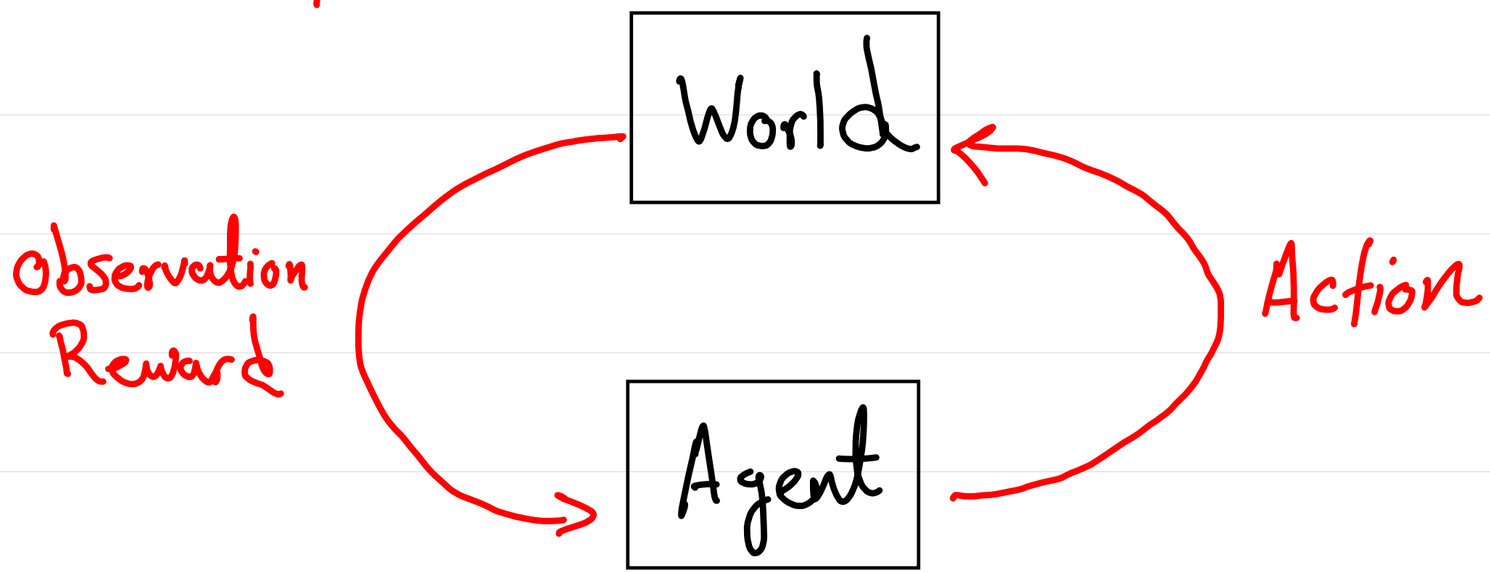
Week 11 - Part 2

- Outcome of this part
  - Sequential Decision process
  - Observation, history, and state
  - Markov Decision Process (MDP)
    - What is Markov about MDP? Why Markov assumption is common?
  - Dynamics Model & Reward Model
    - Transition Graph
  - Return / Utility
  - Policies
  - Finding the "best" policy (i.e., solving the MDP)

