \vdots \\ s_{n-1} & a_{n-1} \\ s_n & \emptyset \end{bmatrix} = \begin{bmatrix} \text{Healthy} & \text{Nothing} \\ \text{Sick} & \text{Nothing} \\ \text{Sick} & \text{Medicine} \\ \vdots & \vdots \\ \text{Sick} & \text{Nothing} \\ \text{Dead} & \emptyset \end{bmatrix}$$

If there is no terminal state, the trajectory can be infinitely long!

# Markov Control Processes

Markov control processes let us control which states we visit

# Markov Control Processes

Markov control processes let us control which states we visit

They do not tell us which states are good to visit, or provide a goal

# Markov Control Processes

Markov control processes let us control which states we visit

They do not tell us which states are good to visit, or provide a goal

How can we make optimal decisions if we do not have a goal or objective?

# Markov Control Processes

Markov control processes let us control which states we visit

They do not tell us which states are good to visit, or provide a goal

How can we make optimal decisions if we do not have a goal or objective?

We need a way to determine “good” and “bad” decisions

# Markov Decision Processes

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of “goodness” to Markov control processes

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of "goodness" to Markov control processes

We use a **reward function** $R$ to measure the goodness of a specific state

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of "goodness" to Markov control processes

We use a **reward function** $R$ to measure the goodness of a specific state

Sutton and Barto:

$$R : S \times A \mapsto \mathbb{R}$$

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of “goodness” to Markov control processes

We use a **reward function** $R$ to measure the goodness of a specific state

Sutton and Barto:

$$R : S \times A \mapsto \mathbb{R}$$

Other books:

$$R : S \times A \times S \mapsto \mathbb{R}$$

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of "goodness" to Markov control processes

We use a **reward function** $R$ to measure the goodness of a specific state

| Sutton and Barto: | Other books: | This course: |
| --- | --- | --- |
| $R : S \times A \mapsto \mathbb{R}$ | $R : S \times A \times S \mapsto \mathbb{R}$ | $R : S \mapsto \mathbb{R}$ |

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of "goodness" to Markov control processes

We use a **reward function** $R$ to measure the goodness of a specific state

| Sutton and Barto: | Other books: | This course: |
|---|---|---|
| $R : S \times A \mapsto \mathbb{R}$ | $R : S \times A \times S \mapsto \mathbb{R}$ | $R : S \mapsto \mathbb{R}$ |

For now, I will use the simplest one

# Markov Decision Processes

Markov decision processes (MDPs) add a measure of "goodness" to Markov control processes

We use a **reward function** $R$ to measure the goodness of a specific state

| Sutton and Barto: | Other books: | This course: |
| --- | --- | --- |
| $R : S \times A \mapsto \mathbb{R}$ | $R : S \times A \times S \mapsto \mathbb{R}$ | $R : S \mapsto \mathbb{R}$ |

For now, I will use the simplest one

You can always make these equivalent by modifying the MDP

# Markov Decision Processes

Markov process

$$(S, \mathrm{Tr})$$

$$\mathrm{Tr} : S \mapsto \Delta S$$

# Markov Decision Processes

Markov process

$$(S, \mathrm{Tr})$$

$$\mathrm{Tr} : S \mapsto \Delta S$$

Markov control process

$$(S, A, \mathrm{Tr})$$

$$\mathrm{Tr} : S \times A \mapsto \Delta S$$

# Markov Decision Processes

Markov process

$$(S, \mathrm{Tr})$$

$$\mathrm{Tr} : S \mapsto \Delta S$$

Markov control process

$$(S, A, \mathrm{Tr})$$

$$\mathrm{Tr} : S \times A \mapsto \Delta S$$

Markov decision process

$$(S, A, \mathrm{Tr}, R, \gamma)$$

$$\mathrm{Tr} : S \times A \mapsto \Delta S$$

$$R : S \mapsto \mathbb{R}$$

In an MDP, an **episode** contains the trajectory and also the rewards

# Markov Decision Processes

Markov process

$$(S, \mathrm{Tr})$$

$$\mathrm{Tr} : S \mapsto \Delta S$$

Markov control process

$$(S, A, \mathrm{Tr})$$

$$\mathrm{Tr} : S \times A \mapsto \Delta S$$

Markov decision process

$$(S, A, \mathrm{Tr}, R, \gamma)$$

$$\mathrm{Tr} : S \times A \mapsto \Delta S$$

$$R : S \mapsto \mathbb{R}$$

In an MDP, an **episode** contains the trajectory and also the rewards

$$\boldsymbol{E} = \begin{bmatrix} s_0 & a_0 & r_0 \\ s_1 & a_1 & r_1 \\ \vdots & \vdots & \vdots \\ s_{n-1} & a_{n-1} & r_{n-1} \\ s_n & \emptyset & \emptyset \end{bmatrix} = [\boldsymbol{\tau} \;\; \boldsymbol{r}]$$

# Markov Decision Processes

Reward is good, we want to maximize the reward

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

$s_d = \text{Dumpling}$ $\qquad$ $s_n = \text{Noodle}$

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

$$s_d = \text{Dumpling} \qquad s_n = \text{Noodle}$$

$$R(s_d) = 10 \qquad R(s_n) = 15$$

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

$$s_d = \text{Dumpling} \qquad s_n = \text{Noodle}$$

$$R(s_d) = 10 \qquad R(s_n) = 15$$

**Result:** Eat noodle

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

$$s_d = \text{Dumpling} \qquad s_n = \text{Noodle}$$

$$R(s_d) = 10 \qquad R(s_n) = 15$$

**Result:** Eat noodle

$$R(s_d) = 5 \qquad R(s_n) = -3$$

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

| | | |
|---|---|---|
| $s_d = \text{Dumpling}$ | | $s_n = \text{Noodle}$ |
| $R(s_d) = 10$ | $R(s_n) = 15$ | **Result:** Eat noodle |
| $R(s_d) = 5$ | $R(s_n) = -3$ | **Result:** Eat dumpling |

