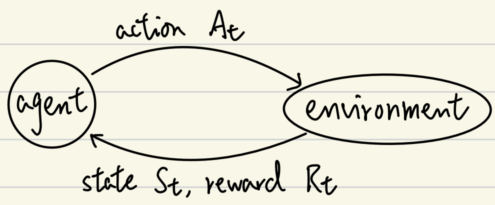
3.1 The Agent-Environment Interface.

MDPs are a classical formalization of sequential decision making. Actions influence not only immediate rewards but also subsequent states and future rewards.



trajectory: $S_0, A_0, R_0, S_1, A_1, R_1, S_2, A_2, R_3, \cdots$

Def

- $\mathcal{S}, \mathcal{A}, \mathcal{R}$ are sets of states, actions, rewards.
  If $\mathcal{S}, \mathcal{A}, \mathcal{R}$ are finite, we say that a MDP is finite, i.e. FMDP.

- We define the dynamics of a MDP

$$p(s', r \mid s, a) := \mathbb{P}(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a) \text{ for all } s', s \in \mathcal{S}, r \in \mathcal{R}, a \in \mathcal{A}(s)$$

i.e. the prob. of each possible value for S_t and R_t depends on the "immediately" proceding state and action s and a.

- $\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
- p is a function that $p: \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \longrightarrow [0,1]$.
- Note that
  1. p is a "deterministic" function.
  2. p is defined with time steps from $t-1$ to t.
  3. Markov property here means that the state must include information about all aspects of the past interaction.

If we have p, we can compute:

- state-transition probabilities $p: \mathcal{S} \times \mathcal{S} \times \mathcal{A} \longrightarrow [0,1]$

$$p(s' \mid s, a) := \mathbb{P}(S_t = s' \mid S_{t-1} = s, A_{t-1} = a) = \sum_{r \in \mathcal{R}} \underbrace{p(s', r \mid s, a)}_{p:\ \text{dynamics}}$$

- expected rewards for state-action pairs $r: \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$

$$r(s, a) := \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r \mid s, a)$$

· expected rewards for state-action-next state triples $r: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$

$$r(s,a,s') := \mathbb{E}[R_t \mid S_{t-1}=s, A_{t-1}=a, S_t=s'] = \sum_{r \in \mathcal{R}} r \frac{p(s',r \mid s,a)}{p(s' \mid s,a)}$$

Remark. The framework is flexible and can be applied to different problems.

- state
- action
- reward

Example : Recycling robot

$\mathcal{S} = \{\text{high}, \text{low}\}$

$\mathcal{A}(\text{high}) = \{\text{search}, \text{wait}\}$

$\mathcal{A}(\text{low}) = \{\text{search}, \text{wait}, \text{recharge}\}$

$\mathcal{R} = \{r_{\text{search}}, r_{\text{wait}}, 0, 1, -3\}$

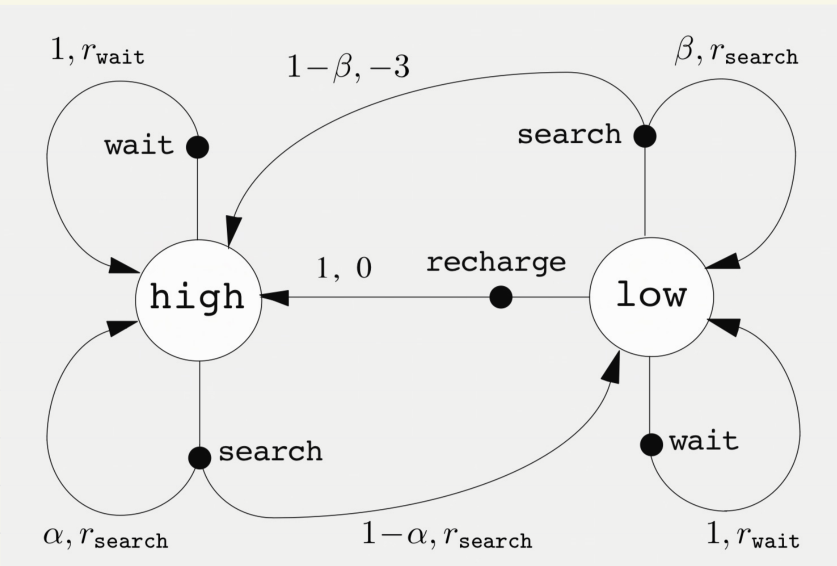
$r_{\text{search}}, r_{\text{wait}}$: expected number of cans that the robot will collect.