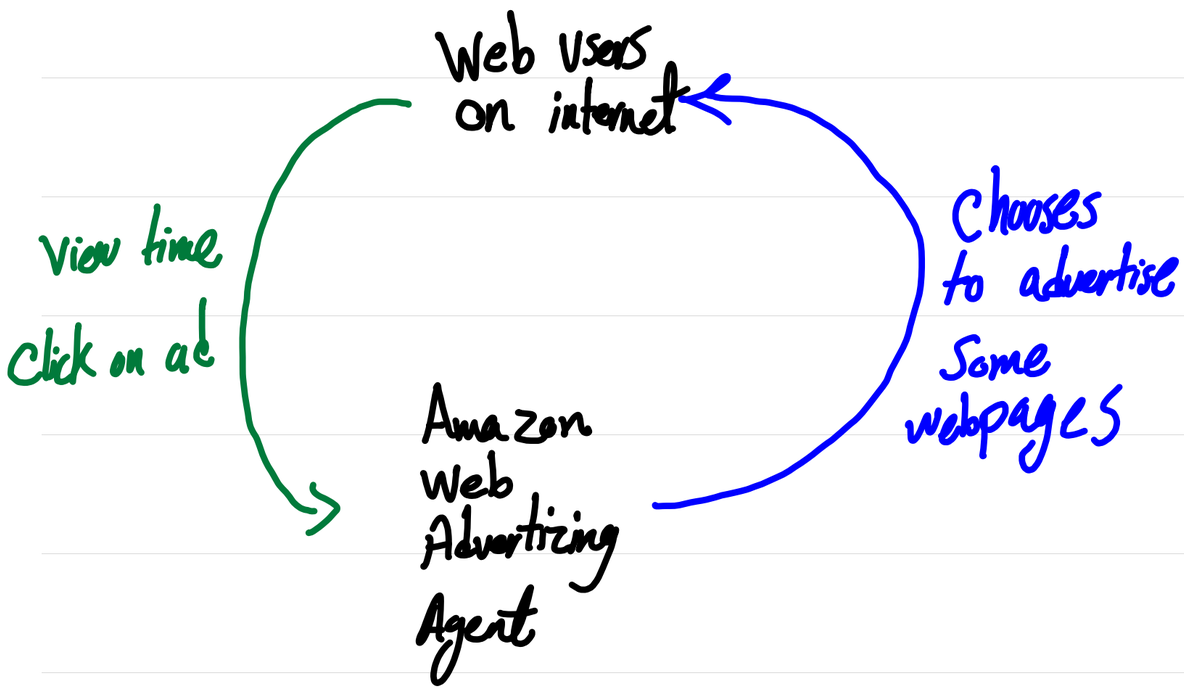
← Next part

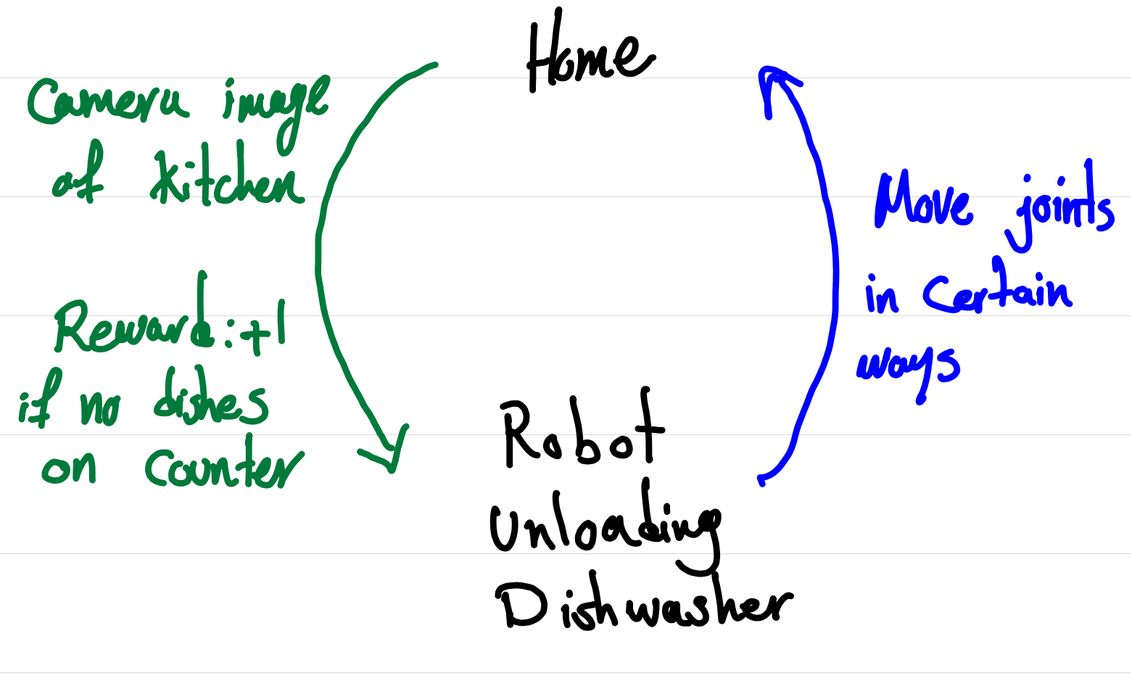
Sequential Decision Making



- **Goal:** Select actions to maximize total expected future reward