# Markov Decision Processes

Reward is good, we want to maximize the reward

The reward function determines the agent behavior

$$s_d = \text{Dumpling} \qquad s_n = \text{Noodle}$$

$$R(s_d) = 10 \qquad R(s_n) = 15$$ **Result:** Eat noodle

$$R(s_d) = 5 \qquad R(s_n) = -3$$ **Result:** Eat dumpling

We can write this mathematically as

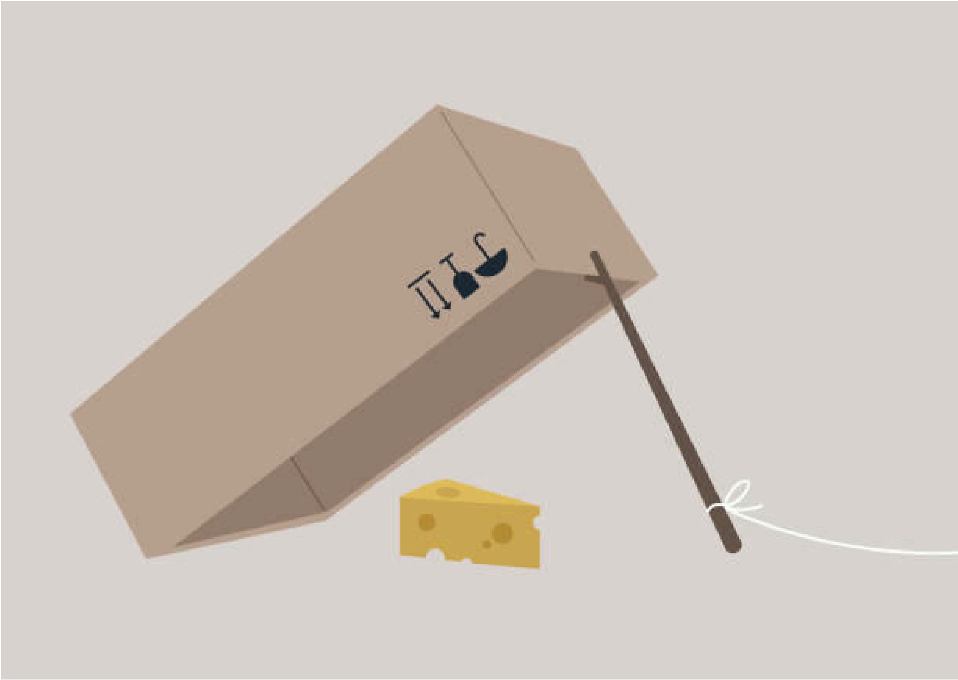
$$\arg\max_{s \in S} R(s)$$

# Markov Decision Processes

However, maximizing the reward is not always ideal

