Terminology

Agent 在 t 时刻

1. 所处的状态 s_t ---------- **2.** 进行的动作 a_t


agent 是如何根据状态来**决定**动作的？换言之，a_t与s_t如何联系起来？

3. Policy $\pi(A|S)$: agent 在处于状态s_t^3的情况下，以特定的目标为指导，做出动作a_t的概率。例如

$\pi(a_t^1|s_t^3)$=0.3

$\pi(a_t^2|s_t^3)$=0.65

$\pi(a_t^3|s_t^3)$=0.05

即，在状态s_t^3的情况下，agent做出action 2的概率最大，是0.65。

接下来的问题是，特定目标是什么？是累积奖励cumulative reward的最大化。

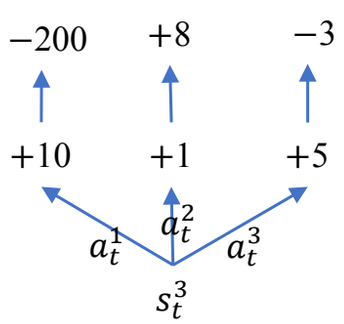
那，什么是奖励？

4. Reward 的设定，就是对agent的世界观进行设定，告诉它什么是对的、什么是错的，从而决定了它的行为准则。例如：

- Collect a coin: $R = +1$
- Win the game: $R = +10000$
- Touch a Goomba: $R = -10000$ (game over).
- Nothing happens: $R = 0$

Agent的trained policy会尽量规避风险，努力向final迈进，但也会偶尔吃金币(假如金币远小于10000)。

注意：agent在基于policy做出action是有**随机性的**, 并不是一味的去做能够获得最大累积奖励的action。例如，agent处于状态s_t^3时可能进行的动作，以及获得的奖励如图1，Policy如图2所示。



Policy:

$\pi(a_t^1|s_t^3)$=0.03

$\pi(a_t^2|s_t^3)$=0.2

$\pi(a_t^3|s_t^3)$=0.77

Potential action pool

a_t^1 a_t^1 a_t^1 3个

a_t^2 。。。 a_t^2 20个

a_t^3 。。。 a_t^3 77个

随机采样 → a_t^3

虽然agent做a_t^2的获益最大，但是agent还是有可能做a_t^3。这就好比agent在一个由policy决定的potential action pool中进行随机采样，虽然a_t^2样本最多，但是完全有可能采样到a_t^3，甚至a_t^1。

综上，在RL中，在一个状态下，agent的动作虽然倾向于最优，但是同样具有随机性，其目的是在博弈时增加不确定性，防止被对手看穿。

当agent进行了动作后，agent就会处于一个新的状态 s_{t+1}，这个状态是确定的吗？不是。如下图所示，当agent做了向上的动作，环境(程序)可能会返回不同的状态s_{t+1}。在这个例子中，环境的随机状态由怪物的随机移动导致。