  - ↳ May require balancing immediate & long-term rewards

Some Examples

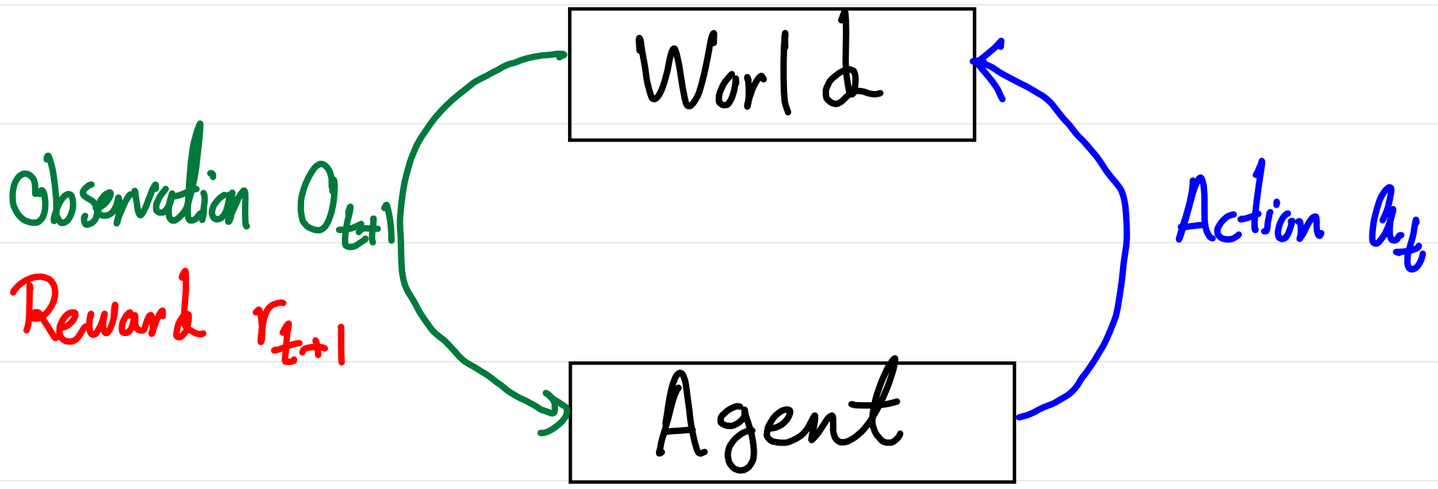




The robot can break all dishes by throwing them on the floor.