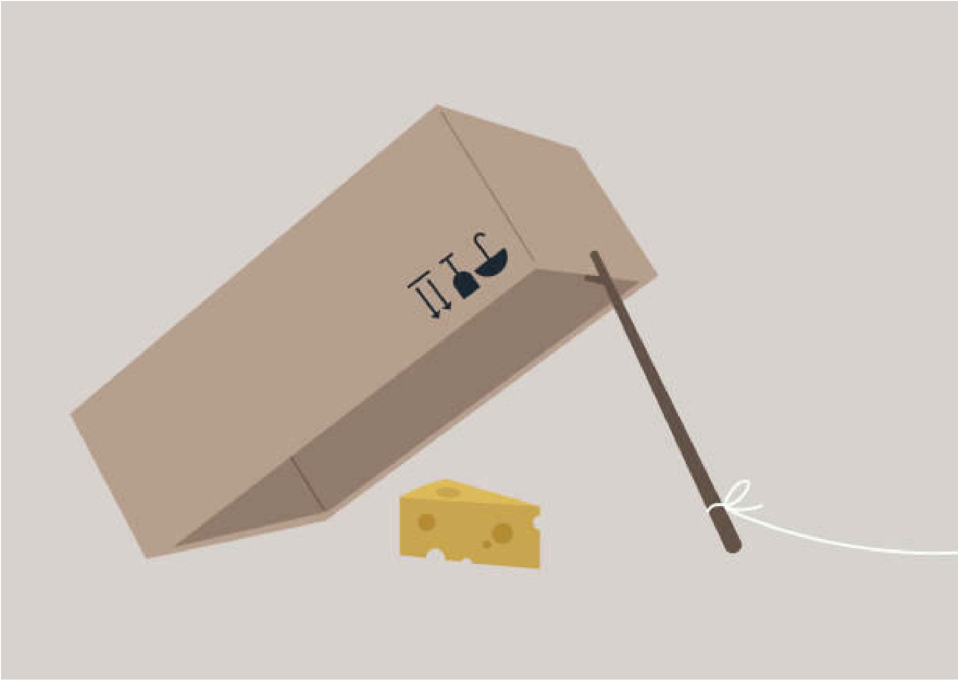
# Markov Decision Processes

However, maximizing the reward is not always ideal

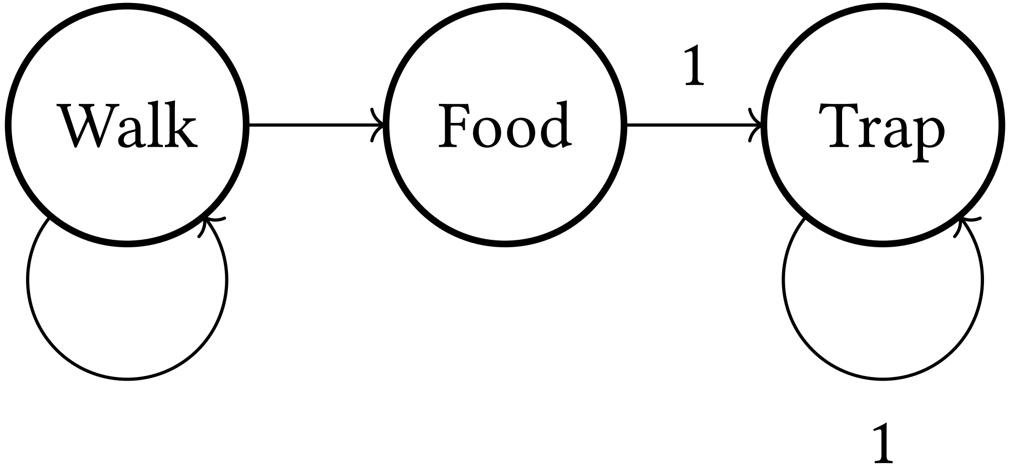


# Markov Decision Processes

However, maximizing the reward is not always ideal

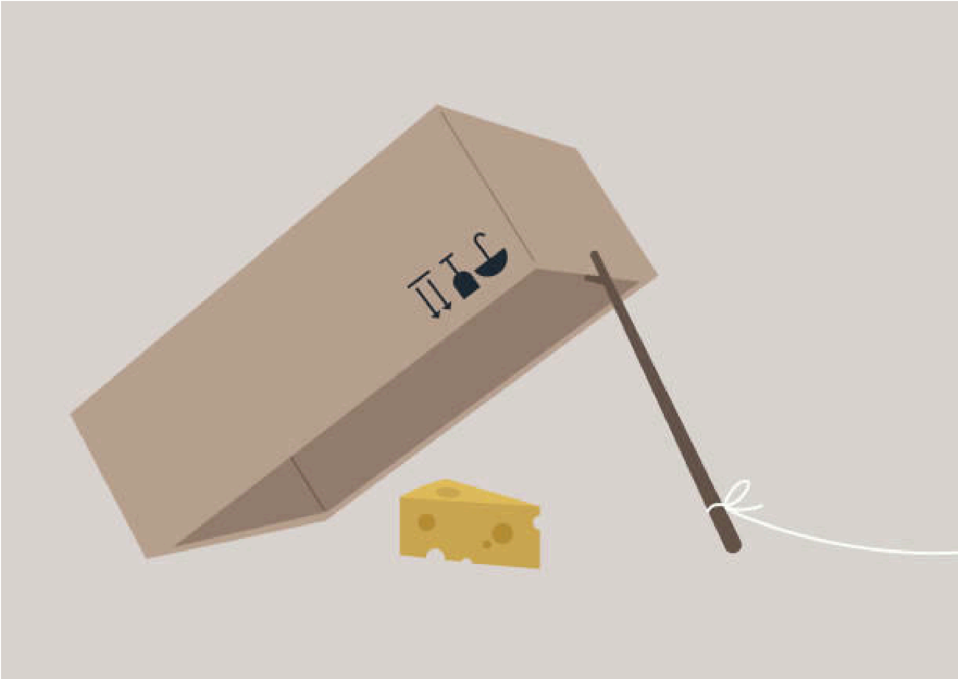


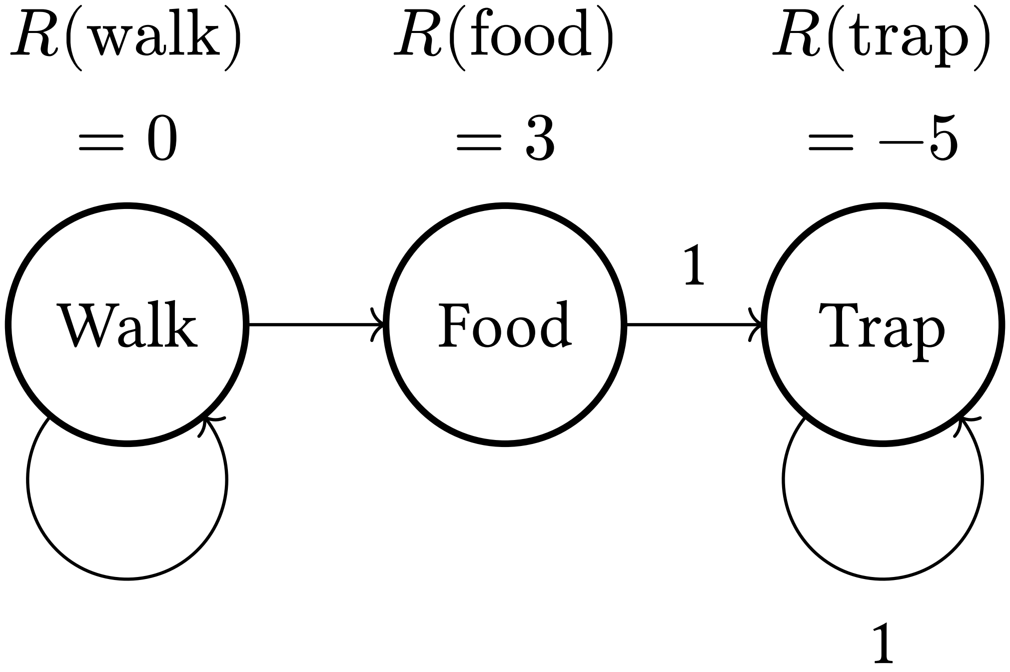
$R(\text{walk}) = 0$ $\quad R(\text{food}) = 3$ $\quad R(\text{trap}) = -5$



