Mahindra™ University
Global Thinkers. Engaged Leaders.

Reinforcement Learning and Autonomous Systems (CS4122)

Lecture 10 (03/09/2024)

Instructor: Gourav Saha

Lecture Content

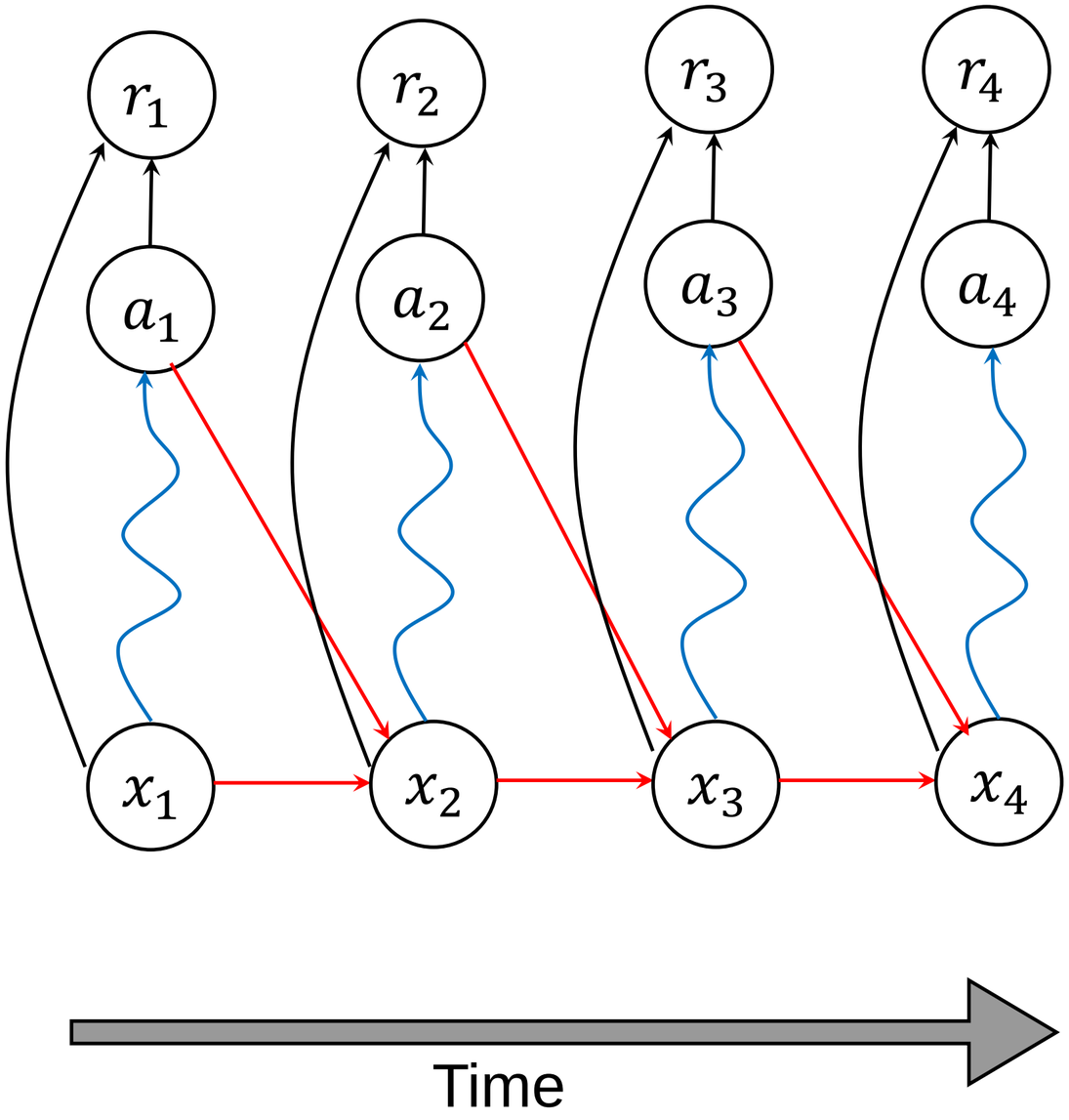
(This lecture is the beginning of Module 2)

- Setup of Markov Decision Process.
- Example of Markov Decision Process.
- Episodic vs Continuing Tasks.

Lecture Content

- Setup of Markov Decision Process.
- Example of Markov Decision Process.
- Episodic vs Continuing Tasks.

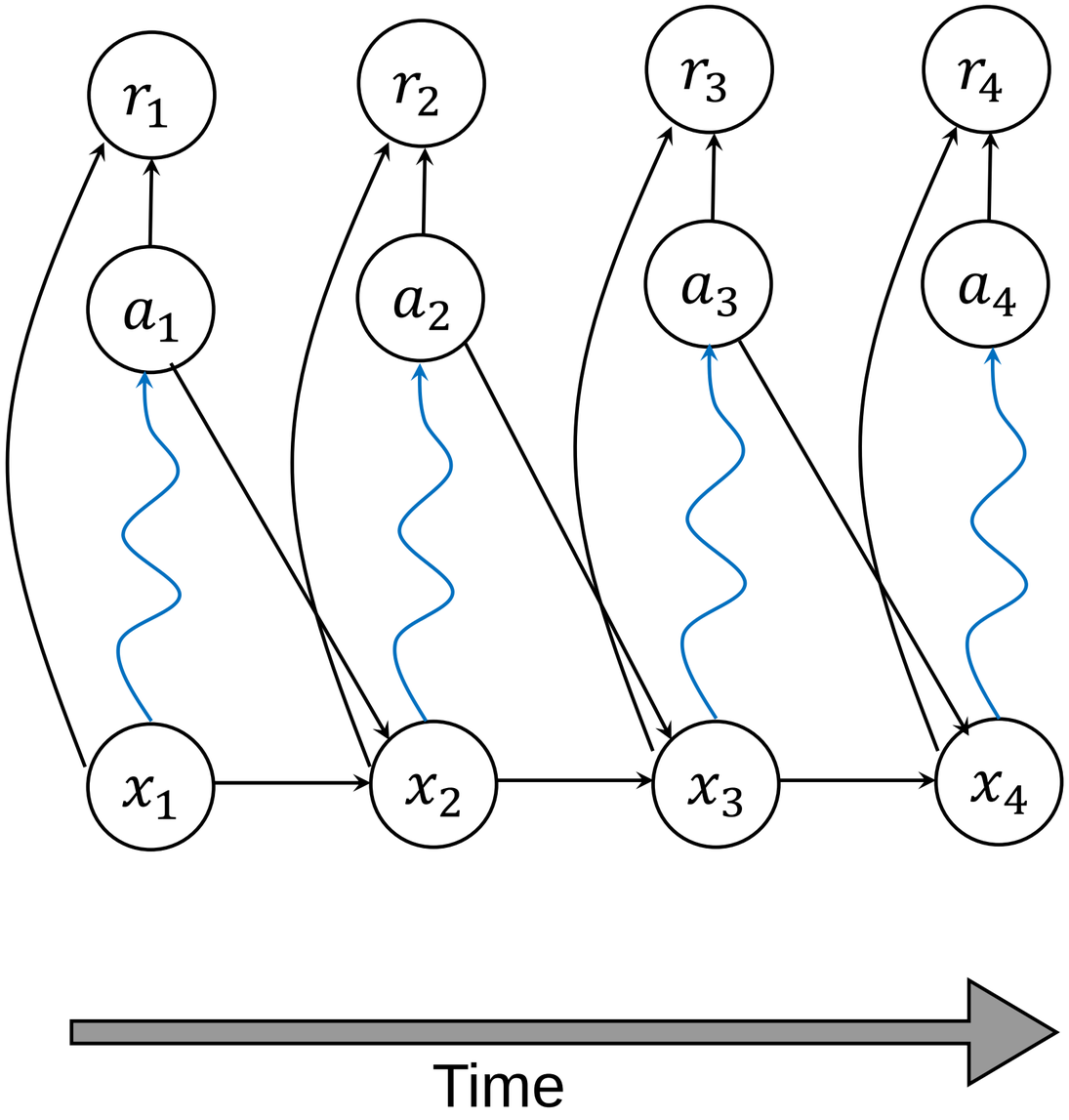
Setup of Markov Decision Process



- Recall the Probabilistic Graphical Model of Markov Decision Process (shown in the left) from Lecture 2 and 3 slides.

- Recall that MDP is a generalization of Contextual Bandit. The difference between MDP and Contextual Bandit are:
  - The arrows in red (not there in contextual bandit but there in MDP).
  - These arrows suggest that the agent's current actions has effect on the future states of the environment (**temporal effect**) and hence the future reward. This idea is sometime also called the **"sequential nature of decision making"**.
  - Therefore **greedy policies** to maximize rewards **may not optimal**.