5. State transition: $p(s_{t+1}^1 | s_t^3, a_t^3)$

当前状态为s_t^3，进行动作a_t^3后，环境使其处于s_{t+1}^1的概率。这是环境可知而玩家不可知的。

注意：在RL中，在agent做出一个动作后，环境使其处于的新状态具有不确定性，即环境用状态转移函数算出随机性，然后用概率随机抽样得到s_{t+1}^4

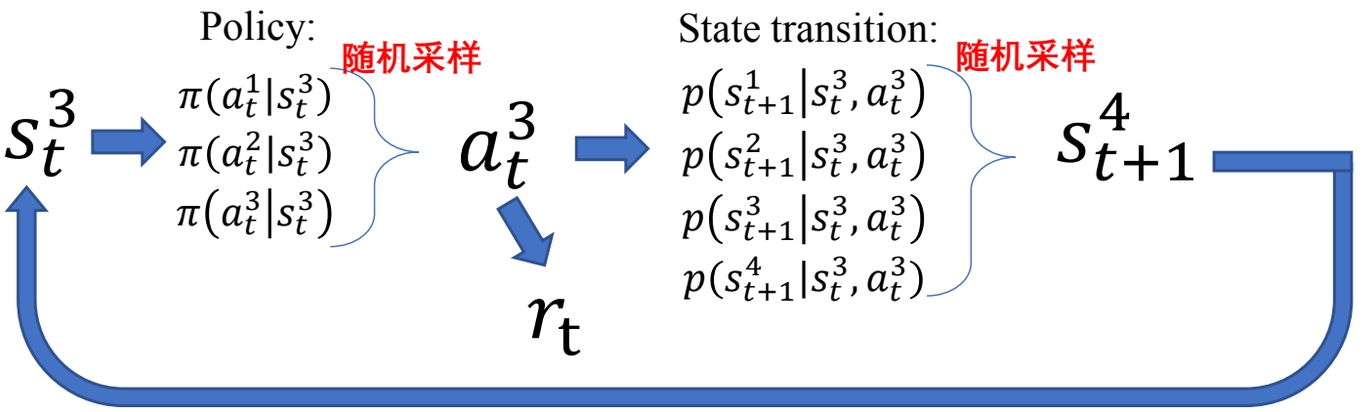
$p(s_{t+1}^1 | s_t^3, a_t^3)$=0.3
$p(s_{t+1}^2 | s_t^3, a_t^3)$=0.5
$p(s_{t+1}^3 | s_t^3, a_t^3)$=0.1
$p(s_{t+1}^4 | s_t^3, a_t^3)$=0.2

s_{t+1}^1 s_{t+1}^1 s_{t+1}^1
s_{t+1}^2 s_{t+1}^2 s_{t+1}^2 s_{t+1}^2 s_{t+1}^2
s_{t+1}^3
s_{t+1}^4 s_{t+1}^4

随机采样 → s_{t+1}^4

综上：阐述了RL中的五个基本概念和两个不确定性来源

State-action-reward trajectory:



接下来，讲述三个高级的概念。

6. Return

Definition: Return (aka cumulative future reward).

- $U_t = R_t + R_{t+1} + R_{t+2} + R_{t+3} + \cdots$
  - Future reward is less valuable than present reward
  - R_t should be given less weight than R_{t+1}

  从人的思维来讲，1年后的reward相比于今天的reward具有更大的不确定性。在衡量两者时，1年后的reward具有较小的权重。

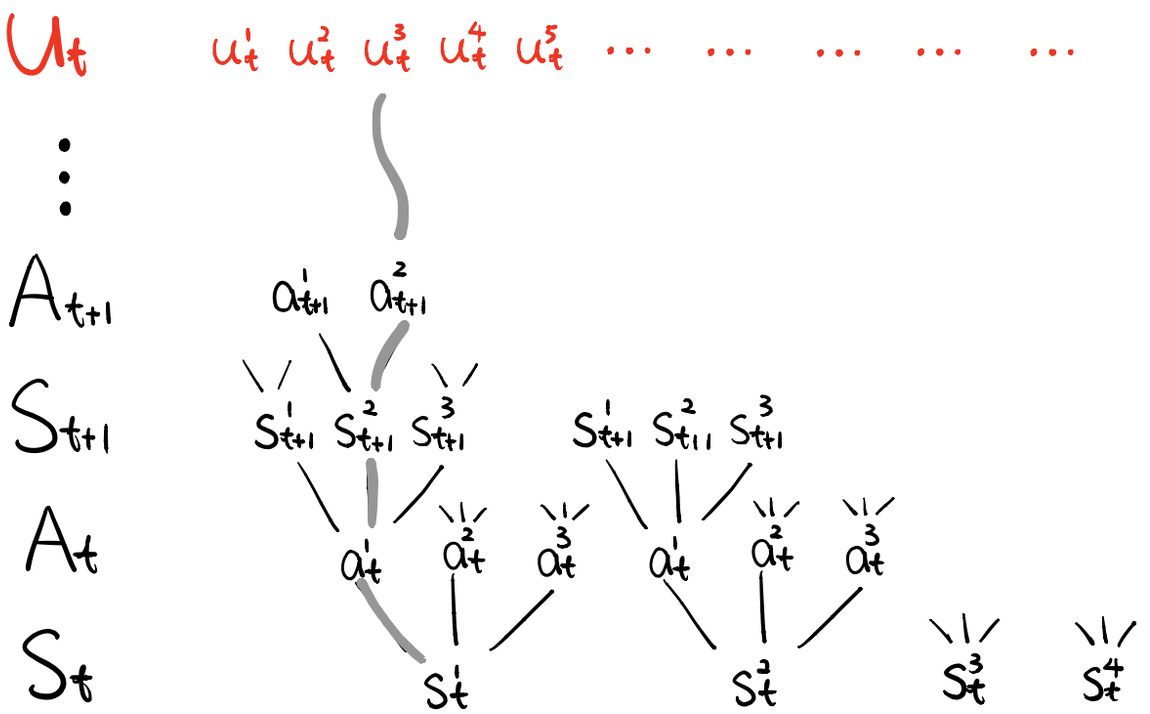
Therefore,

Definition: Discounted return (aka cumulative discounted future reward).

