# 第1章 基本概念

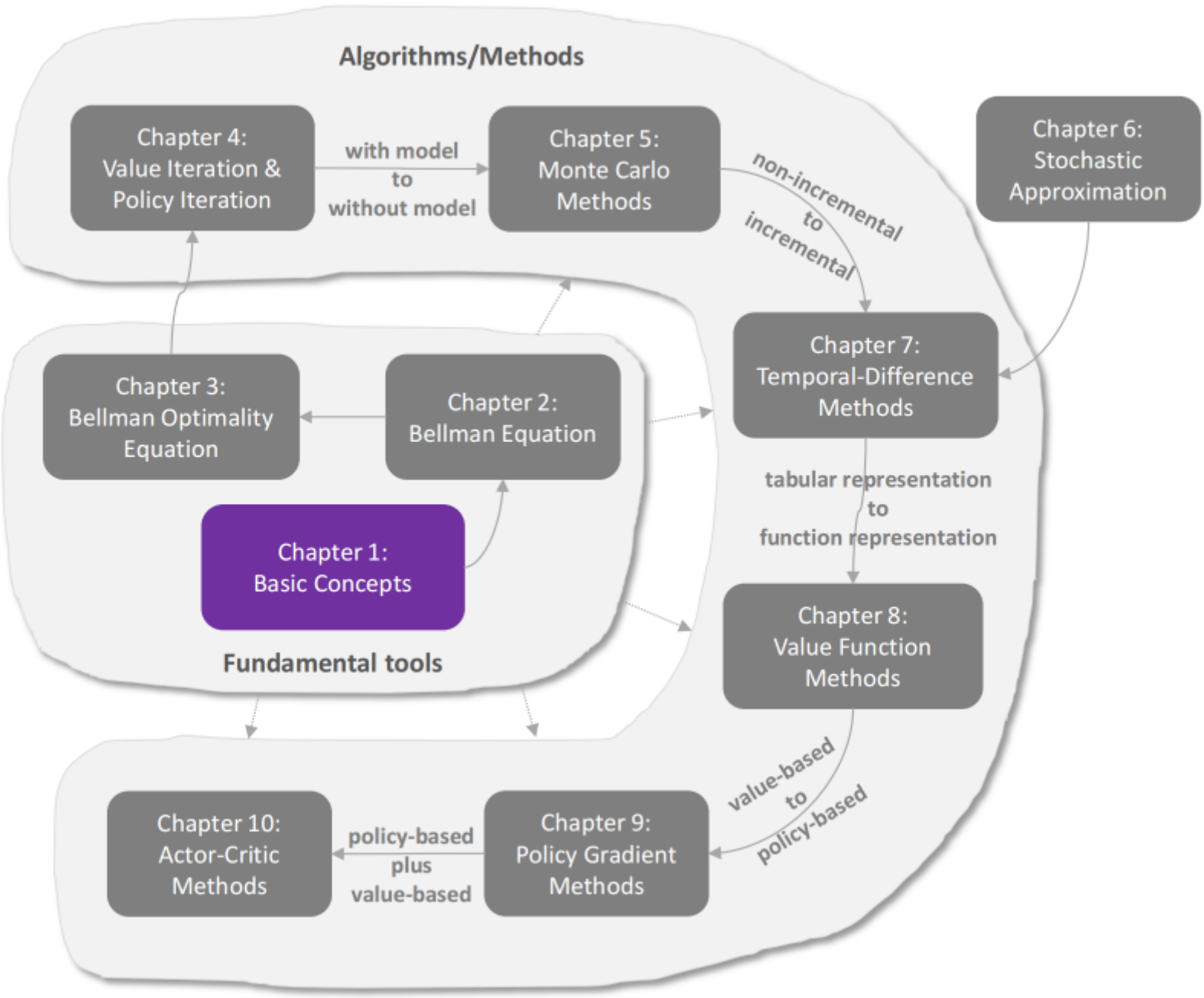



**图 1.1：我们在本书中的位置。**

本章介绍了**强化学习（reinforcement learning）**的基本概念。这些概念非常重要，因为它们将在本书中被广泛使用。我们首先通过示例来介绍这些概念，然后在**马尔可夫决策过程（Markov decision processes）**的框架下对其进行形式化描述。