# Markov Decision Processes

However, maximizing the reward is not always ideal

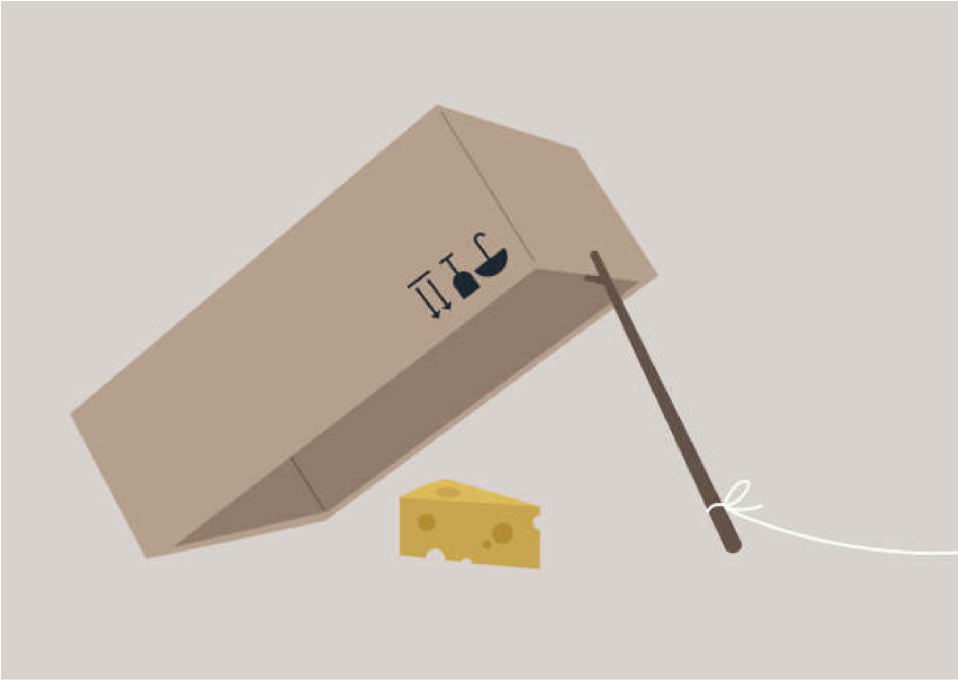


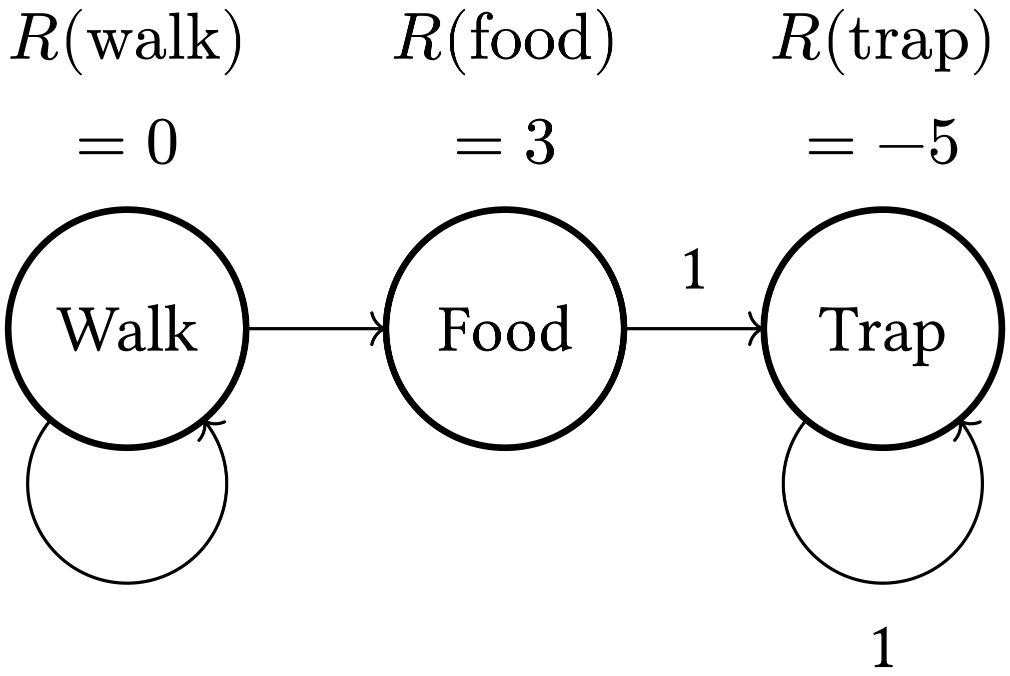



$$\arg\max_{a \in A} R(s) = \text{take the food}$$

# Markov Decision Processes

However, maximizing the reward is not always ideal






$$\arg\max_{a \in A} R(s) = \text{take the food}$$

If we maximize the reward, we are **too greedy**

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

We think about how our actions now will impact reward the future

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

We think about how our actions now will impact reward the future

We call the cumulative sum of rewards, the **return**

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

We think about how our actions now will impact reward the future

We call the cumulative sum of rewards, the **return**

$$G : \mathbb{R}^n \mapsto \mathbb{R}$$

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

We think about how our actions now will impact reward the future

We call the cumulative sum of rewards, the **return**

$$G : \mathbb{R}^n \mapsto \mathbb{R} \qquad\qquad G : S^n \times A^{n-1} \mapsto \mathbb{R}$$

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

We think about how our actions now will impact reward the future

We call the cumulative sum of rewards, the **return**

$$G : \mathbb{R}^n \mapsto \mathbb{R}$$

$$G : S^n \times A^{n-1} \mapsto \mathbb{R}$$

$$G(r_0, r_1, ...) = \sum_{t=0}^{\infty} r_t$$

# Markov Decision Processes

Maximizing the immediate reward can result in bad agents

Instead, we maximize the **cumulative sum** of rewards

We think about how our actions now will impact reward the future

We call the cumulative sum of rewards, the **return**

$$G : \mathbb{R}^n \mapsto \mathbb{R}$$

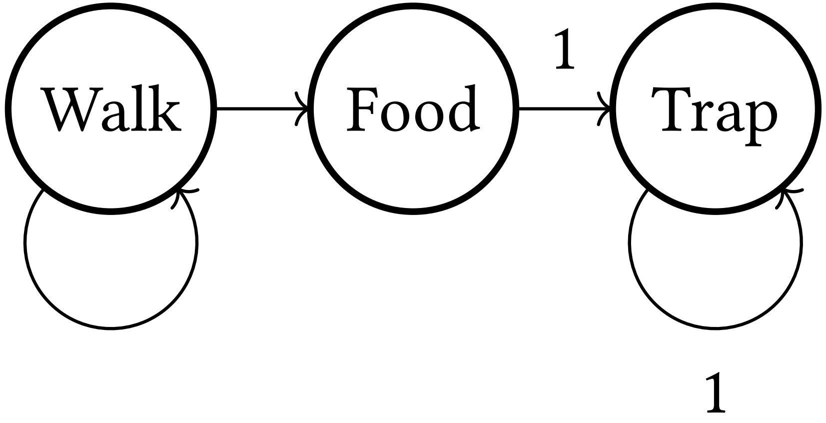
$$G(r_0, r_1, ...) = \sum_{t=0}^{\infty} r_t$$