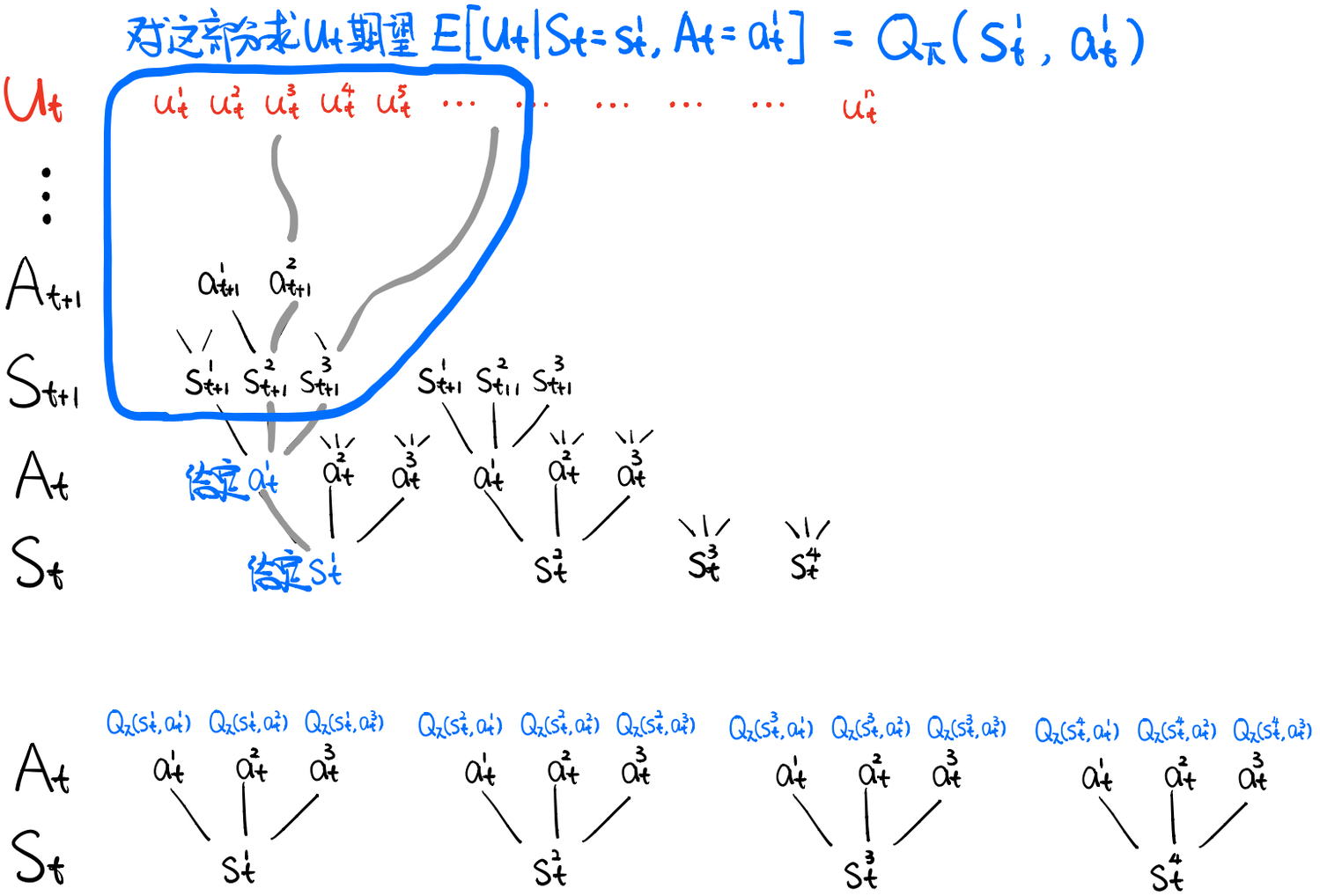
- γ: discount rate (tuning hyper-parameter).
- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \cdots$