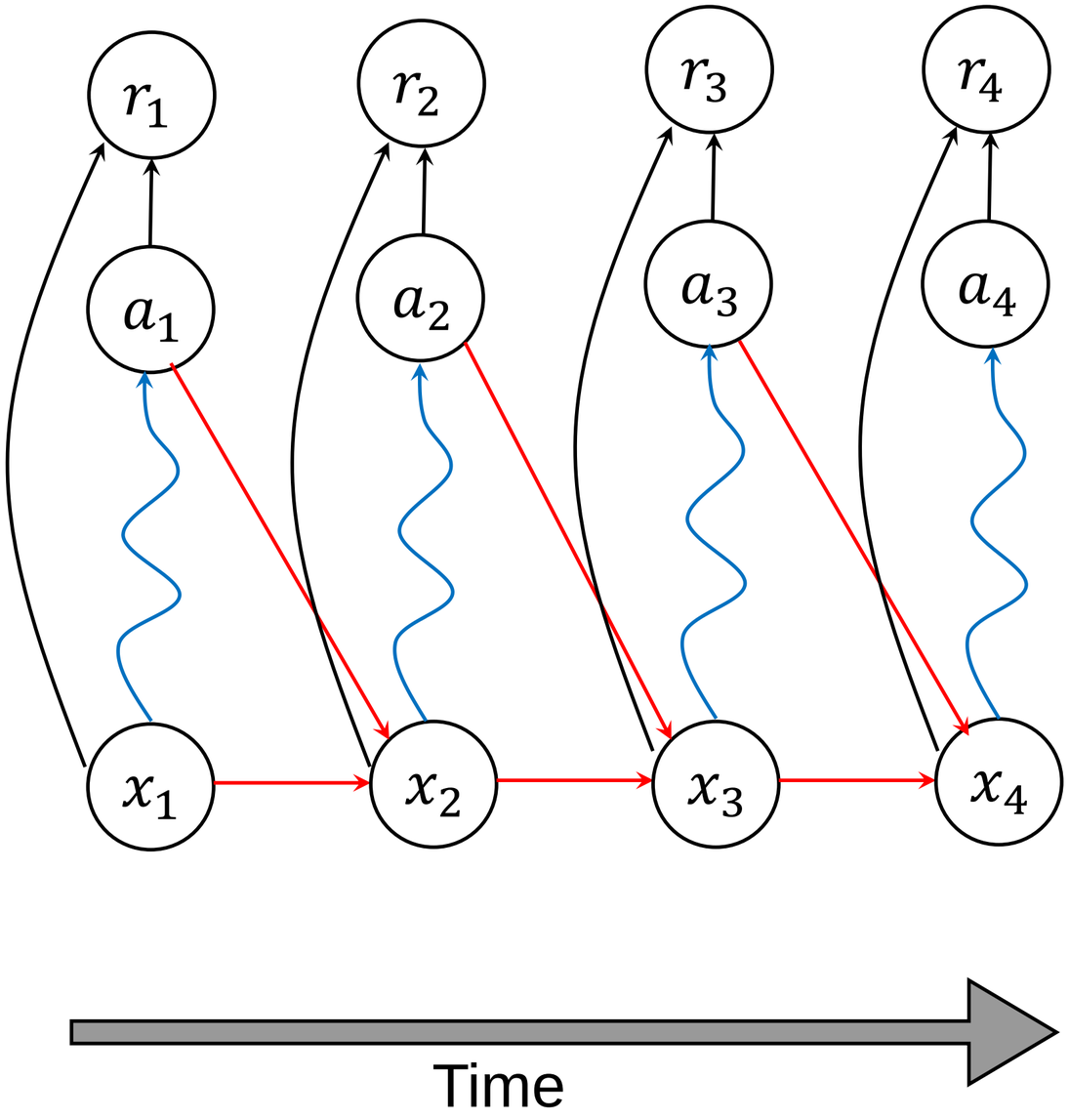
Setup of Markov Decision Process



An MDP is defined by:

- x_t is the **state** of the system at time slot t.
  - The set of all states, also called **state space**, is denoted using S. We have $x_t \in S\ ;\ \forall t$.
- a_t is the **action** (made by an agent) at time slot t.
  - The set of action, also called **action space**, corresponding to state x is denotes using $\mathcal{A}(x)$. We have $a_t \in \mathcal{A}(x_t);\ \forall t$.
  - Note that the **action space is a function of state**.
- r_t is the **reward** (received by the agent) at time slot t.

Setup of Markov Decision Process



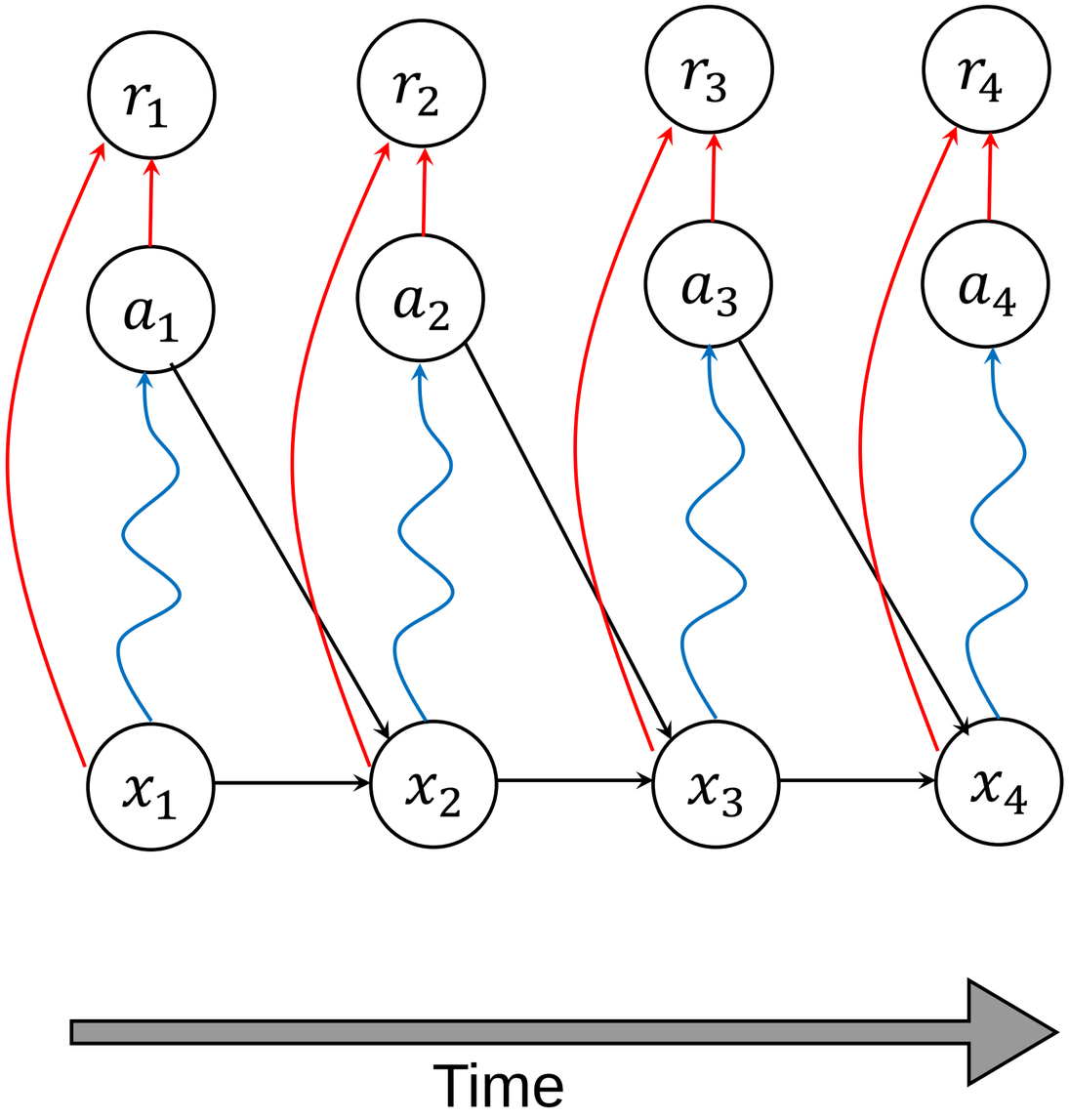
An MDP is defined by:

- **State transition probability**, $P[x'|x,a]$, which is the probability of the next state being x' given that the current state is x and current action is a.
  - In the PGM shown in the left, state transition probability is captured using the red arrows.
  - Not that state transition follows Markovian property, i.e.

$$P[x_{t+1}|x_t, a_t, x_{t-1}, a_{t-1}, x_{t-2}, a_{t-2}, \cdots]$$

$$= P[x_{t+1}|x_t, a_t]$$

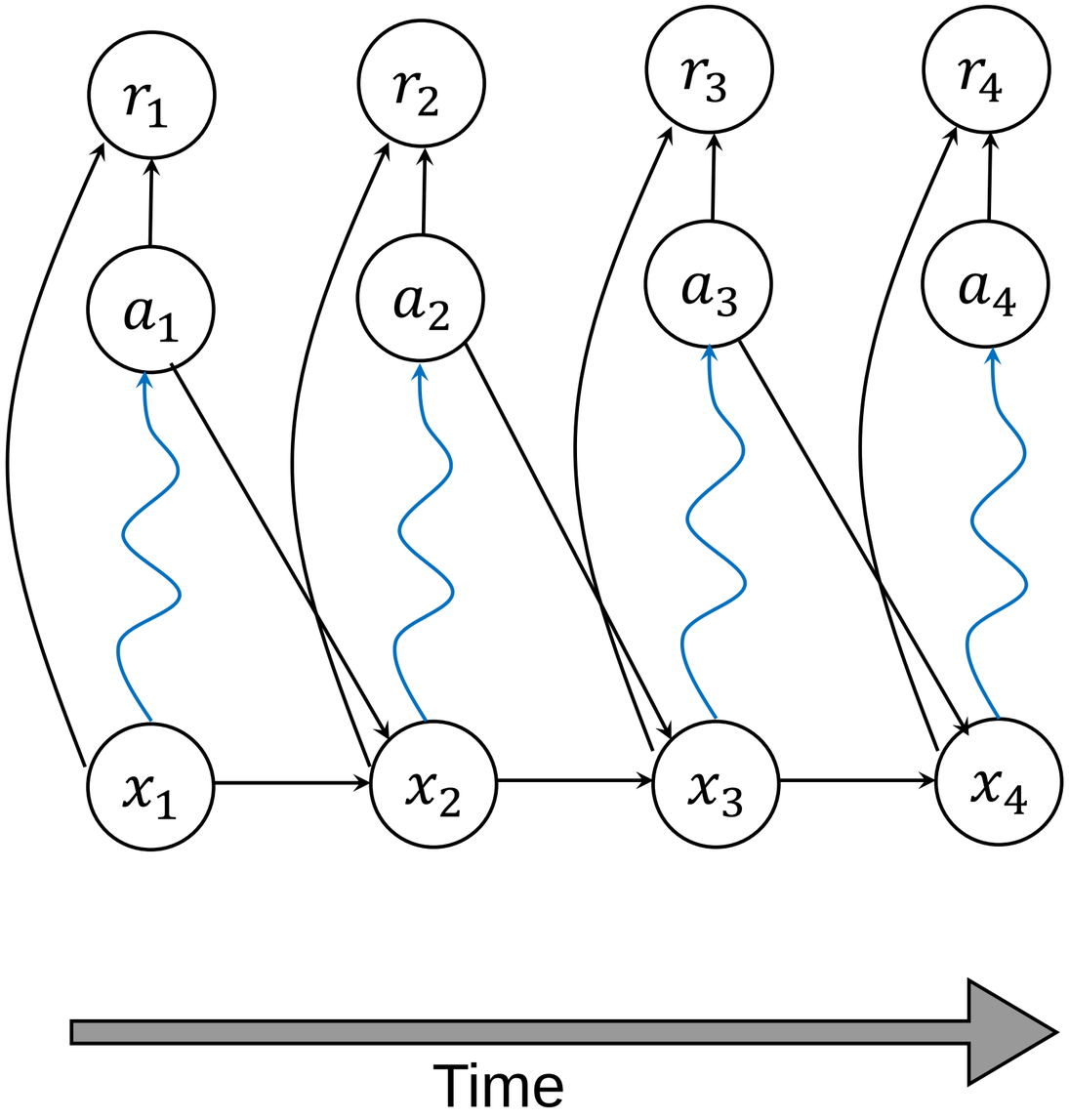
Setup of Markov Decision Process



An MDP is defined by:

- **State transition probability**, $P[x'|x,a]$, which is the probability of the next state being x' given that the current state is x and current action is a.
- **Reward probability**, $P[r|x,a]$, which is the probability of the current reward being r given that the current state is x and current action is a.
  - In the PGM shown in the left, reward probability is captured using the red arrows.