$$G : S^n \times A^{n-1} \mapsto \mathbb{R}$$

$$\begin{aligned} G(\boldsymbol{\tau}) &= G(s_0, a_0, s_1, a_1, ...) \\ &= \sum_{t=0}^{\infty} R(s_{t+1}) \end{aligned}$$

# Markov Decision Processes

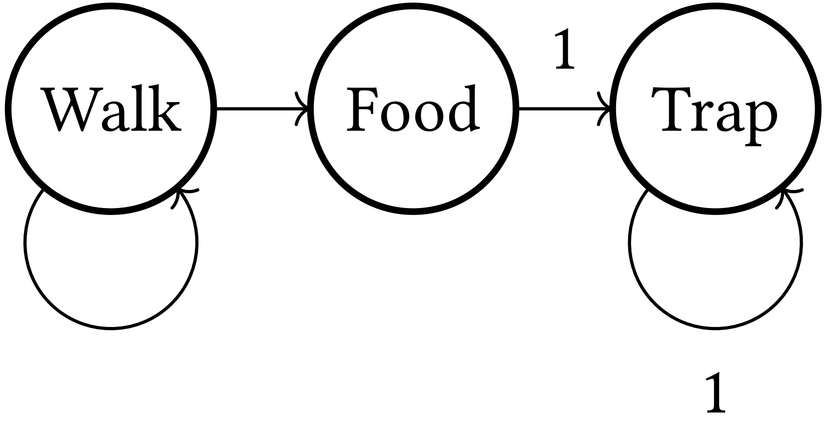
$R(\text{walk}) = 0$ $\quad R(\text{food}) = 3$ $\quad R(\text{trap}) = -5$




$$G(\boldsymbol{\tau}_{\text{greedy}}) = R(\text{food}) + R(\text{trap}) + R(\text{trap}) + \ldots = 3 - 5 - 5 - \ldots = -\infty$$

# Markov Decision Processes

$R(\text{walk}) = 0$ $\quad$ $R(\text{food}) = 3$ $\quad$ $R(\text{trap}) = -5$

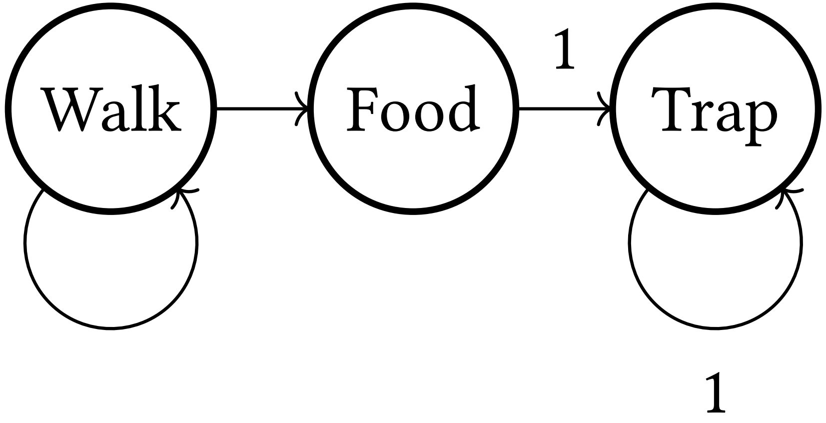



$$G(\boldsymbol{\tau}_{\text{greedy}}) = R(\text{food}) + R(\text{trap}) + R(\text{trap}) + \ldots = 3 - 5 - 5 - \ldots = -\infty$$

$$G(\boldsymbol{\tau}_{\text{smart}}) = R(\text{walk}) + R(\text{walk}) + R(\text{walk}) + \ldots = 0 + 0 + \ldots = 0$$

# Markov Decision Processes

$R(\text{walk}) = 0 \quad R(\text{food}) = 3 \quad R(\text{trap}) = -5$




$$G(\boldsymbol{\tau}_{\text{greedy}}) = R(\text{food}) + R(\text{trap}) + R(\text{trap}) + \ldots = 3 - 5 - 5 - \ldots = -\infty$$

$$G(\boldsymbol{\tau}_{\text{smart}}) = R(\text{walk}) + R(\text{walk}) + R(\text{walk}) + \ldots = 0 + 0 + \ldots \qquad = 0$$