U_t表示：从t时刻起，将来能获得的cumulative reward。由于在任意时刻的状态、相应的决策都具有**随机性**, State-action-reward trajectory具有许多种可能，而每一种可能对应于一个cumulative reward u_t。因此，U_t是一个随机变量，具有随机性(这也是为什么用大写字母的原因)。

- Return U_t depends on states $S_t, S_{t+1}, S_{t+2}, \cdots$ and actions $A_t, A_{t+1}, A_{t+2}, \cdots$



7. Action-Value Function:



Definition: Action-value function for policy π. 与policy有关

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t | S_t = s_t, A_t = a_t\right]$.

  给定policy π，在状态s_t^1条件下选择a_t^1能 { 得到的 U_t的期望。 / 有多好。

Definition: Optimal action-value function. 与policy无关

- $Q^\star(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t)$.

  遍历所有policy，在状态s_t^1条件下选择a_t^1 { 最多能得到的U_t的期望。 / 最好也就是这样了。

Whatever policy function π is used, the result of taking a_t^1 at state s_t^1 cannot be better than $Q^*(s_t^1, a_t^1)$.

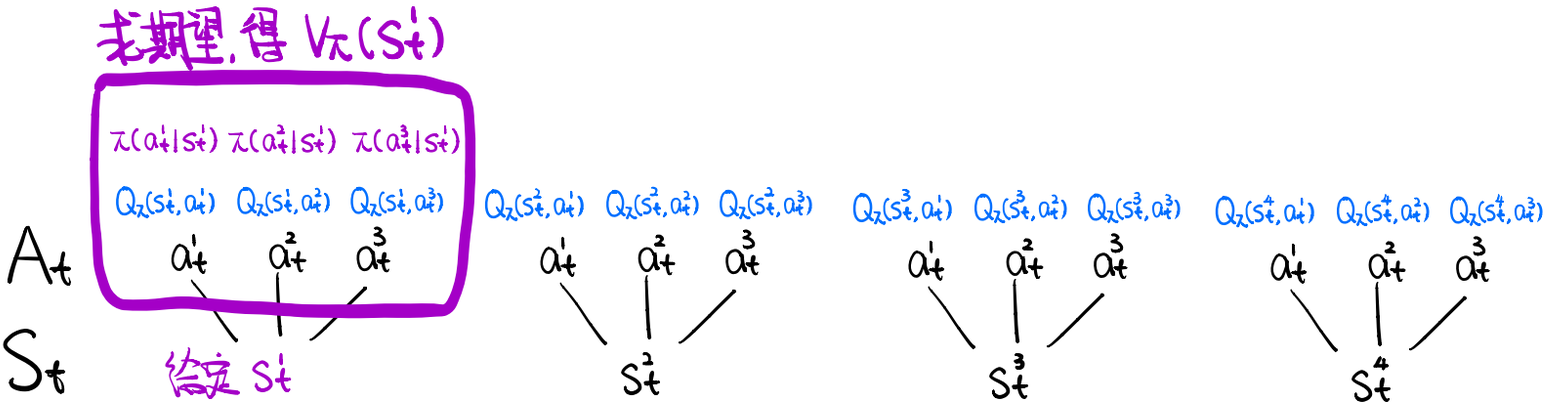
8. State-Value Function:

Definition: State-value function.

- $V_\pi(s_t) = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$. (Actions are discrete.)
- $V_\pi(s_t) = \int \pi(a|s_t) \cdot Q_\pi(s_t, a)\, da$. (Actions are continuous.)

For fixed policy π, $V_\pi(s)$ evaluates how good the situation is in state s.

$\mathbb{E}_S[V_\pi(S)]$ evaluates how good the policy π is.



综上，有两种AI控制agent的方法：

The agent can be controlled by either $\pi(a|s)$ or $Q^\star(s,a)$.

Policy-based learning

Suppose we have a good policy $\pi(a|s)$.

- Upon observing the state s_t,
- random sampling: $a_t \sim \pi(\cdot\,|s_t)$.

Value-based learning

Suppose we know the optimal action-value function $Q^\star(s,a)$.

- Upon observe the state s_t,
- choose the action that maximizes the value: $a_t = \operatorname{argmax}_a Q^\star(s_t, a)$.