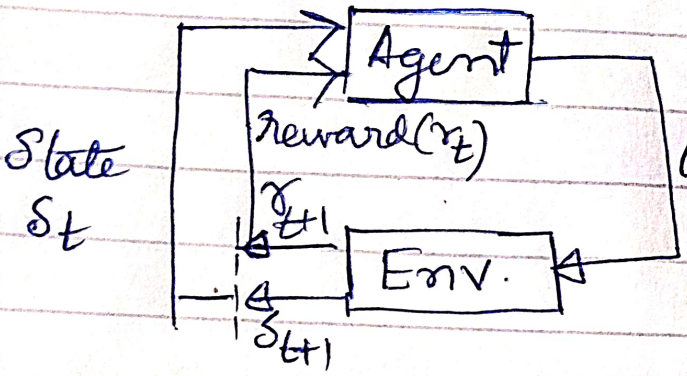
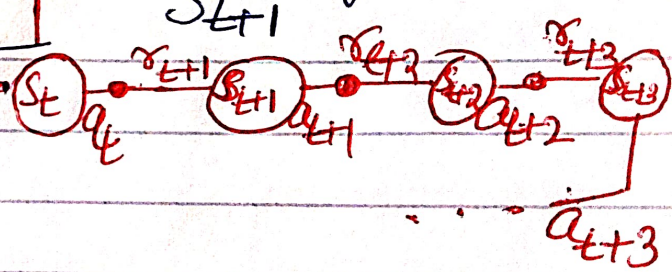
"MDP" - Markov Decision Process

The agent-env. Interface



- Agent & environment interact at discrete time steps: $t = 0, 1, 2 \ldots$
- Agent observes state at step t: $s_t \in S$
- Produces action at step t: $a_t \in A(s_t)$
- Gets resulting reward: $r_{t+1} \in R$
- & resulting next state: s_{t+1}



- close loop
- Few assumptions: • discrete time steps. $t = 0, 1, 2$ (decision steps, not interaction)
  - decision steps ≡ opportunity ≡ not exactly how long
- 1 decision ~ one time step.

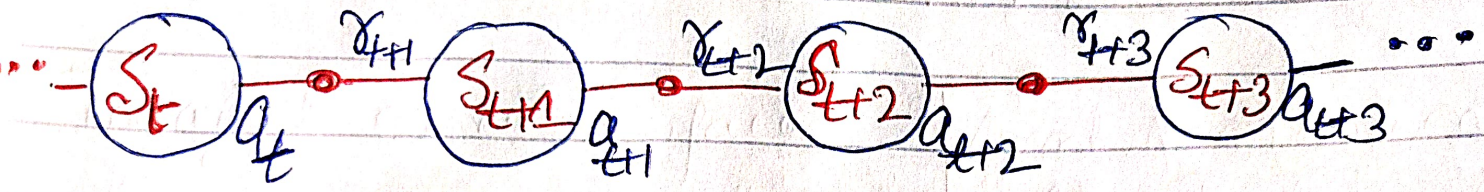
"S" - set of ~~step~~ states (discrete) ≈ 1 to N. [$S_0 = 3$, → not exactly always zeroth step. $S_1 = 7$.]

- A_t is applied to [Environment], causes the environment state to change to S_{t+1}; also produces the reward r_{t+1} [which is a result of taking at in state S_t, & moving to S_{t+1}]

∴ It can be joint distribution

$p(\text{next state}, r \mid s, a) \rightarrow$ $\boxed{p(s', r \mid s, a)}$

∴ What is happening bet^{n} env' & agent'.



At, state S_t take action a_t, get reward r_{t+1} & move to state S_{t+1}. Then again take action a_{t+1} & get reward r_{t+2} & move to S_{t+1} ... So on. ...

This called as "trajectory" to the set of states.

Set of states comes from "S" & set of discrete actions come from set of discrete actions A.

∴ Remember 'S' is set of state space & discrete

Set A is a ~~set~~ no of discrete actions;

The reward is a r_{t+1} which is a real valued scalar quantity.

The first assumption we made is "Markov Property" to make things easier here.

"What is Markov Property"?

① Here "the state" at step t, means whatever information is available to the agent at step t about its environment.

② The state can include immediate "sensations" highly processed sensations & structures built up over time from sequences of sensations.

③ Ideally, a state should summarize past sensations so as to retain all "essential" information, i.e it should have the "Markov Property":

$$\Pr\{S_{t+1}=s', r_{t+1}=r \mid s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\}$$

$$= \Pr\{S_{t+1}=s', r_{t+1}=r \mid s_t, a_t\}$$

$\forall\, s', r$, and histories $s_1, a_1, r_1, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0$

② State could be Agent's position (robot's coordinate), velocity & direction: current speed & movement direction; sensor readings: distance to obstacles, object proximity; Energy level: battery percentage for a drone or robots etc.

"Essential" information

$$Pr\{S_{t+1}=s', r_{t+1}=r \mid S_1, a_t, r_1, S_{t-1}, a_{t-1}, \ldots r_1, S_0, a_0\}$$

So, the assumption is that the state should have "all essential informations to predict the future (behaviors)

$$Pr\{S_{t+1}=s', r_{t+1}=r \mid S_1, a_t, r_1, S_{t+1}, a_{t+1} \ldots$$

If current state S_t & action a_t, the prev. components are not needed.

* Not dependent on history, current state & actions are enough to predict the system behaviour. — Markov Property.

ex: $S_t = A$, $S_t = f$ [no need of knowing history]

$P(s', r \mid s, a)$ → Markov property.

Markov Decision Process

- MDP, M is the tuple : $M = \langle S, A, p, r \rangle$
  - S: set of states (finite discrete)
  - A: set of actions (finite discrete)
  - $p: S \times A \times S \rightarrow [0,1]$: probability of transition (1)
  - $r: S \times A \times S \rightarrow \mathcal{R}$: Expected reward (2)
    [$Pr(r|s,a,s')$; instead saying we going to develop model this distribution, we are finding expected value]
- Policy : $\pi : S \times A \rightarrow [0,1]$ (can be deterministic) (3)
  [is a mapping from state to action]
- Maximize total expected reward.
- Learn an optimal policy.

(1) $p \simeq Pr(s'|s_1, a_1)$ → lets say we are at current state s_1, we do action a_1; what is the probability that we will have another state s'. So for every possible s', we will be having a value. For for every combination of s_1 & a_1 we will have a value (probability). For no of states you have no of values.

(2) $r(s,a,s') = \int r \cdot Pr(r|s,a,s')\,dr$

Some time joint distribution also
$p(s', r | s, a)$

Policy π: The role of the agent is to learn the policy π, which is a mapping from state to action.

$\pi : S \times A \rightarrow [0,1]$

$\pi(a|s)$ or $\pi(s,a) = \Pr(a_t = a | s_t = s)$

For every state you could have 'n' actions; some time transition is probabilistic/stocastic; the policy π can be deterministic

$\pi(a|s) = 1$ for $a = a_4$
$= 0$ otherwise

∴ Some times,
$\pi(s) = a_4$ (deterministic).

The goal of the agent is to maximize the total expected reward: is called optimal policy.