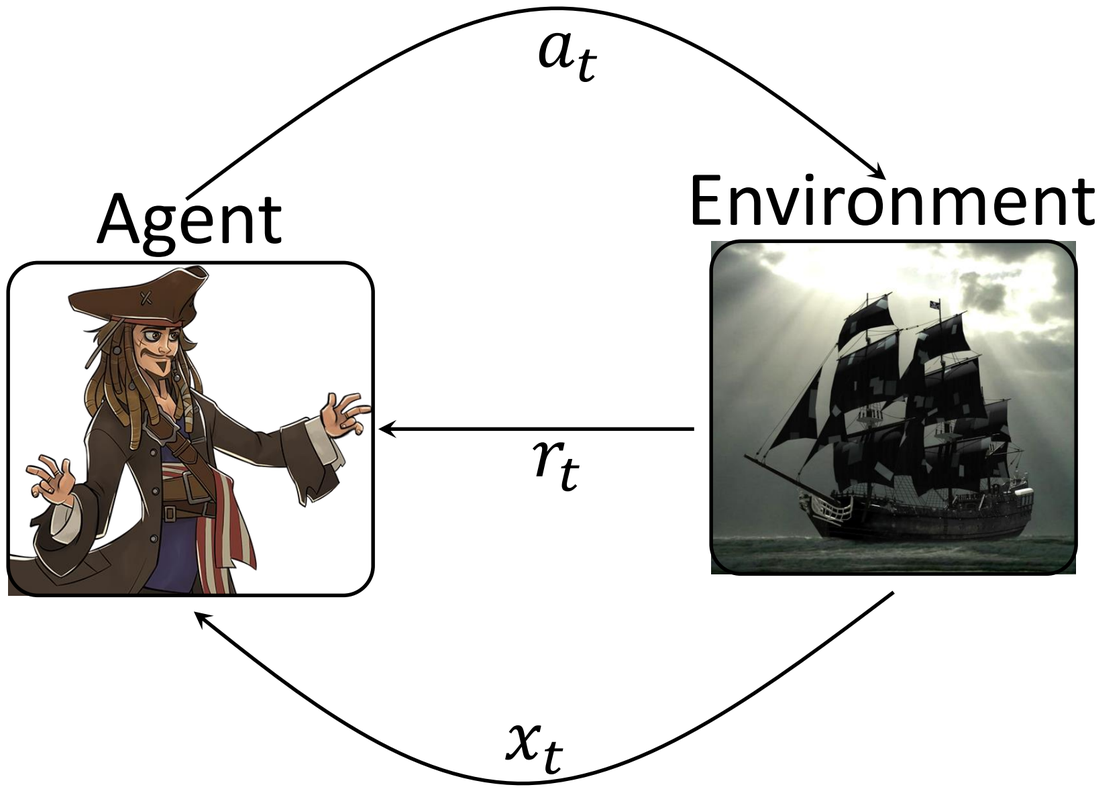
Setup of Markov Decision Process



To put everything in one slide, an MDP is defined by:

- States and the associate state space.
- Actions and the associated actions space.
- Reward.
- State transition probability.
- Reward probability.

Setup of Markov Decision Process



- In time slot t:

  Step 1: The agent observes the state x_t.

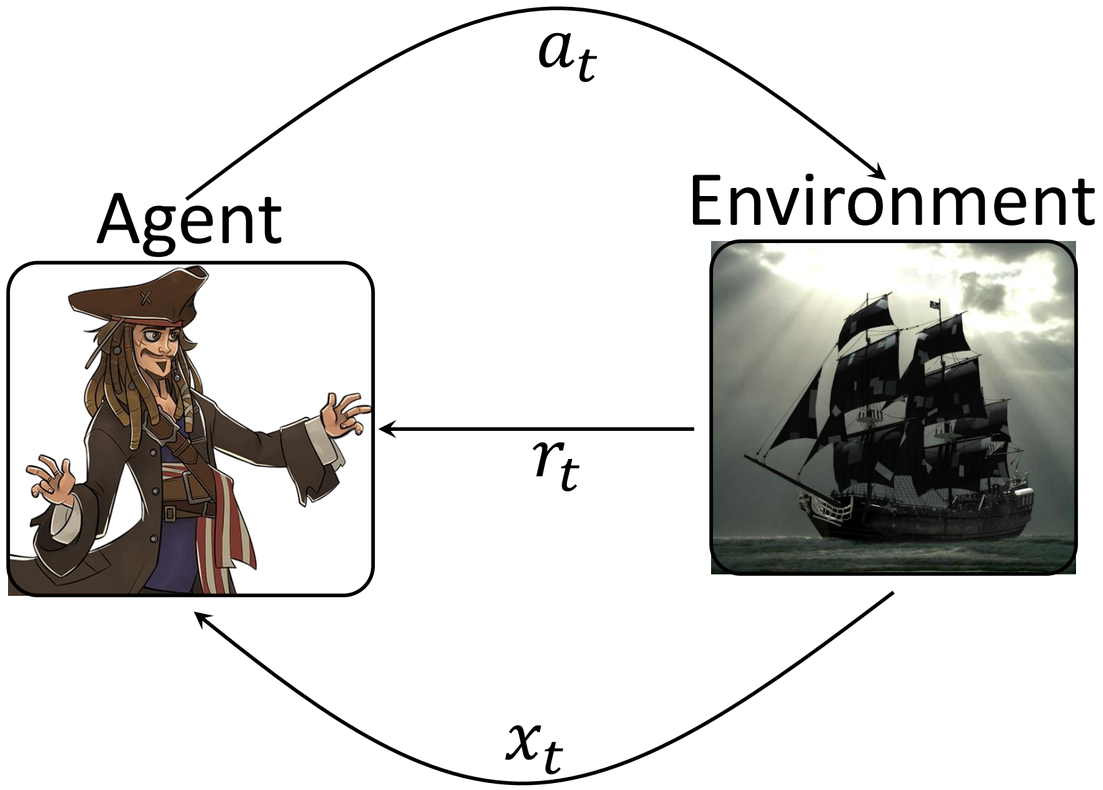
  Step 2: The agent takes action a_t based on x_t.

  Step 3: The agent receives a reward r_t where r_t is sampled from the reward probability distribution $P[r|x_t, a_t]$.

  Step 4: The state of the environment for time slot $t+1$ changes to x_{t+1} where x_{t+1} is sampled from the state transition probability distribution $P[x'|x_t, a_t]$.

Steps 3 and 4 happens simultaneously.

Setup of Markov Decision Process



- In time slot t:

  Step 1: The agent observes the state x_t.

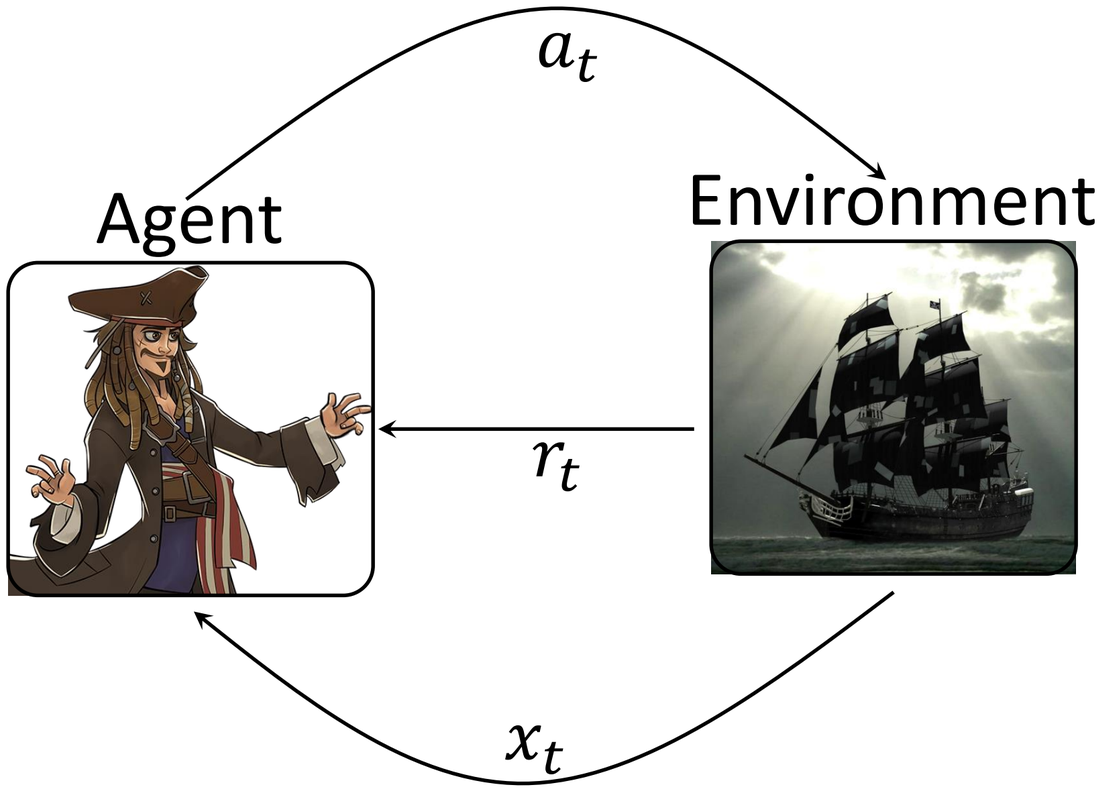
  Step 2: The agent takes action a_t based on x_t.

  Step 3: The agent receives a reward r_t where r_t is sampled from the reward probability distribution $P[r|x_t, a_t]$.

  Step 4: The state of the environment for time slot $t+1$ changes to x_{t+1} where x_{t+1} is sampled from the state transition probability distribution $P[x'|x_t, a_t]$.

One thing I forgot to tell right in the beginning (like lecture 2): For both MDP and bandits, time slot t can be interpreted more generally as "decision epochs" rather than **equally spaced time** steps, e.x. in chess.