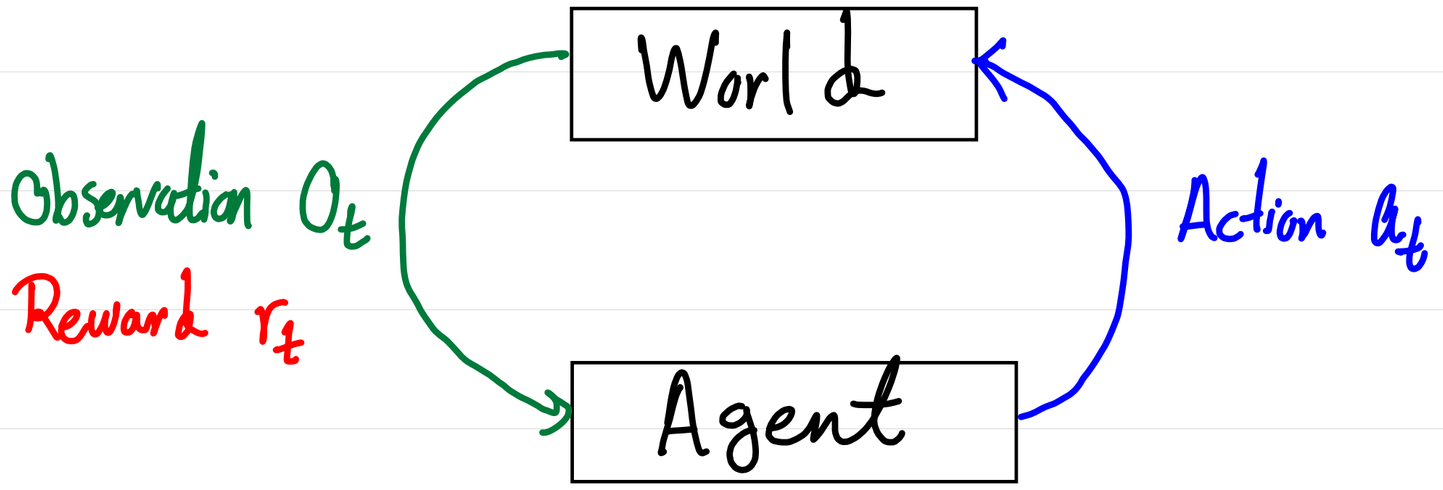
This issue is known as Rewark Hacking

Sequential Decision Process: Agent & the World (Discrete Time)



- Each time step t:
  - Agent take an action a_t
  - Given action a_t, the world updates and emits observation O_{t+1} and reward r_{t+1}
  - the agent receives O_{t+1} and r_{t+1}

History: Sequence of Past Observations, Actions, & Rewards



- History: $h_t = (a_0, O_1, r_1, a_1, O_2, r_2, \ldots, a_{t-1}, O_t, r_t)$
- Agent chooses action based on history
- State: information assumed to determine what happens next
  - State is a Function of history: $S_t = \psi(h_t)$
    - the simplest State function is $S_t = h_t$

$$S_t = O_t$$

Observation V.S. History V.S State

- Can anyone tell me what is the difference between observation, history, and state?

43

- Consider the Atari game. How would you design/choose obeservation, history, and state?
  - observation is the last image snapshot of the game
  - state would be defined as the last few observation
    - This allows us to find the speed & direction of the ball

Markov Assumption

- Often, to make problems tracktable, and because it is not a terrible assumption in reality, we will make Markov Assumption.

$$P(S_{t+1} \mid S_t, a_t) = P(S_{t+1} \mid S_t, a_t, S_{t-1}, a_{t-1}, S_{t-2}, a_{t-2}, \ldots, S_1, a_1)$$

- Future (S_{t+1}) is independent of past ($S_{t-1}, a_{t-1}, \ldots, S_1, a_1$) given the present ($S_t, a_t$)
- with markov assumption

$$P(S_{t+1}, S_t, \ldots, S_0 \mid a_t, \ldots, a_1) = P(S_{t+1} \mid S_t, S_{t-1}, \ldots, S_1, a_t, \ldots, a_1) \times \rightarrow P(S_{t+1} \mid S_t, a_t)$$
$$P(S_t \mid S_{t-1}, S_{t-2}, \ldots, a_t, \ldots a_1) \times \rightarrow P(S_t \mid S_{t-1}, a_{t-1})$$
$$P(S_0)$$

Why is Markov Assumption Popular?

