**17.1 序贯决策问题**

序列决策问题：效用函数依赖于一系列的状态和动作（一个环境历史）

规定对于动作a从s到s‘的每次转移，智能体都会收到一个奖励R(s, a, s’)，奖励可正可负，但绝对值有明确界限Rmax

则历史环境的效用等于目前获得的奖励的总和

马尔科夫决策过程（MDP）：一个**完全可观察**、随机环境的顺序决策问题

MDP包括：

一组状态（具有初始状态s0）

每个状态中的一组 ACTIONS（s）

转换模型P(s’| s, a)

奖励函数 R(s, a, s’)

策略的质量用该策略可能产生的环境历史的期望效用来衡量

**17.1.1 时间上的效用**

效用函数：Uh([s0, a0, s1, a1 . . . , sn]).

决策**有限期**：固定时间N，对之后发生什么都不重要

Uh([s₀, a₀, s₁, a₁, . . . , sN+k]) = Uh([s₀, a₀, s₁, a₁, . . . , sN])

有限期的条件下，给定状态下的最优动作可能取决于还剩多少时间。

依赖于时间的策略被称为非平稳的（nonstationary）策略

决策**无限期**：无固定时间限制

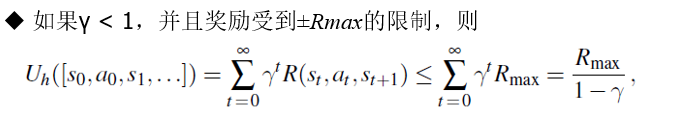
没有理由在不同的时间表现不同。

最佳行动只取决于当前状态，最佳策略是平稳的。

**额外折扣奖励：**



越远的奖励折合效用越小。折扣因子（discount factor）0<γ<1



折扣奖励的作用/合理性：①折扣因子描述了智能体偏好当前奖励甚于未来奖励的程度：γ 接近0，不愿意等待；γ接近1，愿意等待长期回报

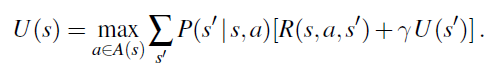
②减少了无限序列的复杂性，因为效用是有限的。

**17.1.2 最优策略与状态效用**

对于无期限，使用折扣效用奖励的最优决策是唯一的，记为π\*

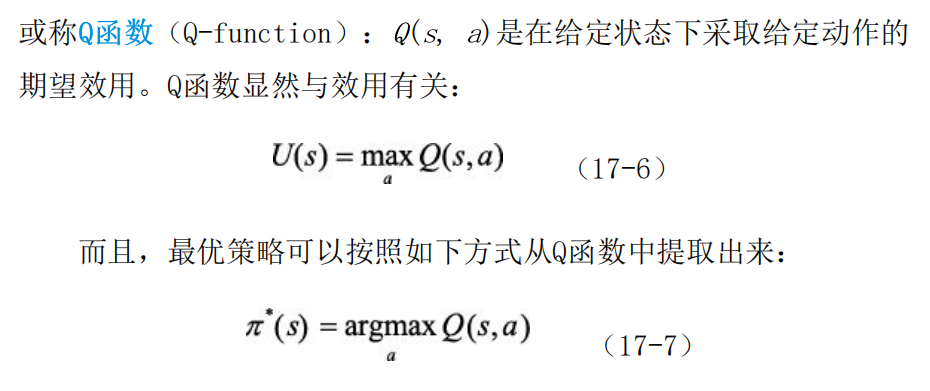
**状态效用函数**

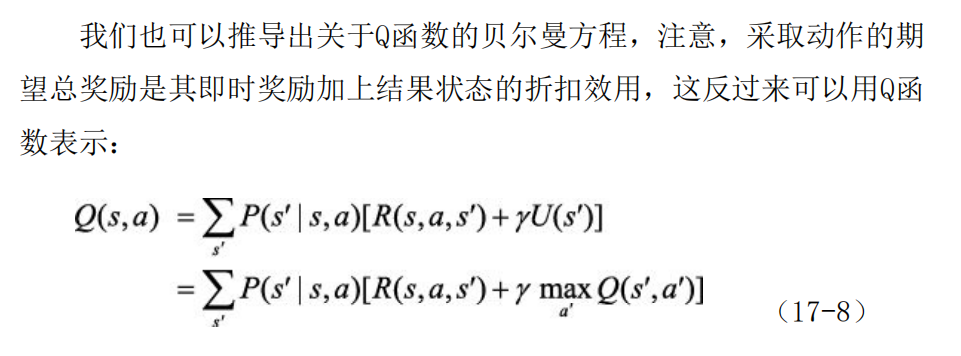
**假设智能体选择最优动作，状态效用是下一次转移的期望奖励加上下一个状态的折扣效用，即：**贝尔曼方程（Bellman equation）