## 1.1 网格世界示例

考虑如图 1.2 所示的一个示例，其中一个机器人在网格世界中移动。这个机器人被称为**智能体（agent）**，可以在网格中的相邻单元格之间移动。在每个时间步（time step），它可以占据一个单元格。白色单元格是**可进入的（accessible）**，橙色单元格是**禁止的（forbidden）**。还有一个机器人想要到达的**目标（target）**单元格。我们将在这本书中一直使用这种网格世界的例子，因为它们对于说明新概念和算法非常直观。

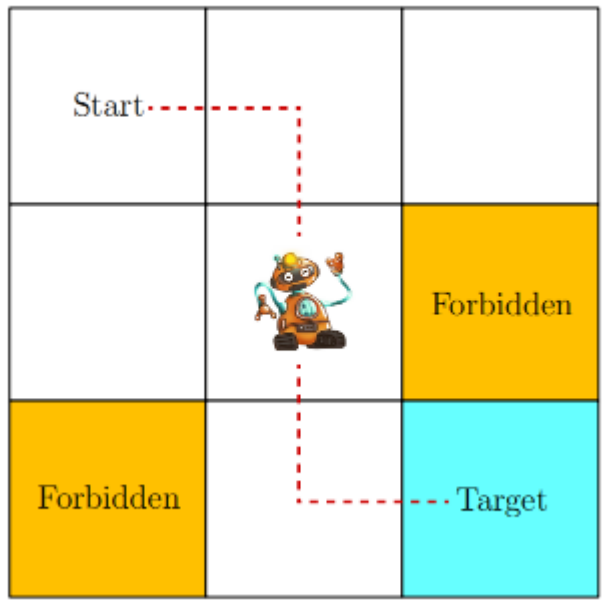



**图 1.2：本书中使用的网格世界示例。**

**智能体**的最终目标是找到一个“好”的策略，使其能够从任何初始单元格出发并到达目标单元格。如何定义策略的“好坏”呢？其核心思想是，智能体应该在不进入任何禁止单元格、不绕不必要的弯路、也不与网格边界碰撞的情况下到达目标。

如果智能体知道网格世界的地图，规划一条到达目标单元格的路径将是微不足道的（trivial）。但如果智能体预先不知道关于环境的任何信息，任务就变得非平凡（nontrivial）了。这时，智能体必须与环境交互，通过试错（trial and error）来找到一个好的策略。为此，本章其余部分介绍的概念是必不可少的。

## 1.2 状态和动作

首先要介绍的概念是**状态（state）**，它描述了智能体相对于环境的状况。在网格世界示例中，状态对应于智能体的位置。因为有九个单元格，所以也有九个状态。它们的记为 $s_1, s_2, \ldots, s_9$，如图 1.3(a) 所示。所有状态的集合称为*状态空间（state space）*，记为 $\mathcal{S} = \{s_1, \ldots, s_9\}$。

对于每个状态，智能体可以采取五种可能的*动作（actions）*：向上移动、向右移动、向下移动、向左移动和保持不动。这五种动作分别记为 $a_1, a_2, \ldots, a_5$（见图 1.3(b)）。所有动作的集合称为*动作空间（action space）*，记为 $\mathcal{A} = \{a_1, \ldots, a_5\}$。不同的状态可以有不同的动作空间。例如，考虑到在状态 $s_1$ 采取 $a_1$ 或 $a_4$ 会导致与边界碰撞，我们可以将状态 $s_1$ 的动作空间设置为 $\mathcal{A}(s_1) = \{a_2, a_3, a_5\}$。在本书中，我们考虑最一般的情况：对于所有的 $i$，$\mathcal{A}(s_i) = \mathcal{A} = \{a_1, \ldots, a_5\}$。