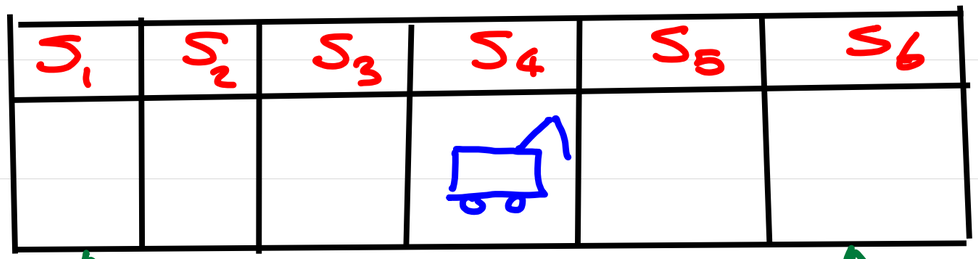
- It is simple and often can be satisfied if includes some previous observations as part of state
- There are many problems that we can satisfy Markov assumption by simply setting $S_t = O_t$.

- Markov assumption significantly simplifies working with the problem.

- Just like all ML problems we studied so far, there is this trade-off between the expressive power of state representation and how long it takes to train your model.

MDP Example: Mars Rover

locations on Mars →



s_1 and s_6 are the terminal locations, you obtain 1 and 5 samples from Soil in locations s_1 and s_6, respectively, and then the robot will shut down

- Mars Rover is the robot that you designed to collect samples on Mars.
- Your robot has to choose from going to right or left, to move between locations.
- Mars is strangely small. Only 6 locations on Mars.

State set: $S = \{s_1, \ldots, s_6\}$

Action Set: $A = \{\text{"Try Left"}, \text{"Try Right"}, \text{"exit"}\}$

Rewards: reward 1 and 5 when you get to s_1 and s_6, respectively.
reward 0, otherwise.

Dynamics Model & Reward Model

- Transition / dynamics model: predicts next state
- In other words, dynamics model specifies the distribution of outcomes when the agent makes a decision

→ $T(s, a, s') = \mathbb{P}[S_{t+1} = s' \mid S_t = s, a_t = a]$

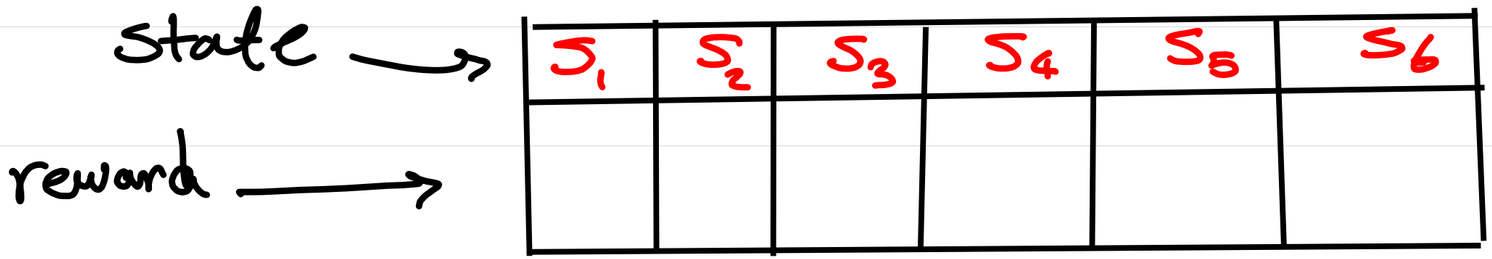
In this course, we assume stationary transition / dynamics model.

i.e., $\mathbb{P}[S_{t+1} = s' \mid S_t = s, a_t = a] = \mathbb{P}[S_{k+1} = s' \mid S_k = s, a_k = a]$

for all t and k.

- Reward model: predicts immediate reward: $R(s, a, s')$
  - Sometimes just $R(s)$, $R(s, a)$, or $R(s')$, depending on the model
  - Note that rewards can be zero, negative, or positive.