By considering the future rewards, we can make optimal decisions

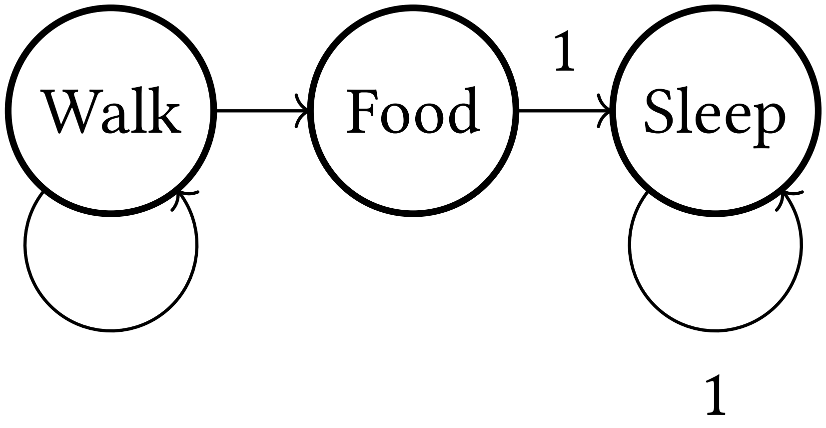
# Markov Decision Processes

Consider one more example

# Markov Decision Processes

Consider one more example

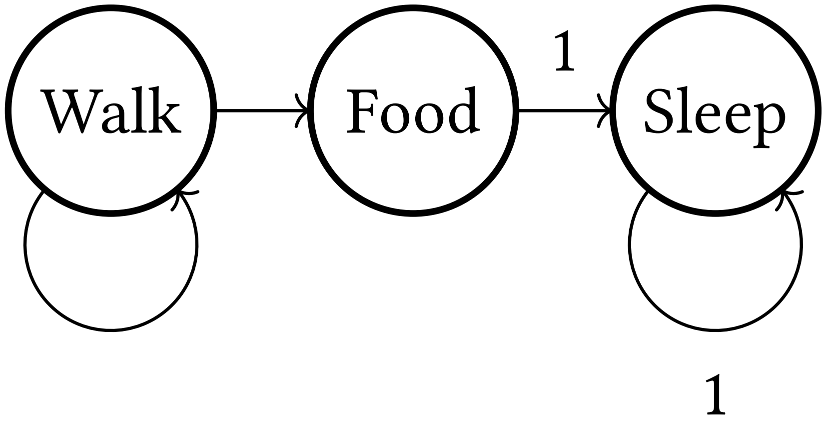
$R(\text{walk}) = 0 \quad R(\text{food}) = 3 \quad R(\text{sleep}) = 0$



# Markov Decision Processes

Consider one more example

$R(\text{walk}) = 0$ $\quad R(\text{food}) = 3$ $\quad R(\text{sleep}) = 0$



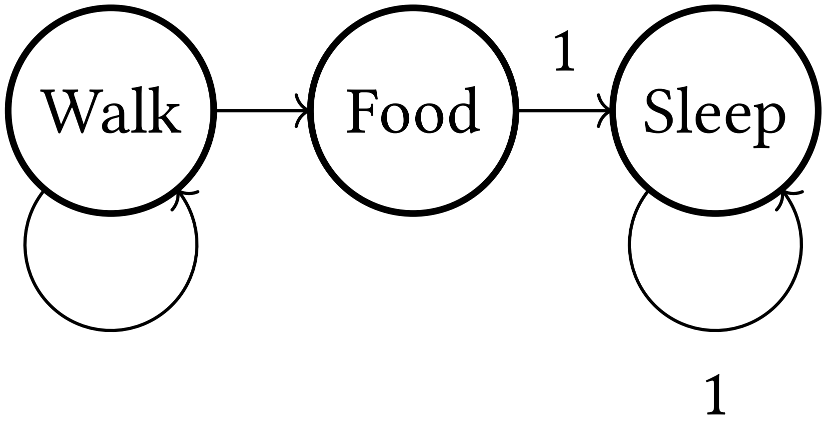

**Question:** What is the optimal sequence of states?

# Markov Decision Processes

Consider one more example

$R(\text{walk})$ $R(\text{food})$ $R(\text{sleep})$

$= 0$ $= 3$ $= 0$



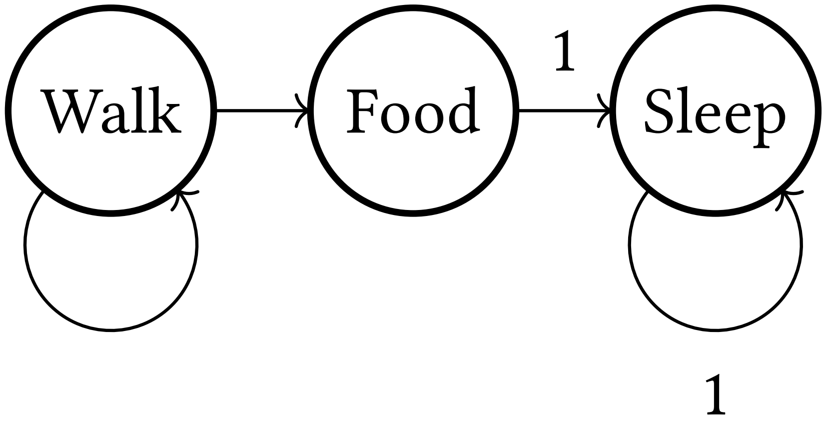

**Question:** What is the optimal sequence of states?

$$\text{Walk} + \text{Food} + \text{Sleep} + \ldots \qquad = 0 + 3 + 0 + \ldots \qquad = 3$$

# Markov Decision Processes

Consider one more example

$R(\text{walk}) = 0 \quad R(\text{food}) = 3 \quad R(\text{sleep}) = 0$




**Question:** What is the optimal sequence of states?

$$\text{Walk} + \text{Food} + \text{Sleep} + \ldots = 0 + 3 + 0 + \ldots = 3$$

$$\text{Walk} + \text{Walk} + \ldots + \text{Food} + \text{Sleep} + \ldots = 0 + 0 + \ldots + 3 + 0 + \ldots = 3$$

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

**Question:** Is this how humans make decisions?

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

**Question:** Is this how humans make decisions?

**Answer:** No, humans are a little bit greedy

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

**Question:** Is this how humans make decisions?

**Answer:** No, humans are a little bit greedy

**Experiment:** Place a cookie in front of a child. If they do not eat the cookie for 5 minutes, they get two cookies

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

**Question:** Is this how humans make decisions?

**Answer:** No, humans are a little bit greedy

**Experiment:** Place a cookie in front of a child. If they do not eat the cookie for 5 minutes, they get two cookies

**Question:** What happens?

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

**Question:** Is this how humans make decisions?

**Answer:** No, humans are a little bit greedy

**Experiment:** Place a cookie in front of a child. If they do not eat the cookie for 5 minutes, they get two cookies

**Question:** What happens?

**Answer:** Child eats the cookie immediately

# Markov Decision Processes

The return is a sum

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

We can eat food now, or in 1000 years, the return is the same

**Question:** Is this how humans make decisions?

**Answer:** No, humans are a little bit greedy

**Experiment:** Place a cookie in front of a child. If they do not eat the cookie for 5 minutes, they get two cookies

**Question:** What happens?

**Answer:** Child eats the cookie immediately

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

**Question:** How can we modify the return to prefer rewards sooner?

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

**Question:** How can we modify the return to prefer rewards sooner?

What if we make future rewards less important?