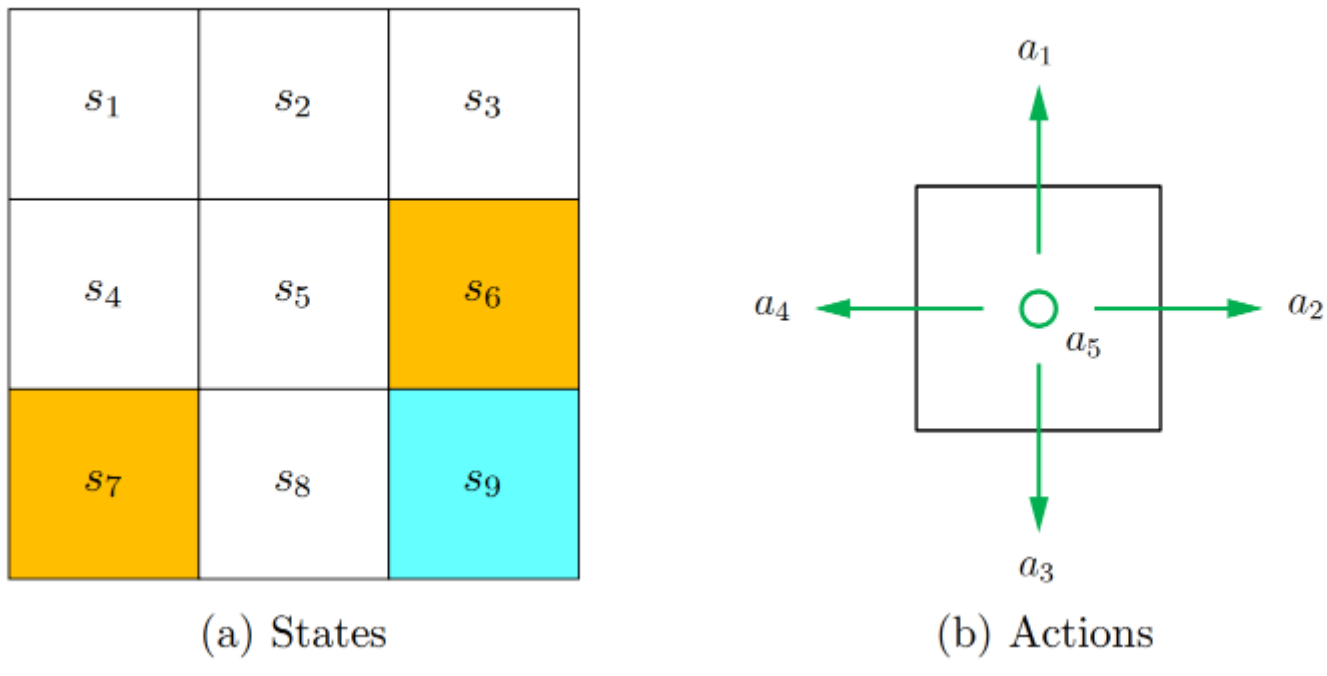



**图 1.3：状态和动作概念的图示。(a) 有 $\{s_1, \ldots, s_9\}$ 九个状态。(b) 每个状态有五种可能的动作 $\{a_1, a_2, a_3, a_4, a_5\}$。**

## 1.3 状态转移

当采取一个动作时，智能体可能会从一个状态移动到另一个状态。这样的过程称为**状态转移（state transition）**。例如，如果智能体处于状态 $s_1$ 并选择动作 $a_2$（即向右移动），那么智能体移动到状态 $s_2$。这个过程可以表示为

$$s_1 \xrightarrow{a_2} s_2.$$

我们接下来考察两个重要的例子。

- 当智能体试图越过边界时，例如在状态 $s_1$ 采取动作 $a_1$ ，下一个状态是什么？答案是智能体将被反弹回来，因为智能体不可能离开状态空间。因此，我们有 $s_1 \xrightarrow{a_1} s_1$.
- 当智能体试图进入一个禁止单元格时，例如在状态 $s_5$ 采取动作 $a_2$ ，下一个状态是什么？可能会遇到两种不同的情况。在第一种情况中，虽然 $s_6$ 是禁止的，但它仍然是*可进入的（accessible）*。在这种情况下，下一个状态是 $s_6$ ；因此，状态转移过程是 $s_5 \xrightarrow{a_2} s_6$ 。在第二种情况中， $s_6$ 是*不可进入的（not accessible）*，例如因为它被墙壁包围。在这种情况下，如果智能体试图向右移动，它将被反弹回 $s_5$ ；因此，状态转移过程是 $s_5 \xrightarrow{a_2} s_5$ 。

我们应该考虑哪种情况？答案取决于物理环境。在本书中，我们考虑第一种情况，即禁止单元格是可进入的，尽管进入它们可能会受到惩罚。这种情况更普遍也更有趣。此外，由于我们考虑的是仿真任务，我们可以根据自己的喜好定义状态转移过程。在实际应用中，状态转移过程由现实世界的动力学决定。

状态转移过程是为每个状态及其关联的动作定义的。这个过程可以用如表 1.1 所示的表格来描述.在这个表格中，每一行.对应一个状态，每一列对应一个动作。每个单元格指示了智能体在相应状态下采取某个动作后将转移到的下一个状态。

