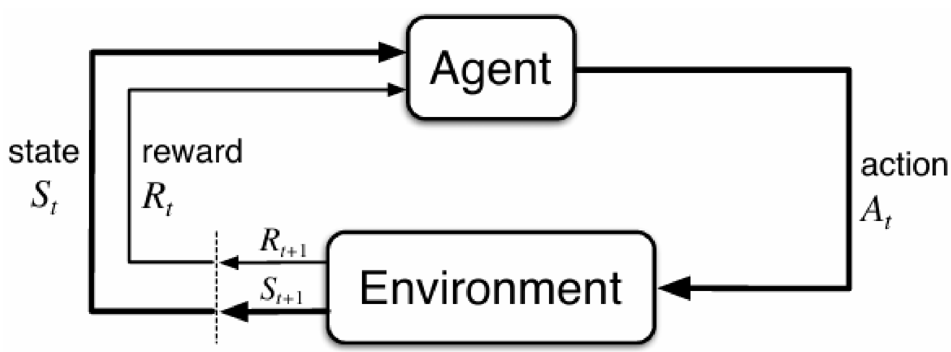
Chapter 3 Finite Markov Decision Process

在赌博机中，我们评估动作a的价值 $q_*(a)$

在MDP中，我们估计每个动作在每个状态s中的价值 $q_*(s,a)$

3.1 “智能体-环境“交互接口

进行学习及实施决策的机器被称为智能体 (agent)。 智能体之外所有与其相互作用的事物都被称为环境 (environment)。



MDP与智能体共同给出了轨迹

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

有限MDP中，状态、动作和收益的集合 (S、 A 和R）有限。

随机变量R和S具有定义明确的离散概率分布，并且只依赖于前继状态和动作。

因此 $p(s', r|s,a) = Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}$（这是定义）

由于这是一个概率，所以必然有

$$\Sigma_{s'\in S}\Sigma_{r\in R}p(s', r|s,a) = 1 \ for \ all \ s \in S, a \in A(s)$$

甚至，我们可以因此导出

$p(s'|s,a) = Pr\{S_t = s'|S_{t-1} = s, A_{t-1} = a\} = \Sigma_{r\in R}p(s', r|s,a)$

甚至可以定义“状态-动作”二元组的期望收益

$r(s,a) = E[R_t|S_{t-1} = s, A_{t-1} = a] = \Sigma_{r\in R}\Sigma_{s'\in S}rp(s', r|s,a)$

和 " 状态－动作－后继状态” 三元组的期望收益

$$r(s,a,s') = E[R_t | S_{t-1} = s, A_{t-1} = a, S_t = s'] = \Sigma_{r \in R} r \frac{p(s',r|s,a)}{p(s'|s,a)}$$

3.2 目标和收益

目标：最大化其收到的总收益。

收益信号只能用来传达什么是你想要实现的目标，而不是如何实现这个目标。譬如，下棋时不能把吃掉其他棋子作为目标，而应该直接把赢下比赛当做目标。

3.3 回报和分幕Returns and Episodes

一般我们寻求最大化期望回报，记为 G_t。最简单的情况下，回报是收益的综合：

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

我们称每个子序列为幕（episodes）T是最终时刻，实际情况下，一盘游戏，一次迷宫的结束。

显然，并不是所有过程都可以分幕，此时 $T = \infty$，再按照上述公式定义回报并不合适（容易变为无穷），引入额外一个概念——折扣(discounting)。

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

$0 \le \gamma \le 1$ 折扣率

折扣率 γ 是一个0到1之间的数。它表示了未来收益相对于即时收益的重要性。γ越接近1，未来收益的权重就越大；γ越接近0，智能体就越"目光短浅"，更注重即时收益。

显然，对于这个公式，我们将邻接时刻的回报用递归方式联系：

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

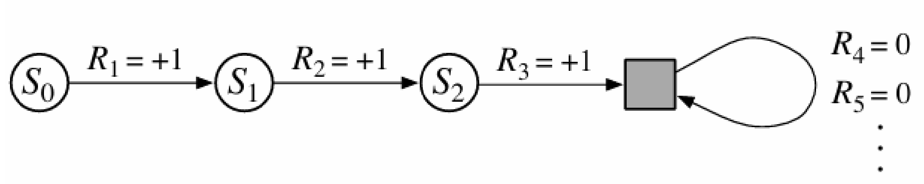
（由于太显而易见故不具体写公式了）

总结：任务分为两种，一种episodic task，一种continuing task，并不绝对（譬如书上的Exercise 3.4,3.5）

3.4 分幕式和持续性任务的统一表示法Unified Notation for Episodic and Continuing Tasks

为了描述分幕式任务在第几幕，我们使用记号 $S_{t,i}$ 表示幕i中时刻t的状态（$A_{t,i}, R_{t,i}, \pi_{t,i}, T_i$同理）

不妨将幕的终止看作一个特殊状态的吸入口



因此，所有回报都可以用如下公式表示：

$G_t = \Sigma_{k=t+1}^{T} \gamma^{k-1-1} R_k$ 允许 $T = \infty, or \gamma = 1$(不同时)