Setup of Markov Decision Process



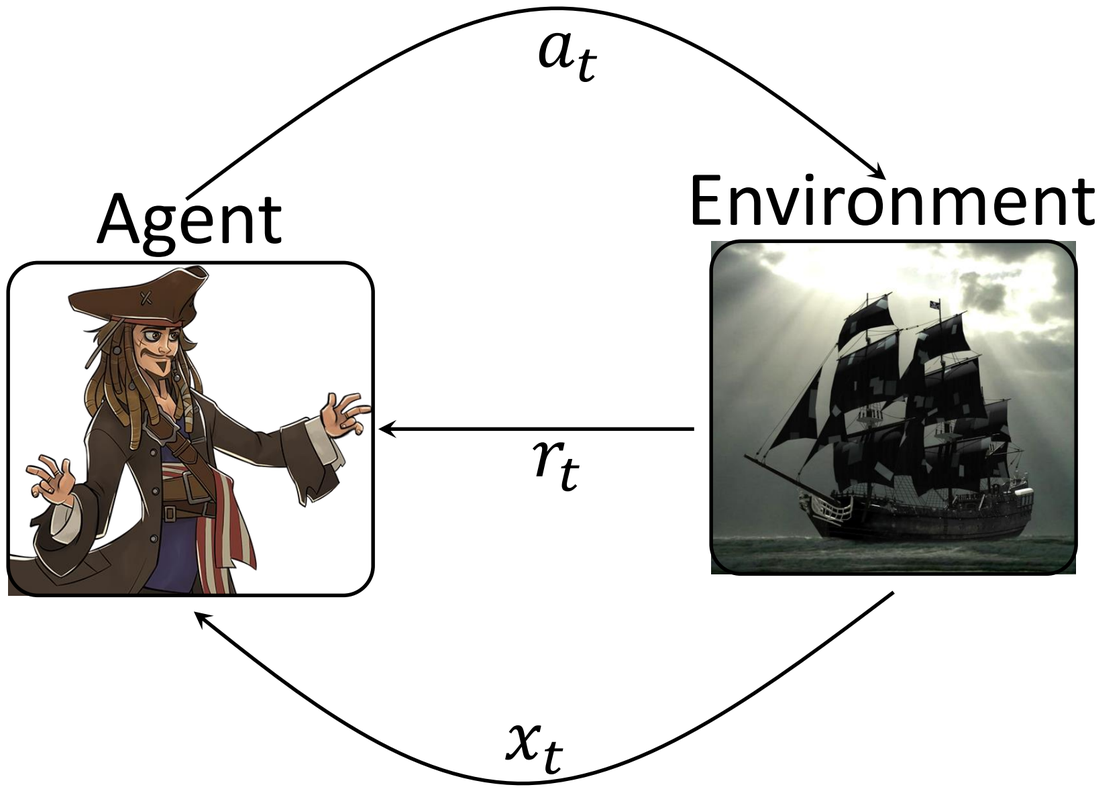
While making decisions, the objective of the agent is to maximize some form of net reward over an optimization horizon.

Objective 1: Sum of rewards

$$R_{sum} = \sum_{t=0}^{T} r_t$$

- Suitable when time horizon T is **finite**.
- It is possible that ***T* is a random variable**. E.x. when balancing an inverted pendulum on a cart, T is the time slot when the angle of pendulum is greater than a threshold (indicating it is about to fall). In this example, probabilistically speaking, T is finite but it s random variable.

Setup of Markov Decision Process



While making decisions, the objective of the agent is to maximize some form of net reward over an optimization horizon.

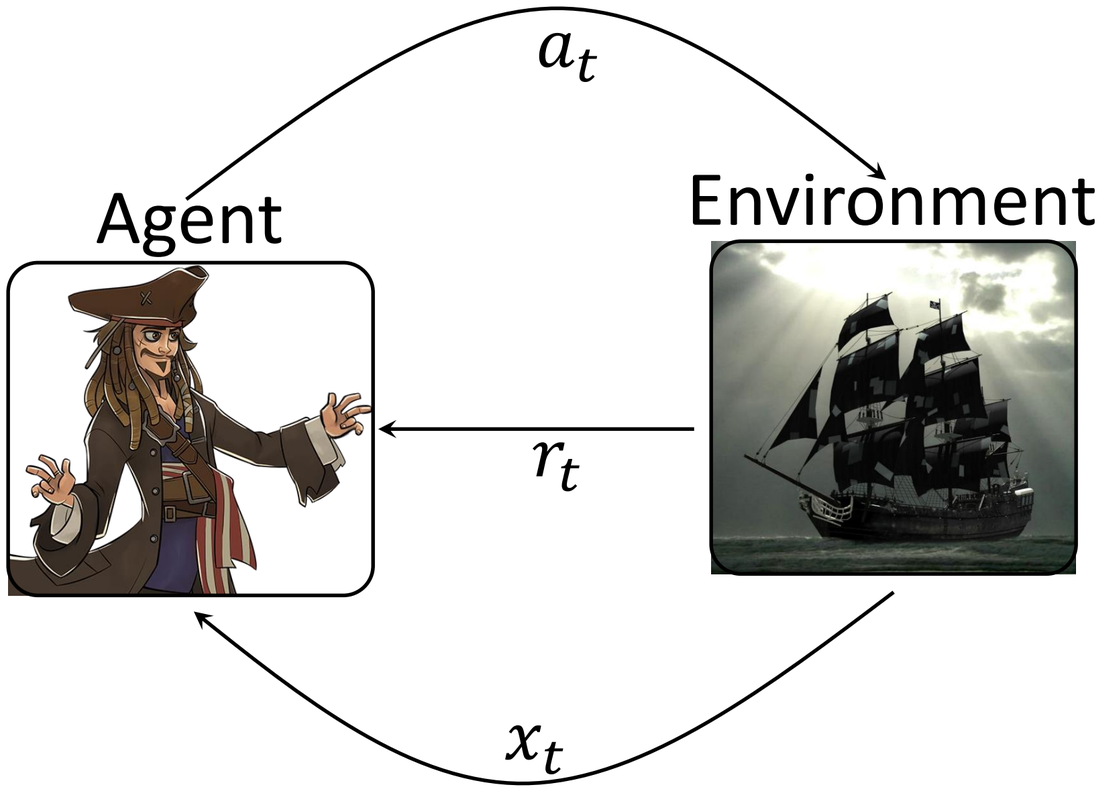
Objective 2: Discounted reward

$$R_{discounted} = \sum_{t=0}^{\infty} \beta^t r_t$$

where $\beta \in (0,1)$ is the discount factor.

- Suitable when time horizon T is **infinite**.

Setup of Markov Decision Process



Module 2 vs modules 3, 4, and 5

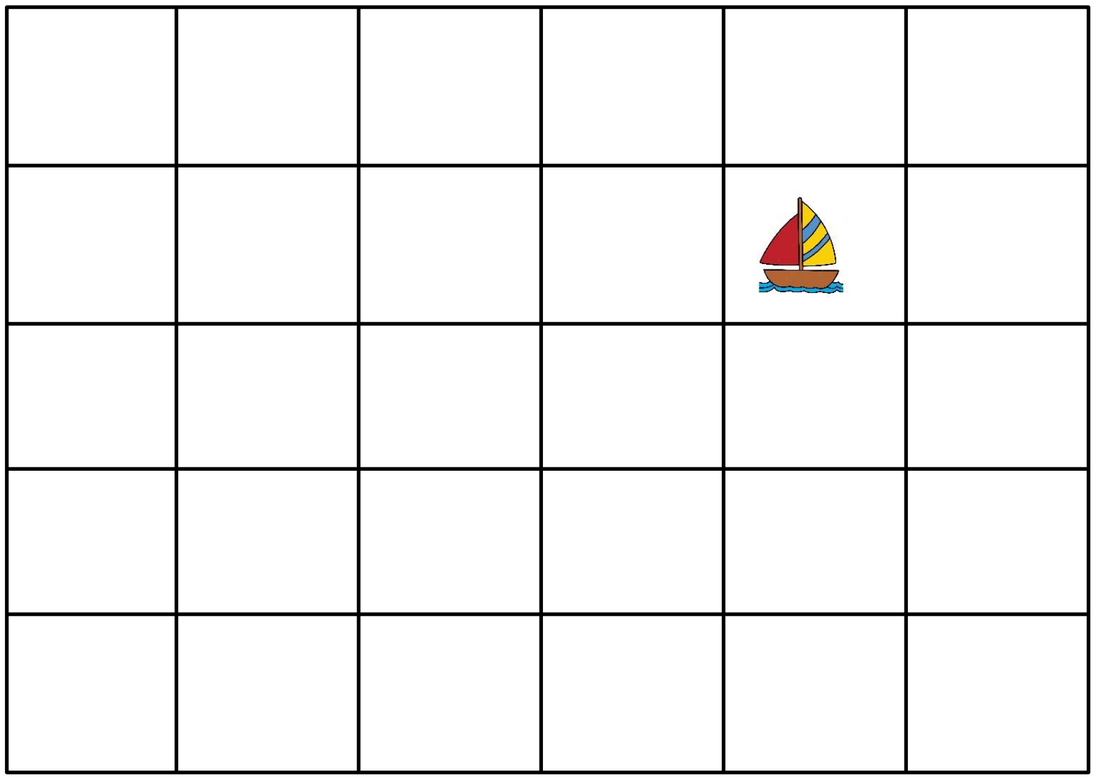
- MDP is associated with the following probability distribution:
  - State transition probability, $P[x'|x,a]$.
  - Reward probability, $P[r|x,a]$.
- In module 2, we assume that these two probability distributions are known to the agent. So it is a **planning problem** (planning involves taking actions that are good for the long term; not be greedy).
- In modules 3, 4, and 5, these two probability distributions are NOT known to the agent. So it is a **learning+planning problem**. This is called **reinforcement learning**.

Lecture Content

- Setup of Markov Decision Process.
- Example of Markov Decision Process.
- Episodic vs Continuing Tasks.

Example of Markov Decision Process

Fishing in Gridworld



- Consider a boat sailing in river to catch fish.
- In order to make this example suitable for "discrete state" MDP*, we "gridify" the entire area of interest of the river as shown in the figure.
- The boat can move up, down, left , and right.
- The waves and the winds leads to the random nature (stochasticity) in boat's movement.
- Every grid has different amount of fish. Even in the same grid, different amount of fish will be caught in different time. Hence, the amount of fish caught is random (stochasticity).
- The idea is to maximize the amount of fish caught in a time horizon.

* Modules 2 and 3 can only deal with discrete state MDP.

Example of Markov Decision Process

Fishing in Gridworld

(0.5, 3.0)	(1.0, 3.0)				(3.0, 3.0)
					
(0.5, 1.0)	(1.0, 1.0)				(3.0, 1.0)
(0.5, 0.5)	(1.0, 0.5)				(3.0, 0.5)