**表 1.1：状态转移过程的表格表示。每个单元格指示了智能体在某个状态采取一个动作后将转移到的下一个状态。**

| | $a_1$（向上） | $a_2$（向右） | $a_3$（向下） | $a_4$（向左） | $a_5$（保持） |
|---|---|---|---|---|---|
| $s1$ | $s1$ | $s2$ | $s4$ | $s1$ | $s1$ |
| $s2$ | $s2$ | $s3$ | $s5$ | $s1$ | $s2$ |
| $s3$ | $s3$ | $s3$ | $s6$ | $s2$ | $s3$ |
| $s4$ | $s1$ | $s5$ | $s7$ | $s4$ | $s4$ |
| $s5$ | $s2$ | $s6$ | $s8$ | $s4$ | $s5$ |
| $s6$ | $s3$ | $s6$ | $s9$ | $s5$ | $s6$ |
| $s7$ | $s4$ | $s8$ | $s7$ | $s7$ | $s7$ |
| $s8$ | $s5$ | $s9$ | $s8$ | $s7$ | $s8$ |
| $s9$ | $s6$ | $s9$ | $s9$ | $s8$ | $s9$ |

从数学上讲，状态转移过程可以用条件概率来描述。例如，对于 $s_1$ 和 $a_2$，条件概率分布为

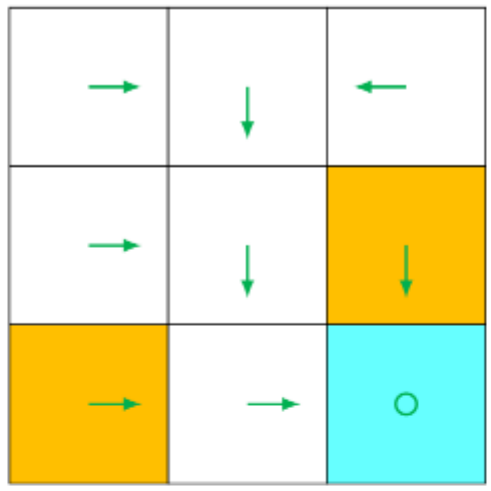
$$p(s_1|s_1,a_2) = 0,$$
$$p(s_2|s_1,a_2) = 1,$$
$$p(s_3|s_1,a_2) = 0,$$
$$p(s_4|s_1,a_2) = 0,$$
$$p(s_5|s_1,a_2) = 0,$$

这表明，当在 $s_1$ 采取 $a_2$ 时，智能体移动到 $s_2$ 的概率为 1，而智能体移动到其他状态的概率为 0。结果是，在 $s_1$ 采取动作 $a_2$ 将必然导致智能体转移到 $s_2$ 。条件概率的预备知识在附录 A 中给出。强烈建议读者熟悉概率论，因为它是学习强化学习所必需的。