**行动效用函数**

Q(s,a)：在**给定状态s**下**采取行动a的预期效用**





**17.1.3 奖励规模**

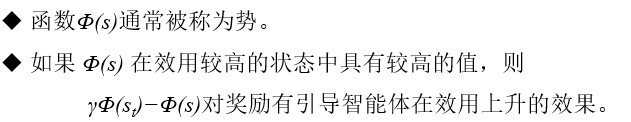
效用函数的规模可以是任意的：仿射变换（线性变换）使最优决策保持不变。

然而效用的加性奖励分解导致在定义奖励时具有更大的自由度。

**函数设计定理（shaping theorem）**

设Φ(s)是关于状态s的任意函数，以下变换能够保持最优策略不变：

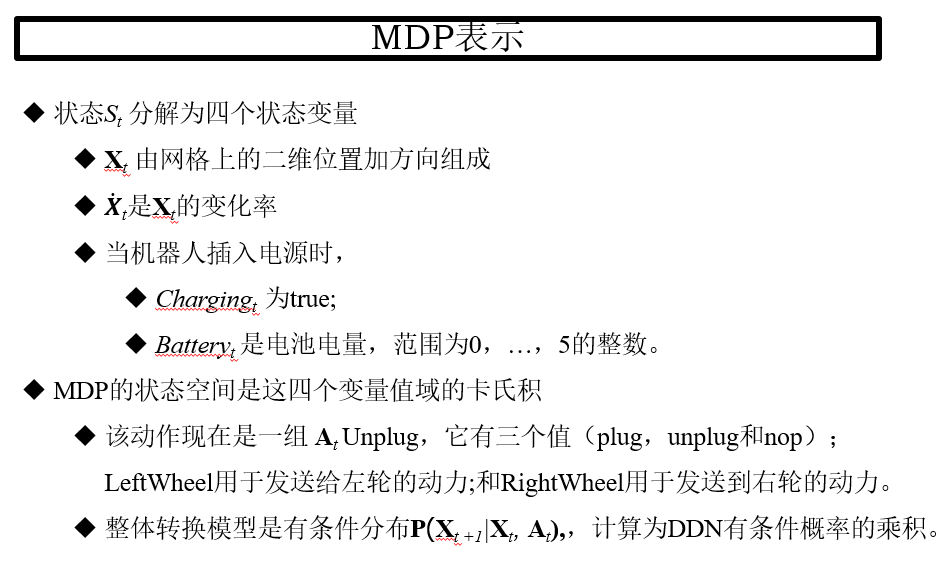