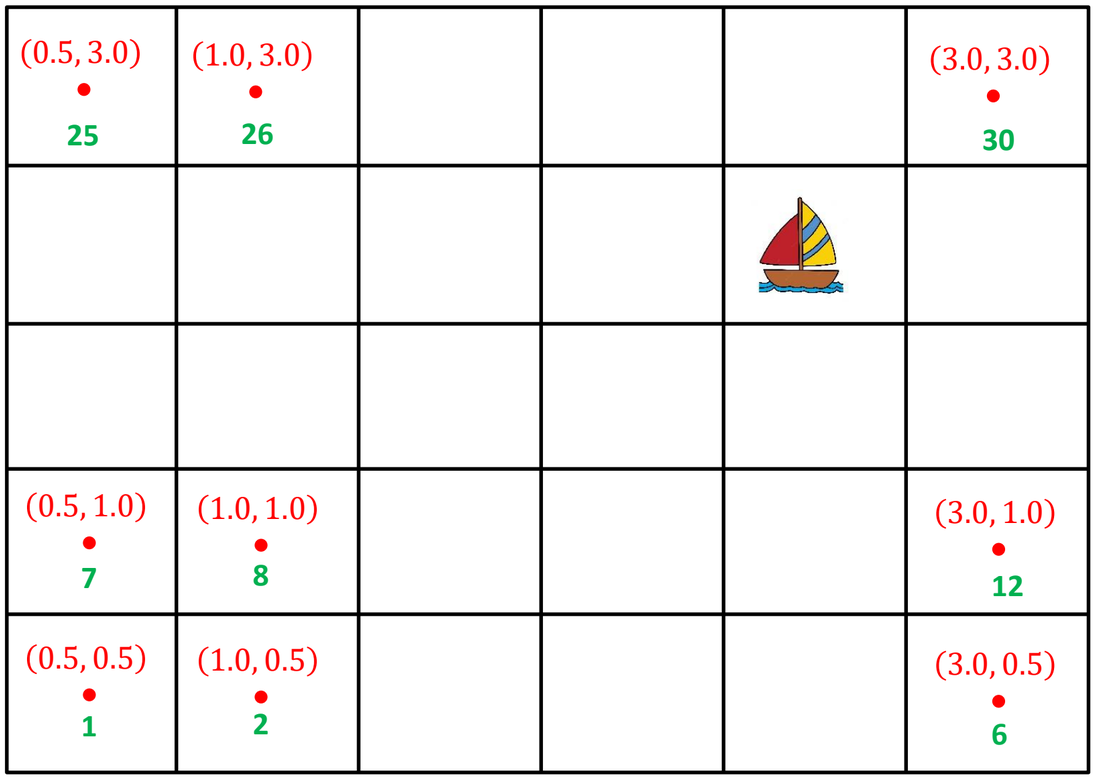
States and state space

- States is (x, y) where x and y are the coordinates the midpoint of the grid the boat is in at a given time slot.
- State space is,

$$S = \{(0.5,0.5), (1.0,0.5), \cdots, (3.0,3.0)\}$$

Example of Markov Decision Process

Fishing in Gridworld



States and state space

To be noted:

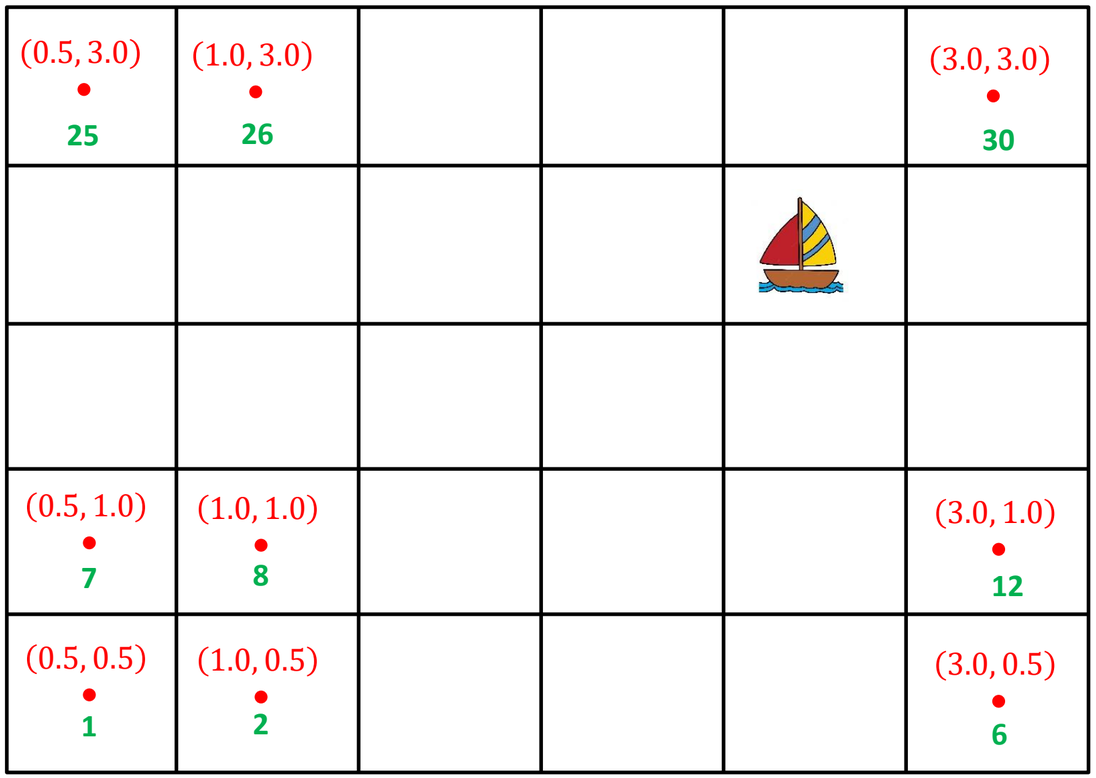
- **Point 1:** As mentioned, modules 2 and 3 deals with discrete state space. But discrete state space **doesn't necessarily mean state space consists of integer** elements only. It just means that we can **enumerate** the elements of the state space with a finite set of integers. For e.x. the state space,

$$S = \{(0.5,0.5), (1.0,0.5), \cdots, (3.0,3.0)\}$$
$$\qquad\quad 1 \qquad\quad 2 \qquad\qquad 30$$

As we can see, we can enumerate the states from 1 to 30 (because there are 30 grids). Hence, this is a discrete state space.

Example of Markov Decision Process

Fishing in Gridworld



States and state space

To be noted:

- **Point 2:** States may not be scalar. **States can be a vector** as well. For e.x. for fishing in gridworld, the state is (x, y) which is a vector.

- **Point 3:** Even is we are dealing with states that are vectors, we can interpret them as scalars by considering the "**enumerated version**" of that state.

$$S = \{\underset{1}{(0.5,0.5)}, \underset{2}{(1.0,0.5)}, \cdots, \underset{30}{(3.0,3.0)}\}$$

For e.x., state $(1.0,0.5)$ can be interpreted as state 2. In fishing in gridworld, "enumerated version" of the state is simply the **grid index**.

Example of Markov Decision Process

Fishing in Gridworld

(0.5, 3.0) • 25	(1.0, 3.0) • 26				(3.0, 3.0) • 30
					
(0.5, 1.0) • 7	(1.0, 1.0) • 8				(3.0, 1.0) • 12
(0.5, 0.5) • 1	(1.0, 0.5) • 2				(3.0, 0.5) • 6

States and state space

To be noted:

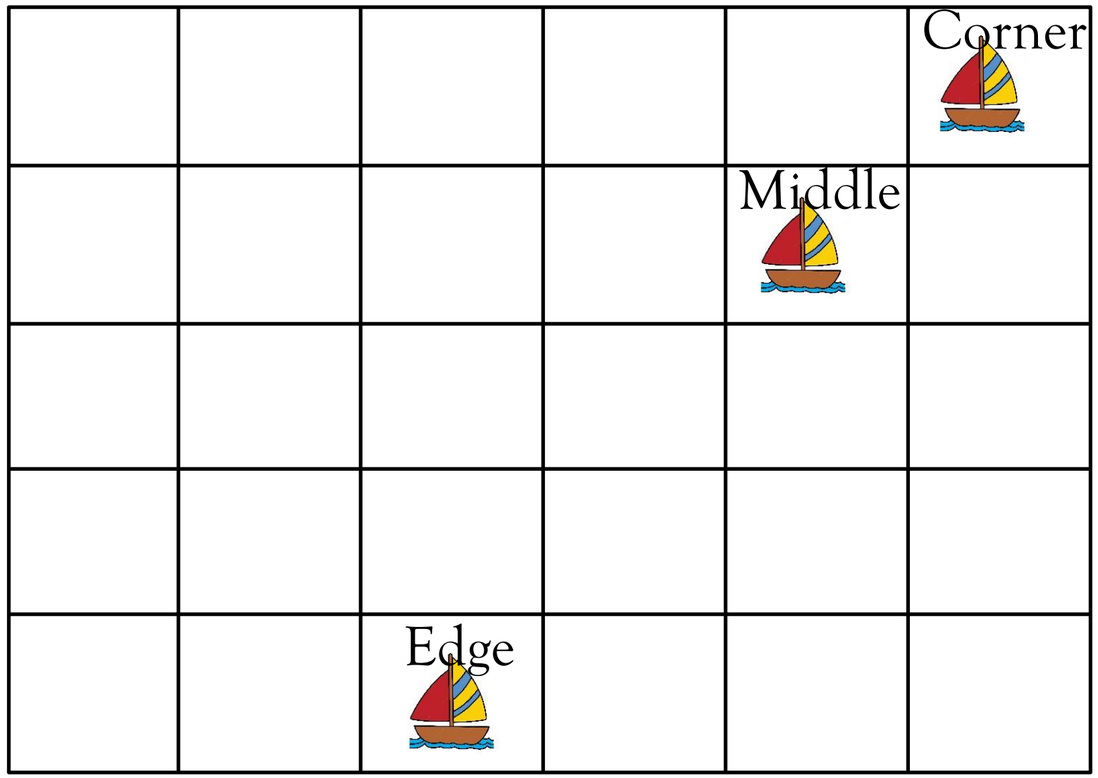
- **Point 4:** For some problems, we can have multiple state representations (just like this examples where state can be x and y coordinate or the grid index). But, some state space representation leads to a simple problem formulation.

  For. e.x.: In this example, the state transition equations are easy to write if we are dealing with x and y coordinate representation of state. This is because we can simply write $(x, y + 0.5)$ for a "up transition" or $(x - 0.5, y)$ left transition.

  But, if we are dealing with grid index version of states, we have to define another function that tells which is the "up grid" or the "left grid" of a particular grid.

Example of Markov Decision Process

Fishing in Gridworld

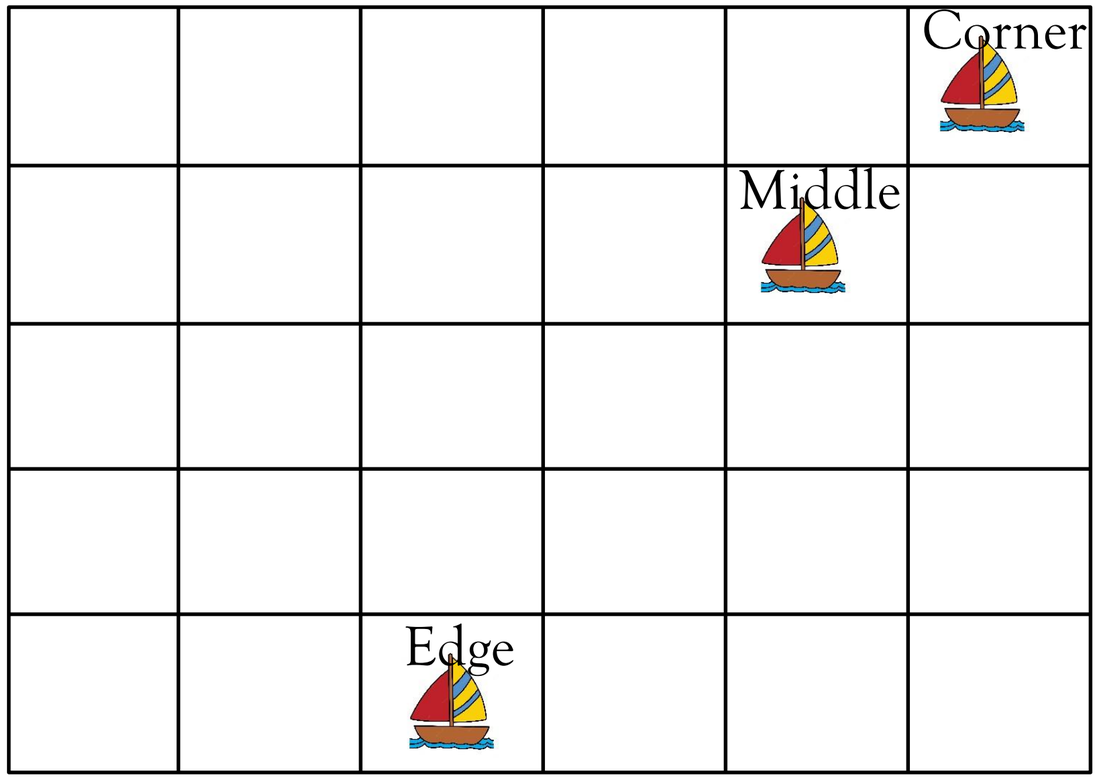


Actions and action space

- Action is a where a is which direction to move in a given time slot.
- Action space depends on the grid the boat is in. For the "middle", "edge", and "corner" grid shown in the figure, the action space are as follows:
  - $\mathcal{A}(middle) = \{up, down, left, right\}$.
  - $\mathcal{A}(edge) = \{up, left, right\}$.
  - $\mathcal{A}(corner) = \{down, left\}$.