Example: Mars Rover Dynamics Model (Reward Model)



States & Actions: $S = \{s_1, \ldots, s_6\}$, $A = \{a_L, a_R, a_e\}$

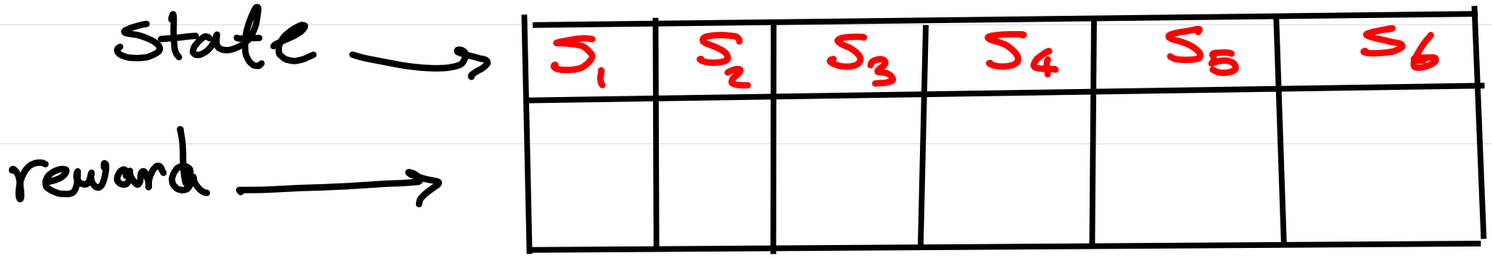
Rewards Model:

$$R(s,a,s') = \begin{cases} 5 & \text{if } s = s_5, a \in \{a_L, a_R\}, s' = s_6 \\ 1 & \text{if } s = s_2, a \in \{a_L, a_R\}, s' = s_1 \\ 0 & \text{otherwise} \end{cases}$$

$$R(s_6, a_e, s_6) = R(s_1, a_e, s_1) = 0$$

It's Mars we are talking about. It's different from our Earth. Our robot is not Completely predictible there. When the robot decides to go toward a direction, with 0.2 probability it may go the opposite direction.

Example: Mars Rover Dynamics Model (Dynamics Model)



It's Mars we are talking about. It's different from our Earth. Our robot is not Completely predictible there. When the robot decides to go toward a direction, with 0.2 probability it may go the opposite direction.

Transition Model:

$$T(S_i, a_L, S_{i-1}) = T(S_i, a_R, S_{i+1}) = 0.8 \quad \forall i \in \{2, \ldots, 5\}$$

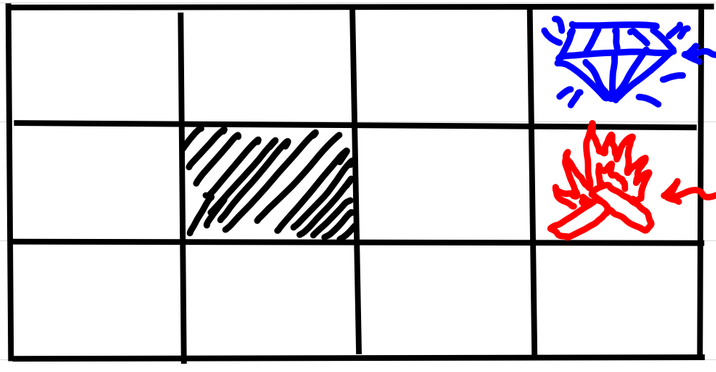
$$T(S_i, a_L, S_{i+1}) = T(S_i, a_R, S_{i-1}) = 0.2$$

$$T(S_1, a_e, S_1) = T(S_6, a_e, S_6) = 1$$

//

Example: Mars Rover In Grid World

- Consider the following Grid Model of Mars



Mars Rover gets a reward of (+1) if gets to this location

Mars Rover gets a reward of (-1) if it gets to this location

$R(s, a, s') = -0.03$ for all non-terminal states s'

- Noisy Movements on Mars: actions do not always go as planned
  - with prob. 0.8, the action takes the agent to the desired direction
  - with prob. 0.1, the actions takes the agent to a direction prependicular to the desired direction.
  - If there's a wall in the direction would have been taken to, it stays put

Homework: Mars Rover In Grid World

- formulate this MDP.