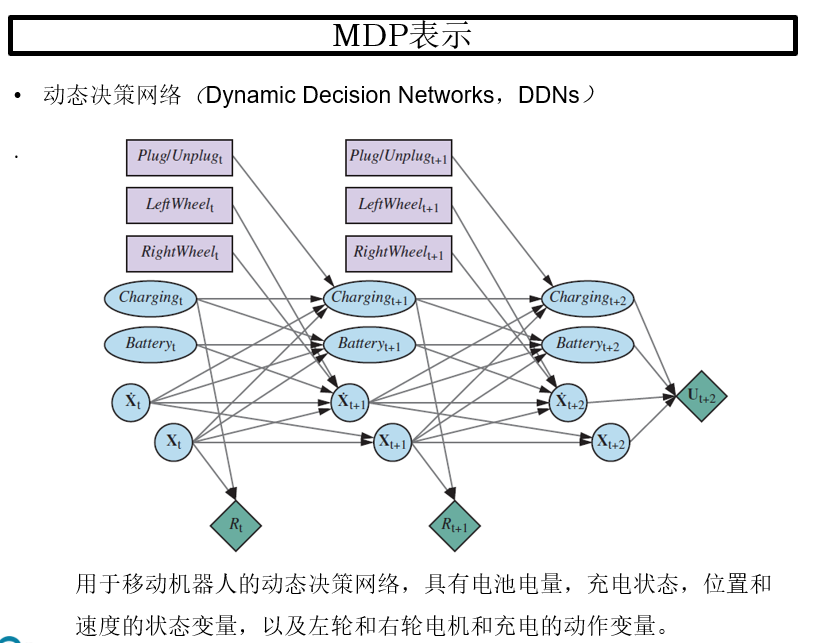


**如果奖励函数处处为零，那么所有的策略都是最优的**

**17.1.4 MDP表示**

**电池世界的MDP表示**





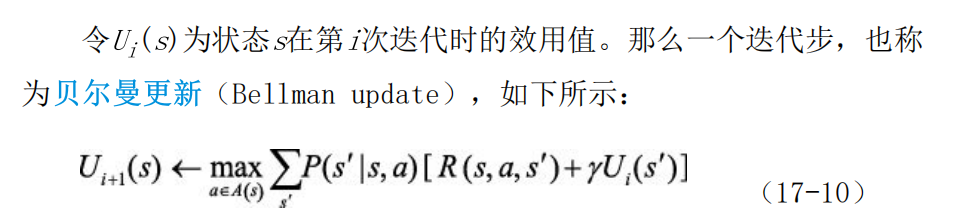
**17.2 MDP的算法**

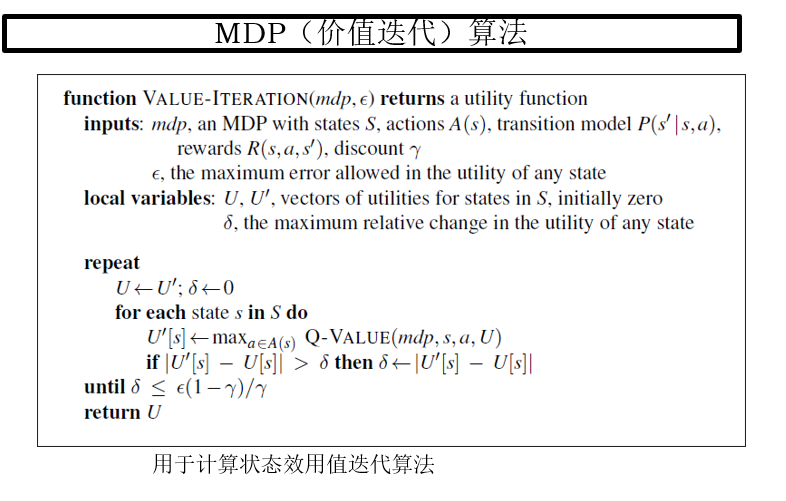
**17.2.1 价值迭代**

如果有n个可能的状态，那么就有n个贝尔曼方程式

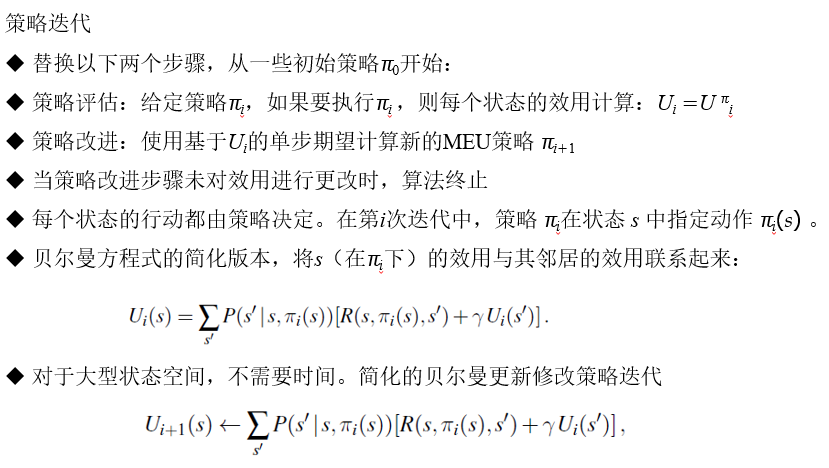
贝尔曼方程组为max方程，无法直接组合求解。可使用迭代方法：