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

**Question:** How can we modify the return to prefer rewards sooner?

What if we make future rewards less important?

$$R(s_{t+1}) = \{1 \mid s_{t+1} \in S\}$$

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

**Question:** How can we modify the return to prefer rewards sooner?

What if we make future rewards less important?

$$R(s_{t+1}) = \{1 \mid s_{t+1} \in S\} \qquad G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} 1 = 1 + 1 + \dots$$

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

**Question:** How can we modify the return to prefer rewards sooner?

What if we make future rewards less important?

$$R(s_{t+1}) = \{1 \mid s_{t+1} \in S\}$$

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} 1 = 1 + 1 + \dots$$

$$G(\boldsymbol{\tau}) = \quad ? \quad = 1 + 0.9 + 0.8 + \dots$$

# Markov Decision Processes

Timing matters, humans prefer reward sooner rather than later

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} R(s_{t+1})$$

**Question:** How can we modify the return to prefer rewards sooner?

What if we make future rewards less important?

$$R(s_{t+1}) = \{1 \mid s_{t+1} \in S\}$$

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} 1 = 1 + 1 + \dots$$

$$G(\boldsymbol{\tau}) = \quad ? \quad = 1 + 0.9 + 0.8 + \dots$$

**Question:** How?

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

With $\gamma = 0$

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

With $\gamma = 0$

$$G(\boldsymbol{\tau}) = 1 + 1 + 1 + \dots$$

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

With $\gamma = 0$

$$G(\boldsymbol{\tau}) = 1 + 1 + 1 + ...$$

With $\gamma = 0.9$

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

With $\gamma = 0$

$$G(\boldsymbol{\tau}) = 1 + 1 + 1 + \dots$$

With $\gamma = 0.9$

$$G(\boldsymbol{\tau}) = (0.9^0 \cdot 1) + (0.9^1 \cdot 1) + (0.9^2 \cdot 1) +$$

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

With $\gamma = 0$

$$G(\boldsymbol{\tau}) = 1 + 1 + 1 + \dots$$

With $\gamma = 0.9$

$$G(\boldsymbol{\tau}) = (0.9^0 \cdot 1) + (0.9^1 \cdot 1) + (0.9^2 \cdot 1) +$$

$$G(\boldsymbol{\tau}) = 1 + 0.9 + 0.81 + \dots$$

We call this the **discounted return**

# Markov Decision Processes

We can introduce a **discount** term $\gamma \in [0, 1]$ to the return

$$G(\boldsymbol{\tau}) = \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

With $\gamma = 0$

$$G(\boldsymbol{\tau}) = 1 + 1 + 1 + ...$$

With $\gamma = 0.9$

$$G(\boldsymbol{\tau}) = (0.9^0 \cdot 1) + (0.9^1 \cdot 1) + (0.9^2 \cdot 1) +$$

$$G(\boldsymbol{\tau}) = 1 + 0.9 + 0.81 + ...$$

We call this the **discounted return**

The discounted return lets makes us prefer rewards sooner, like humans

# Markov Decision Processes

For the rest of the course, we maximize the discounted return

$$\arg\max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg\max_{s \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

# Markov Decision Processes

For the rest of the course, we maximize the discounted return

$$\arg\max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg\max_{s \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

If our agent maximizes the discounted return, then it is **optimal**

# Markov Decision Processes

Let us review

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

- $S$ is the state space

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

- $S$ is the state space
- $A$ is the action space

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

- $S$ is the state space
- $A$ is the action space
- $\mathrm{Tr} : S \times A \mapsto \Delta S$ is the state transition function

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

- $S$ is the state space
- $A$ is the action space
- $\mathrm{Tr} : S \times A \mapsto \Delta S$ is the state transition function
- $R : S \mapsto \mathbb{R}$ is the reward function

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

- $S$ is the state space
- $A$ is the action space
- $\mathrm{Tr} : S \times A \mapsto \Delta S$ is the state transition function
- $R : S \mapsto \mathbb{R}$ is the reward function
- $\gamma \in [0, 1]$ is the discount factor

# Markov Decision Processes

Let us review

**Definition:** A Markov decision process (MDP) is a tuple $(S, A, T, R, \gamma)$

- $S$ is the state space
- $A$ is the action space
- $\mathrm{Tr} : S \times A \mapsto \Delta S$ is the state transition function
- $R : S \mapsto \mathbb{R}$ is the reward function
- $\gamma \in [0, 1]$ is the discount factor

# Markov Decision Processes

Reinforcement learning is designed to solve MDPs

# Markov Decision Processes

Reinforcement learning is designed to solve MDPs

In reinforcement learning, we have a single goal

# Markov Decision Processes

Reinforcement learning is designed to solve MDPs

In reinforcement learning, we have a single goal

Maximize the discounted return of the MDP

# Markov Decision Processes

Reinforcement learning is designed to solve MDPs

In reinforcement learning, we have a single goal

Maximize the discounted return of the MDP

$$\arg\max_{\boldsymbol{\tau}} G(\boldsymbol{\tau}) = \arg\max_{s \in S} \sum_{t=0}^{\infty} \gamma^t R(s_{t+1})$$

You must understand the discounted return!