$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{1,0}$		$S_{1,2}$	$2_{1,3}$
$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$

Example: The undergrad student's life at UofT

- Erfan wants to thrive in his undergrad
- He has three states: happy, tired, burnt-out
  - burnt-out is the terminal state.
- He has two actions: "Work hard", "party hard"
  - Working hard gets double reward: reward 2 for working hard

S	a	S'	T(S, a, S')	R(S, a)
	Work hard		0.5	2
	Work hard		0.5	2
	party hard		1	1
	Work hard		1	2
	party hard		0.5	1
	party hard		0.5	1
	end		1	0

We can summarize the dynamics and reward model with a table

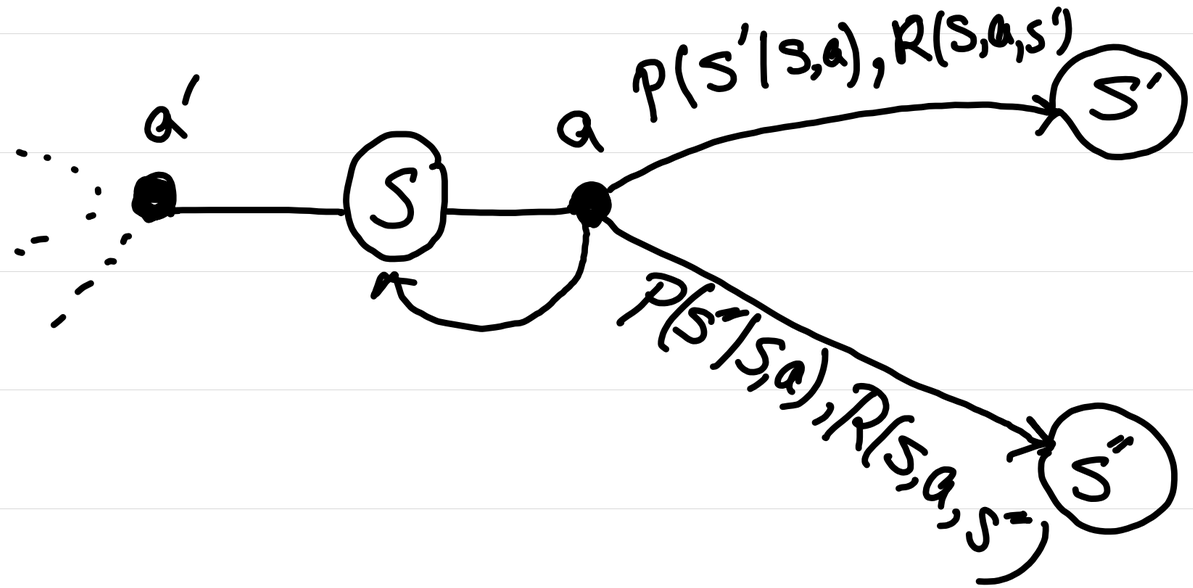
Transition Graph

- Transition Graph is a useful way of summarizing the dynamics of a finite MDP
- There are two types of nodes
  - state nodes: There is a state node for each possible state
    - Typically a large circle labeled by the name of the state, e.g., (s)
  - Action nodes (also called q-state node): There is an action node for each state-action pair.
    - Typically a small solid circle labeled by the name of action or (state, action) pair and connected by a line to the state node, e.g., $\bullet^{a}$ or $\bullet^{(s,a)}$

Transition Graph

- Starting in state "s" and taking action "a" move you along the line from state nods "S" to action node (s, a).

- Then the environment responds with a transition to the next state's node.



Activity: Draw the transition Graph for Erfan the Undergrad