1: collect one can

-3: whenever the robot has to be rescued

0: collect 0 can (including recharging)

(probability, reward)



α: the prob. that it begins with high energy level and ends with high energy level.
β: the prob. that it begins with low energy level and ends with low energy level.

3-2 Goals and Rewards

Informally, the agents goal is to maximize the total amounts of rewards it received.

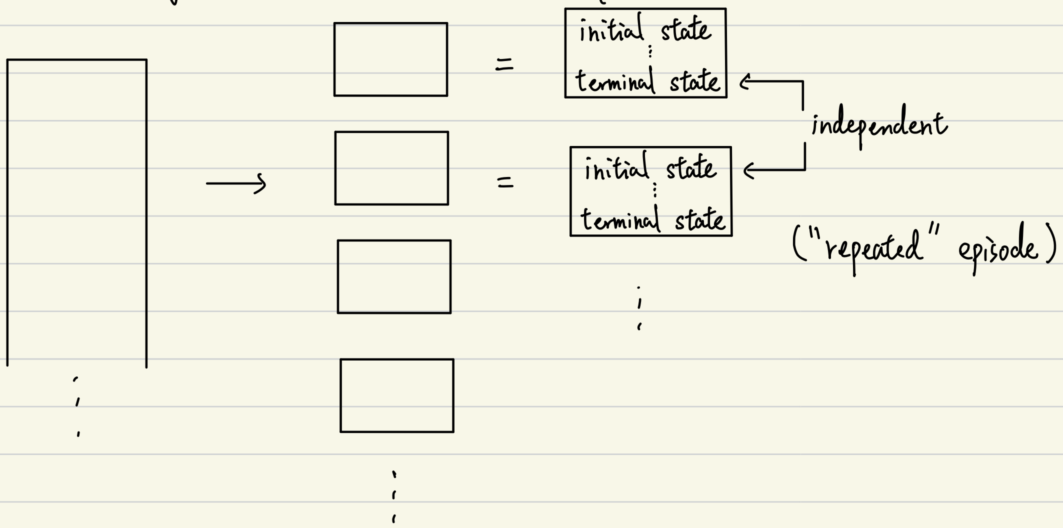
3-3 Returns and Episodes (3-4 is included here)

$R_{t+1}, R_{t+2}, \cdots$: sequence of rewards after time t.

To be precise, we want to maximize the expected reward after time t.

In practice, we cannot deal with the case that $t=\infty$. Thus, it's natural to consider the concept of "final time".

$\Rightarrow$ We break the agent-environment interaction sequence into subsequences $\rightarrow$ episode task



分成很多回合

$G_t := R_{t+1} + R_{t+2} + \cdots + R_T$, T: final time step , 每個 episode 有不同的 T.

$T = \infty \rightarrow$ Continuing task

Remark.

- Start from time $t=0$
- $S_{t,i}$: the state at time t in the i^{th} episode
- We almost never have to distinguish between different episodes. $\rightarrow$ discuss a particular episode
- terminal state $\rightarrow$ set $t=0$

Discounting :

For $0 \le \gamma \le 1$,

$$G_t := R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Future reward discount to the present value
- near to the presence → greater weight "近的比較重要"
  far from the presence → smaller weight

· 避免加到 ∞

$$
\begin{aligned}
G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \\
&= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \cdots) \\
&= R_{t+1} + \gamma G_{t+1} \qquad \text{for } t < T.
\end{aligned}
$$

3-5. Policies and Value Functions

value function → Given a state, how good it is for the agent (the purpose of value functions)
↓ expected return (depends on what actions the agent will take)
the estimation is included in reinforcement learning algorithms

Def A policy is a mapping from states to probabilities of selecting each possible action.

$$\text{policy} : \text{states} \longmapsto \text{probabilities of selecting each possible action}$$

$\pi(a|s) := \mathbb{P}(A_t = a | S_t = s)$, π: policy

At time t, if the agent follows the policy π, it will take the action a given the state s with probability $\pi(a|s)$.

Remark RL methods specify how the agent's policy is changed because of its experience.

Def The value function of state s under the policy π, $V_\pi(s)$, is defined as

$$V_\pi(s) := \mathbb{E}_\pi [G_t | S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s \right] \quad \forall s \in S$$

↳ take the policy π

Def The action-value function, $q_\pi(a,s)$, is defined as

$$q_\pi(s,a) := \mathbb{E}_\pi [G_t | S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s, A_t = a \right]$$

Remark

Exercise 3.12 Give an equation for v_π in terms of q_π and π.

Sol. $V_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$

Exercise 3.13 Give an equation for q_π in terms of v_π and the four-argument p.