3.5 策略和价值函数Policies and Value Functions

价值函数→评估在这个状态下有多好→回报的期望值

策略是从状态到每个动作选择概率的映射

如果智能体在时刻 t选择了策略 π 那么 $\pi(a|s)$ 就是当 $S_t = s$ 时 $A_t = a$ 的概率

当前状态是 S_t, 并根据随机策略 π选择动作 用 π, p 表示 R_{t+1}的期望（Exercise 3.11)

$E_\pi(R_{t+1}|S_t) = \Sigma_{r_{t+1}} p(r_{t+1}|s_t) r_{t+1} = \Sigma_{r_{t+1}} \Sigma_a \Sigma_{s_{t+1}} p(s_{t+1}, r_{t+1}|s_t, a) \pi(a|s_t) r_{t+1}$

定义策略 π 下状态 s 的价值函数（即从状态 s 开始，智能体按照策略π进行决策所获得的回报的概率期望值）为

$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[\Sigma_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s]$ for all $s \in S$，策略π的状态价值函数

定义策略 π 下状态 s 的采取动作a的价值（即从状态 s 开始，智能体按照策略π，执行动作a后，所有可能的决策序列的期望回报）记为 $q_\pi(s, a)$。

$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] = E_\pi[\Sigma_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a]$，策略π的动作价值函数

显然，这两个定义有关系(Exercise 3.12 3.13，可以分别借助 π, p互相表达)

$v_\pi(s) = E_\pi[G_t|S_t = s] = E_\pi[E_\pi[G_t|S_t = s, A_t = a]] = E_\pi[q_\pi(s, a)] = \Sigma_{a \in A} \pi(a|s) q_\pi(s, a)$

Exercise 3.13 不懂（现在搞懂了，，，因为不太熟练所以开始没有做出来）

$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a) = E_\pi(R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a) = E_{s', r}[E_\pi(R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a, S_{t+1} = s', R_{t+1} = r)] = \Sigma_{s' \in S(t+1), r \in R(t+1)} p(s', r|s, a)[r + \gamma v_\pi(s')]$

估计v与q：求平均，Monte Carlo methods，参数化

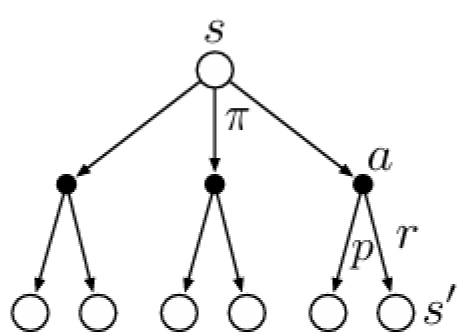
需要注意的是，在Reinforce Learning 和 dynamic programming中，价值函数一般都有递归的基本特性。(事实上就是把Exercise3.12,3.13结合起来）

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

（3.14）被称作 v_π的贝尔曼方程 (Bellman equation for v_π)，表达了相邻两个价值状态的关系。

不妨从一个状态向后观察所有可能到达的后继状态，如图所示。 其中空心圆表示一个状态，而实心圆表示一个“状态－动作“二元组。

贝尔曼方程 (3.14) 就是对所有可能性采用其出现概率进行加权平均。



Backup diagram for v_π

价值函数 v_π是贝尔曼方程的唯一解。

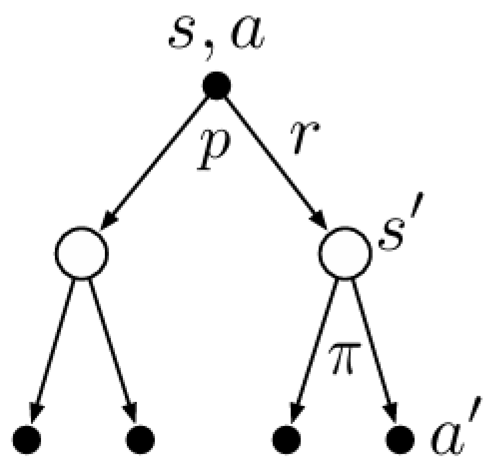
称上图为回溯图。回溯操作就是将后继状态（或“状态－动作“ 二元组）的价值信息回传给当前时刻的状态（或“状态－动作“二元组）。

Example 3.5

不会 Exercise 3.15：没懂为什么符号不重要？是因为 γ会把他们弱化吗？

Example 3.6

Exercise 3.17 q_π的贝尔曼方程 (Bellman equation for q_π)



q_π backup diagram

$$\begin{aligned} q_\pi(s,a) &\doteq E_\pi[G_t|S_t=s,A_t=a] \\ &= E_\pi[R_{t+1}+\gamma G_{t+1}|S_t=s,A_t=a] \\ &= \sum_{s',r} p(s',r|s,a)[r+\gamma \sum_{a'} \pi(a'|s') E_\pi[G_{t+1}|S_{t+1}=s',A_{t+1}=a']] \\ &= \sum_{s',r} p(s',r|s,a)[r+\gamma \sum_{a'} \pi(a'|s') q_\pi(s',a')] \end{aligned}$$