# Markov Decision Processes

Understanding MDPs is the **most important part** of RL

# Markov Decision Processes

Understanding MDPs is the **most important part** of RL

Existing software can train RL agents on your MDP

# Markov Decision Processes

Understanding MDPs is the **most important part** of RL

Existing software can train RL agents on your MDP

You can train an RL agent without understanding RL

# Markov Decision Processes

Understanding MDPs is the **most important part** of RL

Existing software can train RL agents on your MDP

You can train an RL agent without understanding RL

You can only train an agent if you can model your problem as an MDP

# Markov Decision Processes

Understanding MDPs is the **most important part** of RL

Existing software can train RL agents on your MDP

You can train an RL agent without understanding RL

You can only train an agent if you can model your problem as an MDP

Make sure you understand MDPs!

# Exercise

# Exercise


Design a Super Mario Bros MDP

# Exercise


Design a Super Mario Bros MDP

- State space $S$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$
- Reward function $R$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$
- Reward function $R$
- Discount factor $\gamma$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$
- Reward function $R$
- Discount factor $\gamma$

Compute discounted return for:

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$
- Reward function $R$
- Discount factor $\gamma$

Compute discounted return for:

- Eat mushroom at $t = 10$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$
- Reward function $R$
- Discount factor $\gamma$

Compute discounted return for:

- Eat mushroom at $t = 10$
- Collect coins at $t = 11, 12$

# Exercise


Design a Super Mario Bros MDP

- State space $S$
- Action space $A$
- State transition function $\mathrm{Tr}$
- Reward function $R$
- Discount factor $\gamma$

Compute discounted return for:

- Eat mushroom at $t = 10$
- Collect coins at $t = 11, 12$
- Die to bowser at $t = 20$
- Game over screen at $t = 21...\infty$

# Coding

# Coding

In this course, we will implemented MDPs using **gymnasium**

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

Must define:

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

Must define:

- state space ($S$)

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

Must define:

- state space ($S$)
- action space ($A$)

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

Must define:

- state space ($S$)
- action space ($A$)
- step ($\mathrm{Tr}$, $R$, terminated)

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

Must define:

- state space ($S$)
- action space ($A$)
- step ($\mathrm{Tr}$, $R$, terminated)
- reset ($s_0$)

# Coding

In this course, we will implemented MDPs using **gymnasium**

Originally developed by OpenAI for reinforcement learning

Gymnasium provides an **environment** (MDP) API

Must define:

- state space ($S$)
- action space ($A$)
- step ($\mathrm{Tr}$, $R$, terminated)
- reset ($s_0$)

https://gymnasium.farama.org/api/env/

# Coding

Gymnasium uses **observations** instead of **states**

# Coding

Gymnasium uses **observations** instead of **states**

**Question:** What was the Markov condition for MDPs?

# Coding

Gymnasium uses **observations** instead of **states**

**Question:** What was the Markov condition for MDPs?

The next Markov state only depends on the current Markov state

# Coding

Gymnasium uses **observations** instead of **states**

**Question:** What was the Markov condition for MDPs?

The next Markov state only depends on the current Markov state

$$\Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_1) = \Pr(s_{t+1} \mid s_t)$$

# Coding

Gymnasium uses **observations** instead of **states**

**Question:** What was the Markov condition for MDPs?

The next Markov state only depends on the current Markov state

$$\Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_1) = \Pr(s_{t+1} \mid s_t)$$

If the Markov property is broken, $s_t \in S$ is not a Markov state

# Coding

Gymnasium uses **observations** instead of **states**

**Question:** What was the Markov condition for MDPs?

The next Markov state only depends on the current Markov state

$$\Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_1) = \Pr(s_{t+1} \mid s_t)$$

If the Markov property is broken, $s_t \in S$ is not a Markov state

Then, we change $s_t \in S$ to an **observation** $o_t \in O$ (more later)

# Coding

Gymnasium uses **observations** instead of **states**

**Question:** What was the Markov condition for MDPs?

The next Markov state only depends on the current Markov state

$$\Pr(s_{t+1} \mid s_t, s_{t-1}, ..., s_1) = \Pr(s_{t+1} \mid s_t)$$

If the Markov property is broken, $s_t \in S$ is not a Markov state

Then, we change $s_t \in S$ to an **observation** $o_t \in O$ (more later)

Gymnasium uses observations, but for MDPs we treat them as states

# Coding

```
import gymnasium as gym

MyMDP(gym.Env):
  def __init__(self):
    self.action_space = gym.spaces.Discrete(3) # A
    self.observation_space = gym.spaces.Discrete(5) # S

  def reset(self, seed=None) -> Tuple[Observation, Dict]

  def step(self, action) -> Tuple[
    Observation, Reward, Terminated, Truncated, Dict
  ]
```

# Coding

https://colab.research.google.com/drive/1rDNik5oRl27si8wdtMLE7Y41U5J2bx-I#scrollTo=9pOLI5OgKvoE

# Exam Next Class

# Exam Next Class

We will have an exam next week

# Exam Next Class

We will have an exam next week

1 hour 15 minutes, no coding, only math

# Exam Next Class

We will have an exam next week

1 hour 15 minutes, no coding, only math

No book, no notes, no calculator – only pencil and pen

# Exam Next Class

We will have an exam next week

1 hour 15 minutes, no coding, only math

No book, no notes, no calculator – only pencil and pen

Study notation, probability, bandits, and Markov processes

# Exam Next Class

We will have an exam next week

1 hour 15 minutes, no coding, only math

No book, no notes, no calculator – only pencil and pen

Study notation, probability, bandits, and Markov processes

Practice expectations, bandit problems, state transitions, and returns

# Exam Next Class

We will have an exam next week

1 hour 15 minutes, no coding, only math

No book, no notes, no calculator – only pencil and pen

Study notation, probability, bandits, and Markov processes

Practice expectations, bandit problems, state transitions, and returns

You must have intuition, not memorize

# Exam Next Class

We will have an exam next week

1 hour 15 minutes, no coding, only math

No book, no notes, no calculator – only pencil and pen

Study notation, probability, bandits, and Markov processes

Practice expectations, bandit problems, state transitions, and returns

You must have intuition, not memorize

Too many A’s last term, exam will be **difficult**