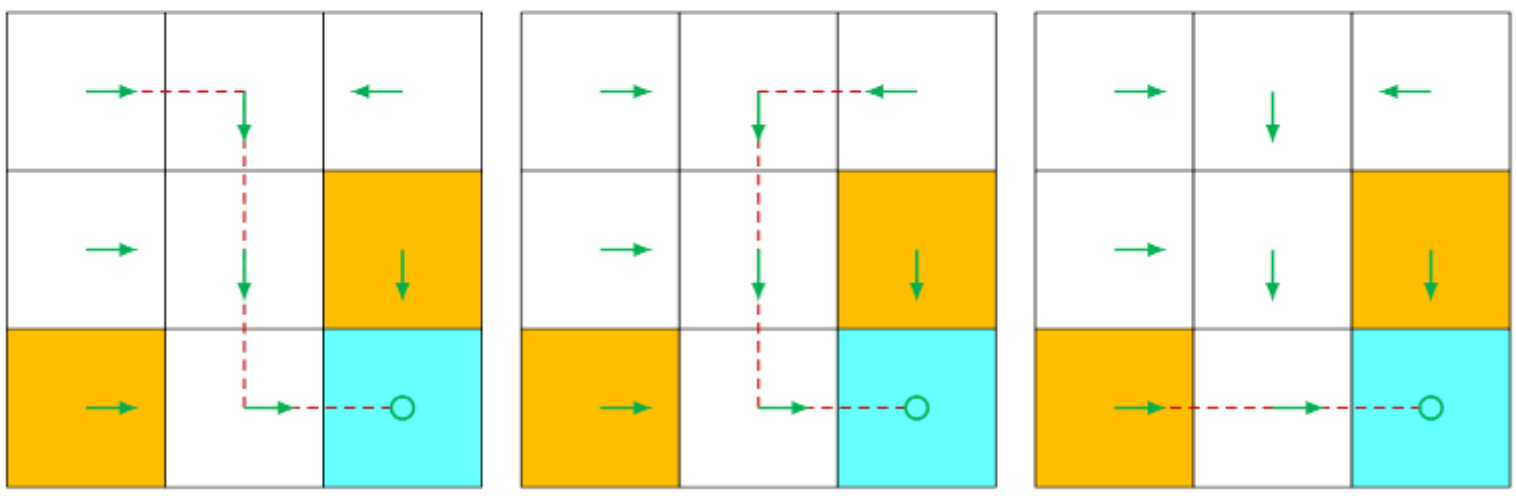
虽然表格表示法很直观，但它只能描述*确定性（deterministic）*的状态转移。一般而言，状态转移可以是*随机的（stochastic）*，并且必须由条件概率分布来描述。例如，当网格上刮起随机阵风时，如果在 $s_1$ 采取动作 $a_2$ ，智能体可能会被吹到 $s_5$ 而不是 $s_2$ 。在这种情况下，我们有 $p(s_5|s_1,a_2) > 0$ 。尽管如此，为了简单起见，我们在本书的网格世界示例中仅考虑确定性的状态转移。

## 1.4 策略

*策略（policy）* 告诉智能体在每个状态下应该采取哪些动作。直观地说，策略可以描绘为箭头（见图 1.4(a)）。遵循一个策略，智能体可以生成一条从初始状态开始的轨迹（见图 1.4(b)）。



(a) A deterministic policy



(b) Trajectories obtained from the policy

**(a) 一个确定性策略. (b) 从该策略获得的轨迹**

**图 1.4：用箭头表示的策略，以及从不同初始状态出发获得的几条轨迹。**

从数学上讲，策略可以用条件概率来描述。将图 1.4 中的策略记为 $\pi(a|s)$，这是一个为*每个*状态定义的条件概率分布函数。例如，针对 $s_1$ 的策略是

$$\begin{aligned}\pi(a_1|s_1) &= 0,\\ \pi(a_2|s_1) &= 1,\\ \pi(a_3|s_1) &= 0,\\ \pi(a_4|s_1) &= 0,\\ \pi(a_5|s_1) &= 0,\end{aligned}$$

这表明在状态 $s_1$ 采取动作 $a_2$ 的概率为 1，而采取其他动作的概率为 0。

上述策略是*确定性（deterministic）*的。一般而言，策略可能是*随机的（stochastic）*。例如，图 1.5 所示的策略就是随机的：在状态 $s_1$，智能体可能会采取动作向右或向下移动。采取这两个动作的概率是相同的（均为 0.5）。在这种情况下，$s_1$ 的策略是

$$\begin{aligned}\pi(a_1|s_1) &= 0,\\ \pi(a_2|s_1) &= 0.5,\\ \pi(a_3|s_1) &= 0.5,\\ \pi(a_4|s_1) &= 0,\\ \pi(a_5|s_1) &= 0.\end{aligned}$$