Exercise 3.18 3.19 (Exercise 3.12 3.13的另一种理解)

3.6 最优策略和最优价值函数Optimal Policies and Optimal Value Functions

最优策略：

若对于所有的 $s \in S, \pi \geq \pi$’，那么应当 $v_\pi(s) \geq v_{\pi'}(s)$。总会存在至少一个策略不劣于其他所有的策略，这就是最优策略。

用 π_*表示。定义:for any $s \in S, v_*(s) = max_\pi v_\pi(s)$

最优的策略也共享相同的最优动作价值函数，记为 q_*

定义:for any $s \in S, a \in A, q_*(s,a) = max_\pi q_\pi(s,a)$

有如下等式（该等式阐述了状态和价值的一致性）

$$q_*(s,a) = E(R_{t+1} + \gamma v_*(S_{t+1})|S_t = s, A_t = a)$$

Example 3.7（Exercise 3.25-3.29就是在教为什么要这么写这些方程）

需要注意的是，贝尔曼方程中状态和价值有一致性条件，因此如下等式才得以成立。

$$
\begin{aligned}
v_*(s) &= \max_{a\in\mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t=s, A_t=a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1}+\gamma G_{t+1} \mid S_t=s, A_t=a] && \text{(by (3.9))} \\
&= \max_a \mathbb{E}[R_{t+1}+\gamma v_*(S_{t+1}) \mid S_t=s, A_t=a] && (3.18) \\
&= \max_a \sum_{s',r} p(s',r|s,a)\big[r+\gamma v_*(s')\big]. && (3.19)
\end{aligned}
$$

最后两个等式就是 v_*的贝尔曼最优方程的两种形式。 q_*的贝尔曼最优方程如下

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}\Big[R_{t+1}+\gamma \max_{a'} q_*(S_{t+1},a') \;\Big|\; S_t=s, A_t=a\Big] \\
&= \sum_{s',r} p(s',r|s,a)\Big[r+\gamma \max_{a'} q_*(s',a')\Big]. && (3.20)
\end{aligned}
$$

对于有限 MDP 来说， v_* 的贝尔曼最优方程（3.19) 有独立千策略的唯一解。

由于 v_*的定义本身就包含了未来的信息，因此一旦确定，最优策略的确定也非常容易。如果在一个策略中，只有这些动作的概率非零，那么这个策略就是一个最优策略。（贪心）

在给定 q_*的清况下，选择最优动作的过程变得更加容易。 给定 q_*， 甚至不需要进行单步搜索的过程，也就是说，对于任意状态 s，智能体只要找到使得 $q_*(s,a)$ 最大化的动作a就可以了。

~~哈哈哈，因为上面这两段话是复制的所以我完全不懂呢~~

Exercise 3.29 ***非常重要

3.29 Rewrite the four Bellman equations for the four value functions ($v_\pi, v_*, q_\pi, and\ q_*$) in terms of the three argument function p (3.4) and the two-argument function r (3.5).

- 先全列出来:

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

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

- 依次替换:

$$\begin{aligned} v_\pi &= \textstyle\sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r+\gamma v_\pi(s')] \\ &= \textstyle\sum_a \pi(a|s)[\sum_r \sum_{s'} p(s',r|s,a)r + \sum_{s'} \sum_r p(s',r|s,a)\gamma v_\pi(s')] \\ &= \textstyle\sum_a \pi(a|s)[r(s,a) + \sum_{s'} p(s'|s,a)\gamma v_\pi(s')] \end{aligned}$$

$$\begin{aligned} v_* &= \textstyle\max_a \sum_{s',r} p(s',r|s,a)[r+\gamma v_*(s')] \\ &= \textstyle\max_a[\sum_r \sum_{s'} p(s',r|s,a)r + \sum_{s'} \sum_r p(s',r|s,a)\gamma v_*(s')] \\ &= \textstyle\max_a[r(s,a) + \sum_{s'} p(s'|s,a)\gamma v_*(s')] \end{aligned}$$

$$\begin{aligned} q_\pi(s,a) &= \textstyle\sum_{s',r} p(s',r|s,a)[r+\gamma \sum_{a'} \pi(a'|s')q_\pi(s',a')] \\ &= \textstyle\sum_r \sum_{s'} p(s',r|s,a)r + \sum_{s'} \sum_r p(s',r|s,a)\gamma \sum_{a'} \pi(a'|s')q_\pi(s',a') \\ &= \textstyle r(s,a) + \sum_{s'} p(s'|s,a)\gamma \sum_{a'} \pi(a'|s')q_\pi(s',a') \end{aligned}$$

$$\begin{aligned} q_*(s,a) &= \textstyle\sum_{s',r} p(s',r|s,a)[r+\gamma \max_{a'} q_\pi(s',a')] \\ &= \textstyle\sum_r \sum_{s'} p(s',r|s,a)r + \sum_{s'} \sum_r p(s',r|s,a)\gamma \max_{a'} q_\pi(s',a') \\ &= \textstyle r(s,a) + \sum_{s'} p(s'|s,a)\gamma \max_{a'} q_\pi(s',a') \end{aligned}$$