Example of Markov Decision Process

Fishing in Gridworld



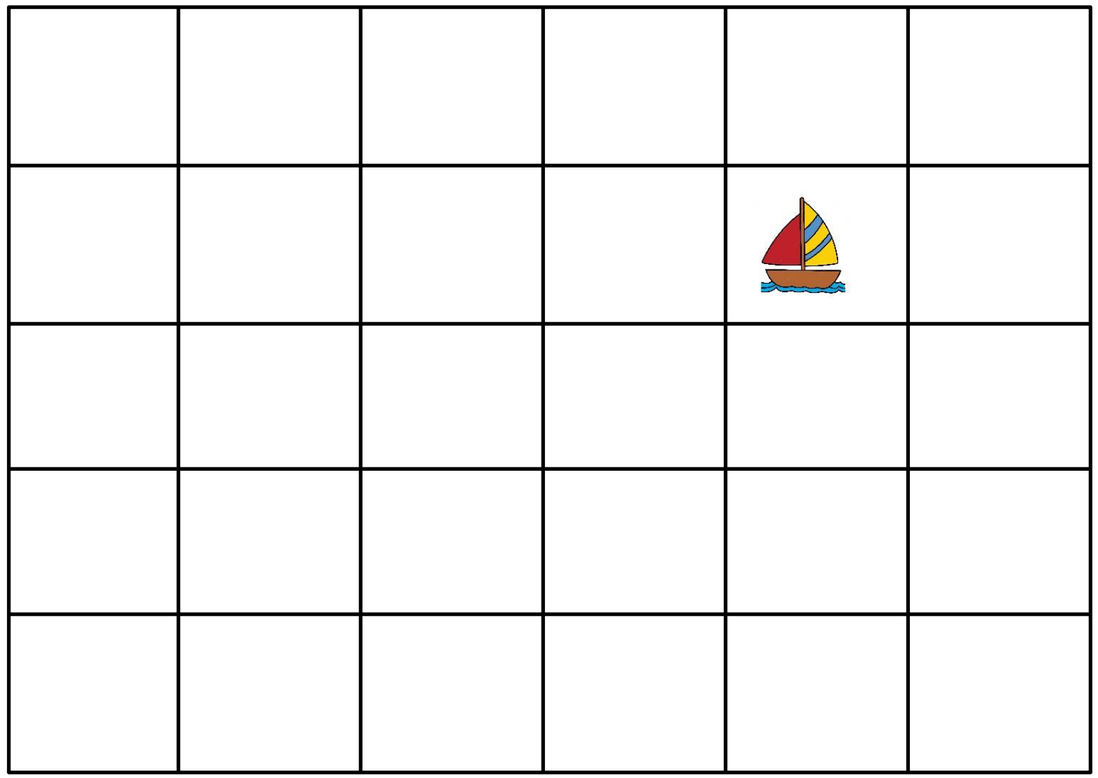
Actions and action space

To be noted: Just like states and state space,

- **Actions can be a vector**, e.x. for fishing in gridworld, action can be *(direction, speed)* instead of just direction.
- **Action space must be discrete** for most part of this course. Not just for modules 2 and 3, for almost all the modules. We will not deal with continuous action space (except may be an optional extra lecture on deep deterministic policy gradient in module 4).
- Even if actions are vectors, we can **enumerate** them.

Example of Markov Decision Process

Fishing in Gridworld



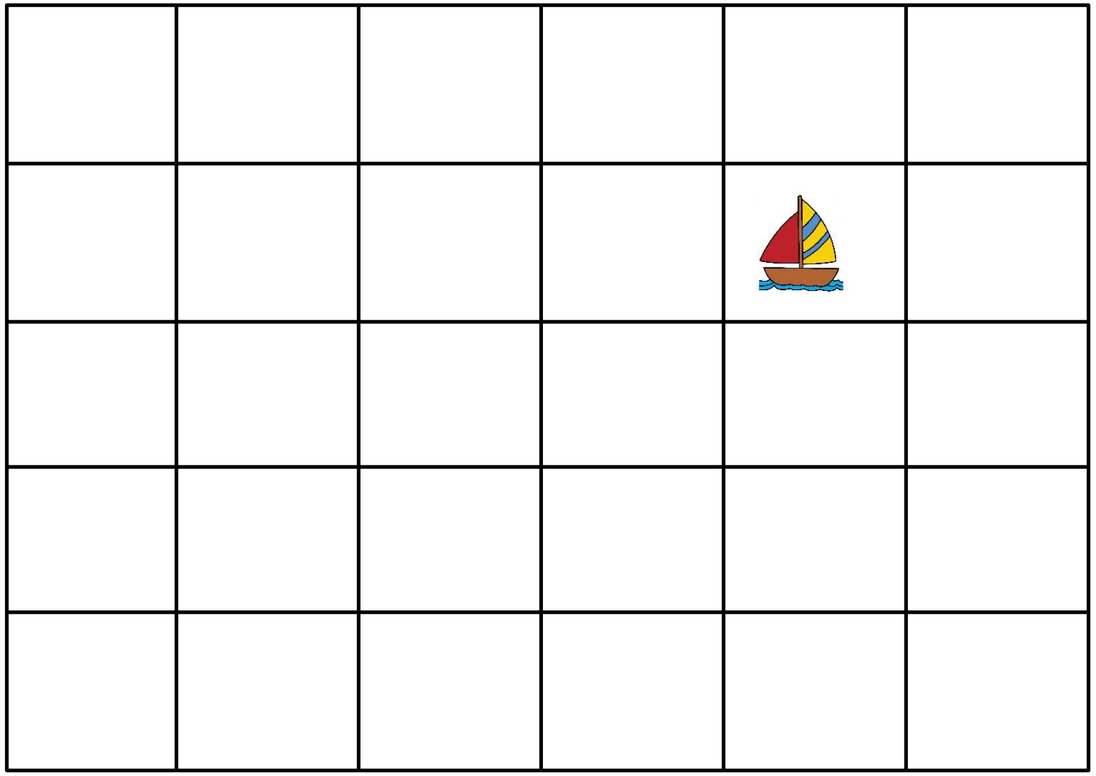
Reward and reward probability distribution

- Reward r is the amount of fish caught in a given time slot.
- Reward probability distribution: The probability of catching r kgs of fish in grid (x, y) is $\theta_{x,y,m}$ where $r \in \{0,1,\cdots,M\}$, M being the maximum amount of fish that can be caught in any time slot. So, the reward probability distribution is,

$$P[r|x, y, a] = \theta_{x,y,r}\ , \forall(x, y), \forall r \in \{0,1,\cdots,M\}$$

Example of Markov Decision Process

Fishing in Gridworld



Reward and reward probability distribution

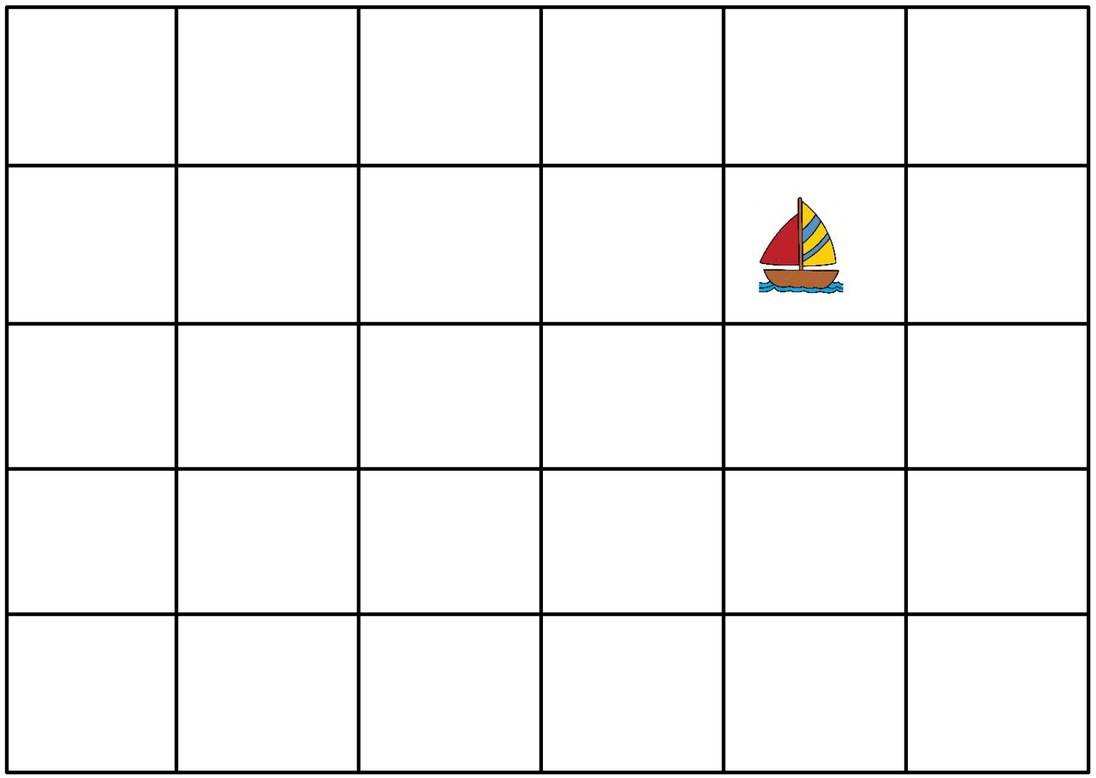
- Reward r is the amount of fish caught in a given time slot.
- Reward probability distribution: The probability of catching r kgs of fish in grid (x, y) is $\theta_{x,y,m}$ where $r \in \{0,1,\cdots,M\}$, M being the maximum amount of fish that can be caught in any time slot. So, the reward probability distribution is,

$$P[r|x, y, a] = \theta_{x,y,r}\ , \forall(x, y), \forall r \in \{0,1,\cdots,M\}$$

NOTE: For this example, the reward does not depend on action a. But in general, reward depends on the action as well. E.x. If the action also included both speed and direction and we had to account for fuel cost which in turn depends on speed. In this case, reward is the amount of fish caught minus fuel cost (we can add a weight to fuel cost).