我们从效用的任意初始值出发计算方程的右侧，并将其值代入方程的左侧，从而根据其邻居的效用更新每个状态的效用。我们重复这个过程，直到达到平衡。

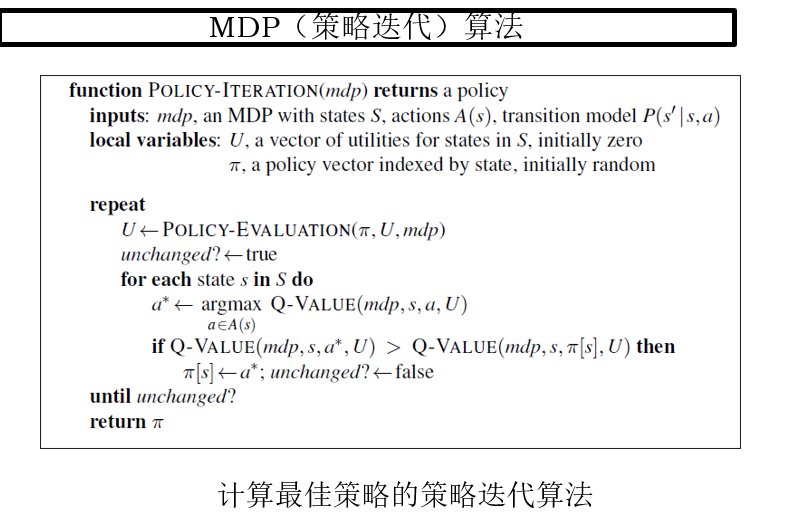




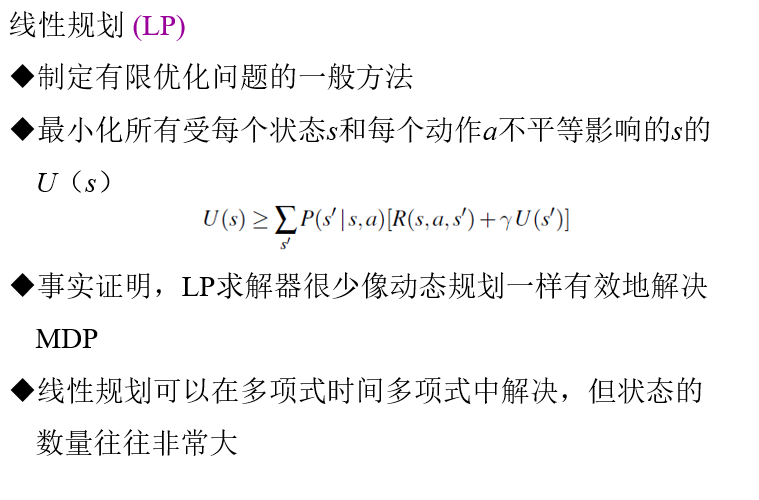
**17.2.2 策略迭代**



如果策略改进步骤对效用不产生任何改变，该算法将终止。我们已经知道效用函数Ui是贝尔曼更新的不动点，因此它是贝尔曼方程的解，并且 π定是最优策略。



**17.2.3 线性规划**



**17.2.4 MDP的在线算法**