Sol. four-argument p: $p(s',r|s,a) = \mathbb{P}(S_{t+1}=s', R_{t+1}=r \mid S_t=s, A_t=a)$

Note that

$$G_t = R_{t+1} + \gamma G_{t+1}.$$

Then it's natural to use the linearity of expectation:

$$\begin{aligned}\mathbb{E}_\pi[G_t|S_t=s, A_t=a] &= \mathbb{E}_\pi[R_{t+1}|S_t=s, A_t=a] + \mathbb{E}_\pi[\gamma G_{t+1}|S_t=s, A_t=a] \\ &= \underbrace{\mathbb{E}_\pi[R_{t+1}|S_t=s, A_t=a]}_{①} + \gamma \underbrace{\mathbb{E}_\pi[G_{t+1}|S_t=s, A_t=a]}_{②}.\end{aligned}$$

$$① = \sum_{r\in\mathcal{R}} \Big(\sum_{s'\in\mathcal{S}} p(s',r|s,a)\Big)\cdot r$$

$$\begin{aligned}② &= \gamma\, \mathbb{E}_\pi\big[\mathbb{E}_\pi[G_{t+1}|S_{t+1}=s'] \,\big|\, S_t=s, A_t=a\big] \\ &= \gamma\, \mathbb{E}_\pi[\underbrace{v_\pi(s')}_{\text{at time } t+1} \mid S_t=s, A_t=a] \\ &= \gamma \sum_{s'\in\mathcal{S}}\Big(\sum_{r\in\mathcal{R}} p(s',r|s,a)\Big) v_\pi(s')\end{aligned}$$

It follows that

$$q_\pi(s,a) = \sum_{s'\in\mathcal{S}}\sum_{r\in\mathcal{R}} p(s',r|s,a)\,[r+\gamma v_\pi(s')]. \quad \square$$

☆☆☆

<u>Theorem</u> (Bellman equation).

$$\begin{aligned}v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t=s] \\ &= \mathbb{E}_\pi[R_{t+1}+\gamma G_{t+1} \mid S_t=s] && \text{(by (3.9))} \\ &= \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)\Big[r+\gamma\mathbb{E}_\pi[G_{t+1}|S_{t+1}=s']\Big] && \text{(by Exercise 3.13)} \\ &= \sum_a \pi(a|s)\sum_{s',r} p(s',r|s,a)\Big[r+\gamma v_\pi(s')\Big], \quad \text{for all } s\in\mathcal{S}, && (3.14)\end{aligned}$$

3-6 Optimal Policies and Optimal Value Functions

<u>Partial Ordering</u> Given two policies π, π'. We say that $\pi \geq \pi'$ if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s\in\mathcal{S}$.

<u>Def</u>

- The optimal value function is defined as

$$v_*(s) = \max_\pi v_\pi(s) \qquad \forall s\in\mathcal{S}.$$

- The optimal action-value function is defined as

$$q_*(s,a) = \max_\pi q_\pi(s,a) \qquad \forall s\in\mathcal{S},\ a\in\mathcal{A}(s).$$

Remark

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}[G_t \mid S_t = s, A_t = a] \\
&= \mathbb{E}[G_t + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \mathbb{E}[G_t + \gamma \mathbb{E}[G_{t+1} \mid S_{t+1}] \mid S_t = s, A_t = a] \\
&= \mathbb{E}[G_t + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]
\end{aligned}
$$

Bellman optimality equation for v_*:

$$
\begin{aligned}
v_*(s) &= \max_{\pi} v_{\pi}(s) \\
&= \max_{\pi} \sum_a \pi(a|s)\, q_{\pi}(s,a) \\
&= \sum_a \pi_*(a|s)\, q_*(s,a) \\
&= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_{a \in \mathcal{A}(s)} \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s',r|s,a)\,(r + \gamma v_*(s'))
\end{aligned}
$$

Denote the probability of taking action a followed by the optimal policy $\pi_*(a|s)$.
By definition,

$$
\pi_*(a|s) = \begin{cases} 1, & a = \underset{a \in \mathcal{A}(s)}{\operatorname{argmax}}\, q_*(s,a) \\ 0, & \text{else} \end{cases}
$$

Bellman optimality equation for q_*:

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a] \\
&= \sum_{s',r} p(s',r|s,a)\,[r + \gamma \max_{a'} q_*(S_{t+1}, a')]
\end{aligned}
$$

Remark

1. The solution of Bellman optimality equation is rarely useful.
   - looking ahead of all possibilities
   - computing the probabilities of occurrences

   $\longrightarrow$ Difficult

2. Assumptions of Bellman optimality equation (rarely true!!)
   a. the dynamics of the environment is accurately known
   b. computational resources are enough
   c. the states have the Markov property not always have