Example of Markov Decision Process

Fishing in Gridworld



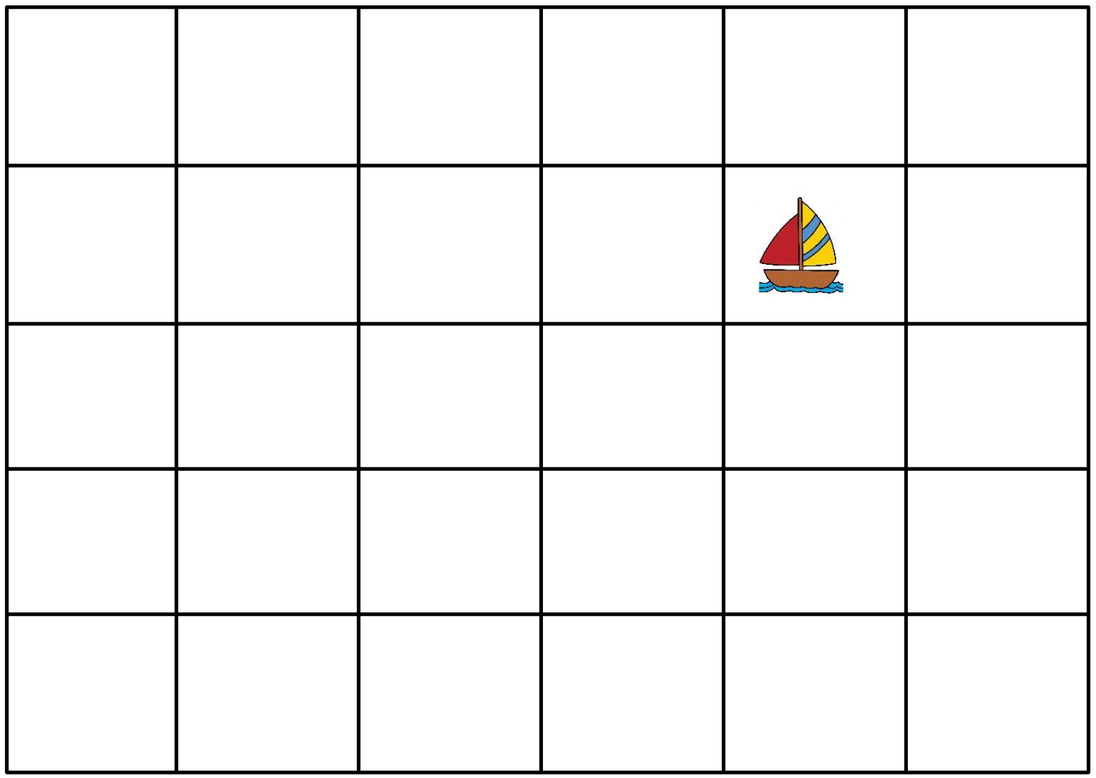
State transition probability distribution

- The state transition in this example will be governed by:
  - Which grid we are at. This is because the water current, waves, and the wind depends on the location.
  - What action the boat takes. If the boat chooses the “up” action, it will go up with a certain probability and will stay in the current grid itself with a certain probability. We are using a probabilistic model because water forces and wind are highly unpredictable.

NOTE: This is just the model we choose for this example. In general, the boat can drift anywhere no matter what action it chooses if the water current is too strong.

Example of Markov Decision Process

Fishing in Gridworld



State transition probability distribution

- The state transition probability distribution for this example can be written using two different approaches both of which are **equivalent**.

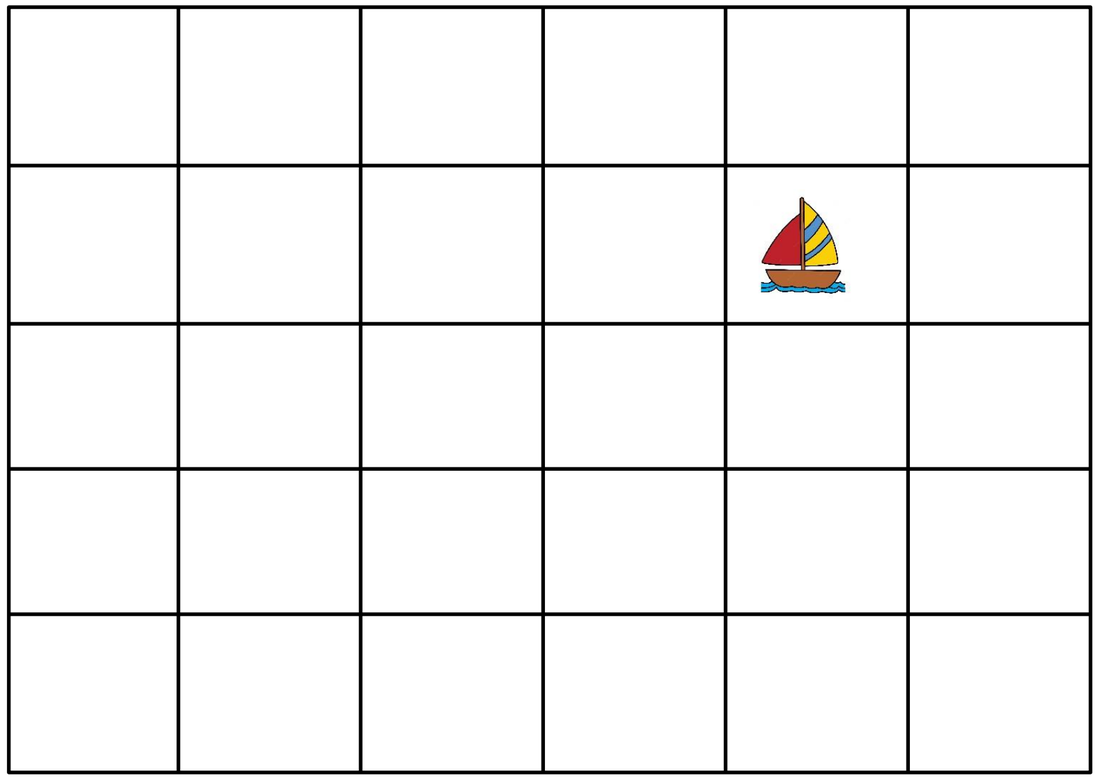
Approach 1

$$\left(x^{'}, y^{'}\right) = \begin{cases} (x + \delta_x, y + \delta_y) & \text{, w.p. } \alpha_{x,y} \\ (x, y) & \text{, w.p. } 1 - \alpha_{x,y} \end{cases}$$

In the above formula, δ_x and δ_y are directions associated with the action. More specifically, we are treating action a as a vector which is equal to (δ_x, δ_y). For e.x. the "up" action is $(\delta_x, \delta_y) = (0, 1)$, the "right" action is $(\delta_x, \delta_y) = (1, 0)$, "do nothing" action is $(\delta_x, \delta_y) = (0, 0)$. $\alpha_{x,y}$ is the probability of going to the grid as suggested by the action.

Example of Markov Decision Process

Fishing in Gridworld



State transition probability distribution

- The state transition probability distribution for this example can be written using two different approaches both of which are **equivalent**.

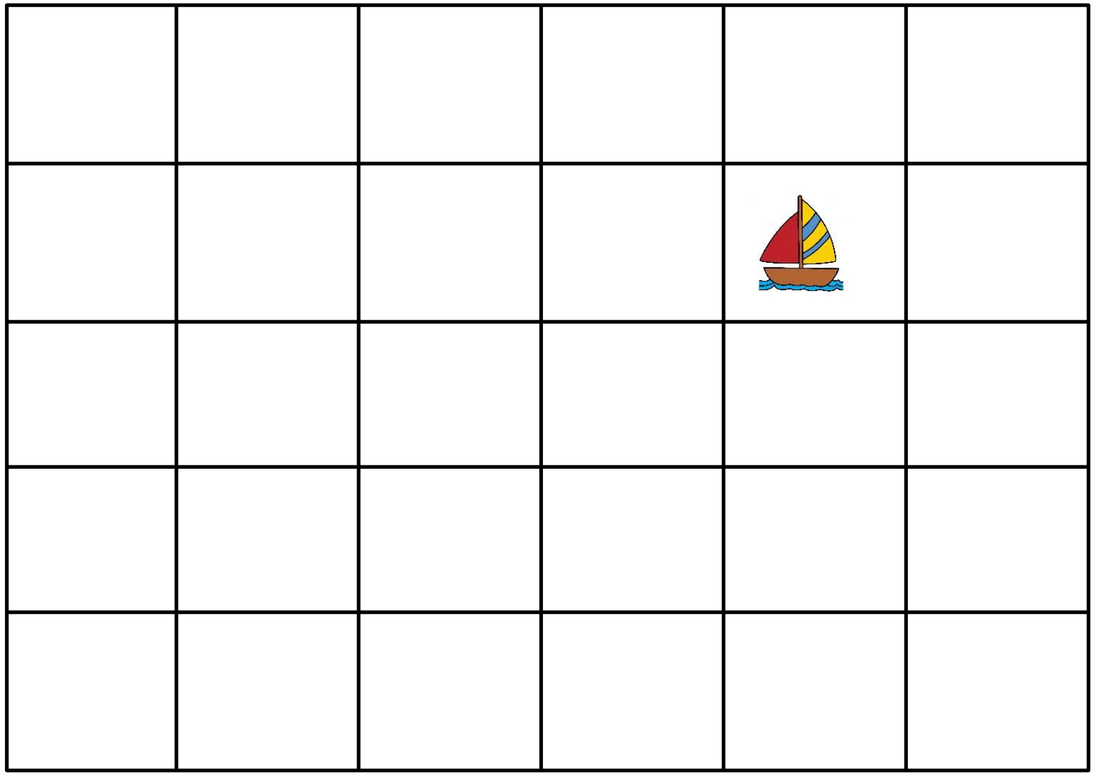
Approach 2

$$P\left[\left(X^{'},Y^{'}\right)|\left(x,y\right),\left(\delta_x,\delta_y\right)\right]=\begin{cases}\alpha_{x,y} & ,\left(X^{'},Y^{'}\right)=\left(x+\delta_x,y+\delta_y\right)\\ 1-\alpha_{x,y} & ,\left(X^{'},Y^{'}\right)=\left(x,y\right)\\ 0 & ,\text{otherwise}\end{cases}$$

While approach 2 looks more like a probability distribution, approach 1 looks like state transition. For most part, approach 1 is favorable.

Example of Markov Decision Process

Fishing in Gridworld



State transition probability distribution

- The state transition probability distribution for this example can be written using two different approaches both of which are **equivalent**.

Approach 2

$$P\left[\left(X^{'},Y^{'}\right)|\left(x,y\right),\left(\delta_x,\delta_y\right)\right]=\begin{cases}\alpha_{x,y} & ,\left(X^{'},Y^{'}\right)=\left(x+\delta_x,y+\delta_y\right)\\ 1-\alpha_{x,y} & ,\left(X^{'},Y^{'}\right)=\left(x,y\right)\\ 0 & ,\text{otherwise}\end{cases}$$

While approach 2 looks more like a probability distribution, approach 1 looks like state transition. For most part, approach 1 is favorable.

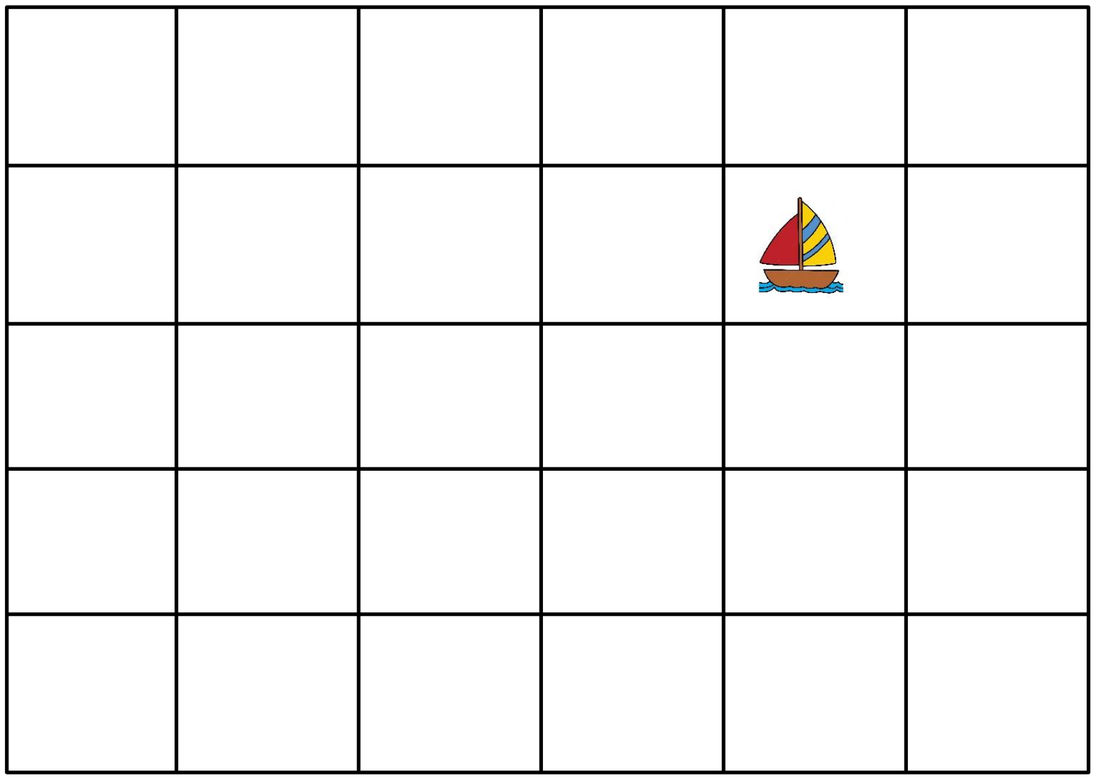
I did not account for the edge cases (corner of the grid, edge of the grid) for any of the two approaches. The state transition will get "clipped" for these edge cases. It is your homework to account for these edge cases.

Lecture Content

- Setup of Markov Decision Process.
- Example of Markov Decision Process.
- Episodic vs Continuing Tasks.

Episodic vs Continuing Tasks

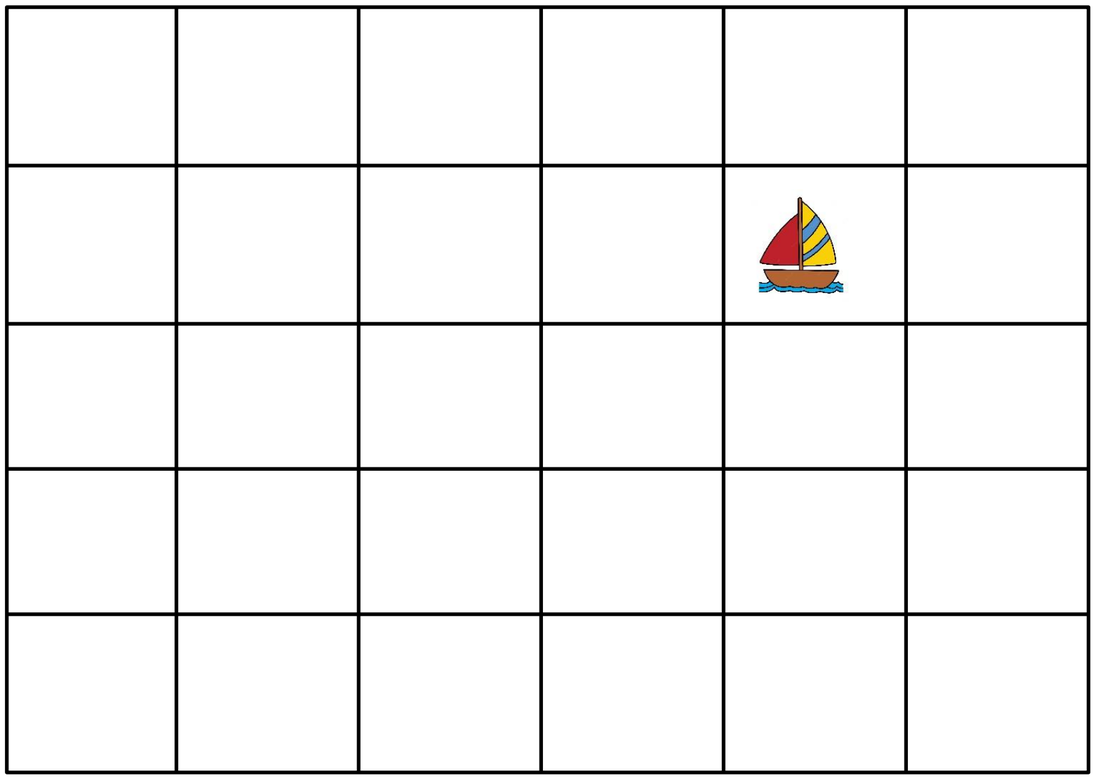
Fishing in Gridworld



- In order to understand, episodic and continuing tasks, we have to first understand the concept of an episode.
- An episode is simply a **sequence of consecutive time slots that represents a task** in action.
- **Continuing tasks:** Episode length is infinite (task is never ending). Examples:
  - Operation of a process industry.
  - Fishing in Gridworld where the fisherman keeps on catching fish.

Episodic vs Continuing Tasks

Fishing in Gridworld

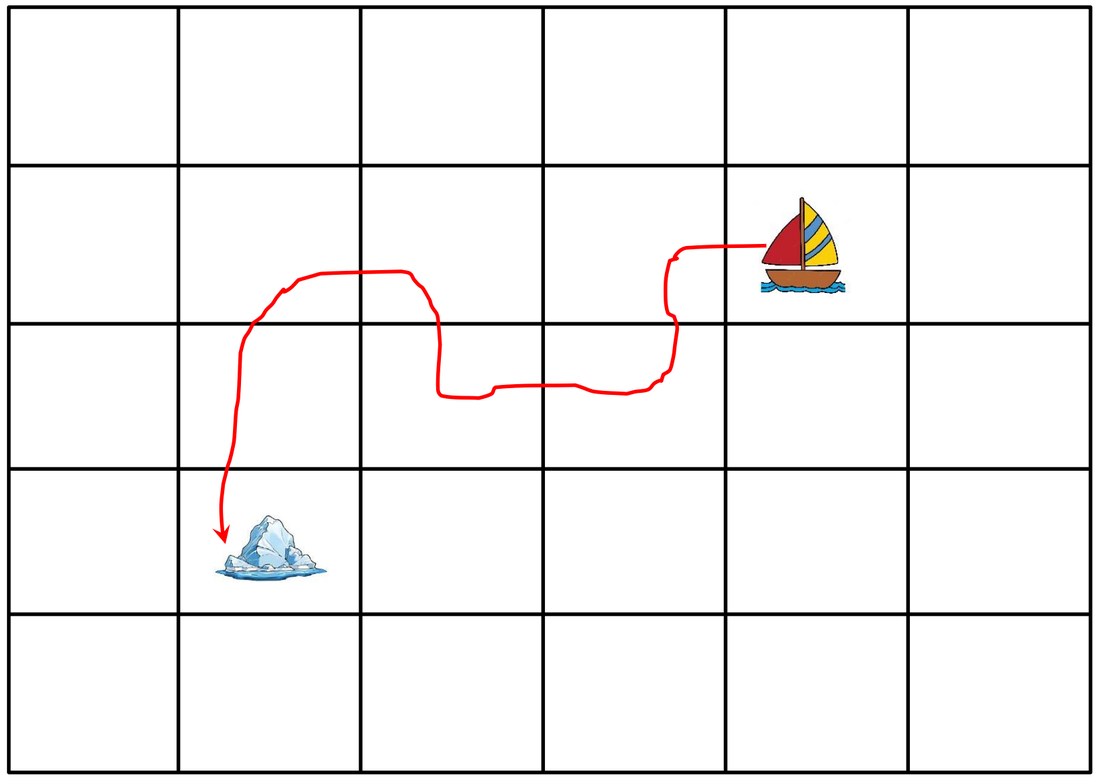


- In order to understand, episodic and continuing tasks, we have to first understand the concept of an episode.
- An episode is simply a **sequence of consecutive time slots that represents a task** in action.
- **Continuing tasks:** Episode length is infinite (task is never ending). Examples:
  - Operation of a process industry.
  - Fishing in Gridworld where the fisherman keeps on catching fish.

In practice, we have to **truncate** after a certain number of time slots. This is ok because due the discount factor β, the effective reward $\beta^t r_t$ will be very small for a large t because β^t will be very small.

Episodic vs Continuing Tasks

Fishing in Gridworld



- Throughout this course we will develop algorithms/policies for continuing task only.
- An episodic task can be converted into an **equivalent** continuing task by introducing a dummy state called the **"end state"**.
- Say that for an episodic task, the episode got over at time T where T can be random, e.x. the boat hits the iceberg at time T. Then from time $T + 1$, the state of the system becomes equal to the end state.
  - **End state always loops back to itself with probability 1**. Therefore, the state of the system will be always equal to the end state after time T.
  - The reward at end state is **zero**.


Thank you