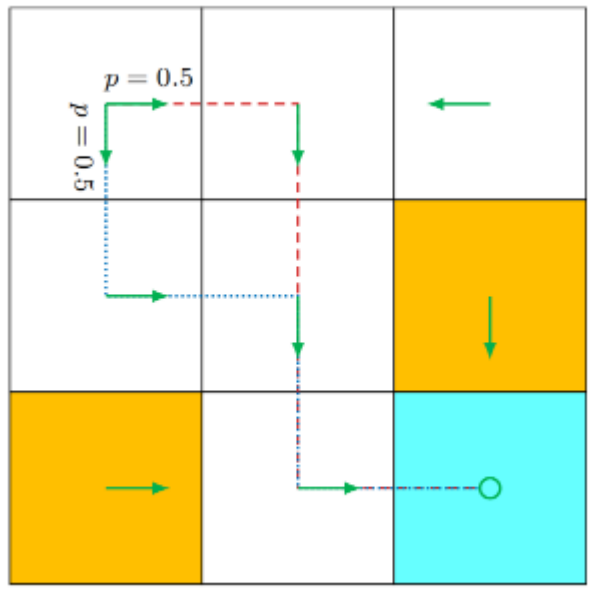



图 1.5：一个随机策略。在状态 $s_1$，智能体可能会以 0.5 的相等概率向右或向下移动。

由条件概率表示的策略可以存储为表格。例如，表 1.2 表示了图 1.5 中描述的随机策略。第 $i$ 行和第 $j$ 列的条目是在第 $i$ 个状态下采取第 $j$ 个动作的概率。这种表示形式称为**表格表示（tabular representation）**。我们将在第 8 章介绍另一种将策略表示为参数化函数的方法。

表 1.2：策略的表格表示。每个条目指示了在某个状态下采取某个动作的概率。

| | $a_1$ (upward) | $a_2$ (rightward) | $a_3$ (downward) | $a_4$ (leftward ) | $a_5$ (still) |
|---|---|---|---|---|---|
| $s_1$ | 0 | 0.5 | 0.5 | 0 | 0 |
| $s_2$ | 0 | 0 | 1 | 0 | 0 |
| $s_3$ | 0 | 0 | 0 | 1 | 0 |
| $s_4$ | 0 | 1 | 0 | 0 | 0 |
| $s_5$ | 0 | 0 | 1 | 0 | 0 |
| $s_6$ | 0 | 0 | 1 | 0 | 0 |
| $s_7$ | 0 | 1 | 0 | 0 | 0 |
| $s_8$ | 0 | 1 | 0 | 0 | 0 |
| $s_9$ | 0 | 0 | 0 | 0 | 1 |

## 1.5 奖励

**奖励（Reward）**是强化学习中最独特的概念之一。

在某个状态下执行一个动作后，智能体会获得一个奖励，记为 $r$ ，作为来自环境的反馈。奖励是状态 $s$ 和动作 $a$ 的函数。因此，它也记为 $r(s,a)$。它的值可以是正实数、负实数或零。不同的奖励对智能体最终将学到的策略有不同的影响。一般来说，通过正奖励，我们鼓励智能体采取相应的动作。通过负奖励，我们阻止（或抑制）智能体采取该动作。

在网格世界示例中，奖励设计如下：

- 如果智能体试图越过边界，令 $r_{\text{boundary}} = -1$。
- 如果智能体试图进入禁止单元格，令 $r_{\text{forbidden}} = -1$。
- 如果智能体到达目标状态，令 $r_{\text{target}} = +1$。
- 否则，智能体获得的奖励为 $r_{\text{other}} = 0$。

需要特别注意目标状态 $s_9$。奖励过程不一定在智能体到达 $s_9$ 后终止。如果智能体在 $s_9$ 采取动作 $a_5$，下一个状态仍然是 $s_9$，且奖励为 $r_{\text{target}} = +1$。如果智能体采取动作 $a_2$，下一个状态也是 $s_9$，但奖励为 $r_{\text{boundary}} = -1$。

奖励可以解释为一种人机接口，通过它我们可以引导智能体按照我们的期望行事。例如，通过上述设计的奖励，我们可以期望智能体倾向于避免越过边界或踏入禁止单元格。设计适当的奖励是强化学习中的重要步骤。然而，对于复杂的任务，这一步是非平凡的（nontrivial），因为它可能需要用户对给定问题有很好的理解。尽管如此，这可能仍然比使用其他需要专业背景或对给定问题有深刻理解的方法来解决问题要容易得多。

执行动作后获得奖励的过程可以直观地表示为一个表格，如表 1.3 所示. 表格的每一行对应一个状态，每一列对应一个动作。表格中每个单元格的值表示在某个状态下采取某个动作所能获得的奖励。

初学者可能会问这样一个问题：如果给定了奖励表，我们是否可以通过简单地选择具有最大奖励的动作来找到好的策略？答案是否定的。这是因为这些奖励是即时奖励（immediate rewards），是在采取动作后立即获得的。为了确定一个好的策略，我们必须考虑长期获得的总奖励（total reward）（更多信息请参见 1.6 节）。具有最大即时奖励的动作可能不会导致最大的总奖励。

尽管直观，但表格表示法只能描述*确定性（deterministic）*的奖励过程。更通用的方法是使用条件概率 $p(r|s,a)$ 来描述奖励过程。例如，对于状态 $s_1$，我们有

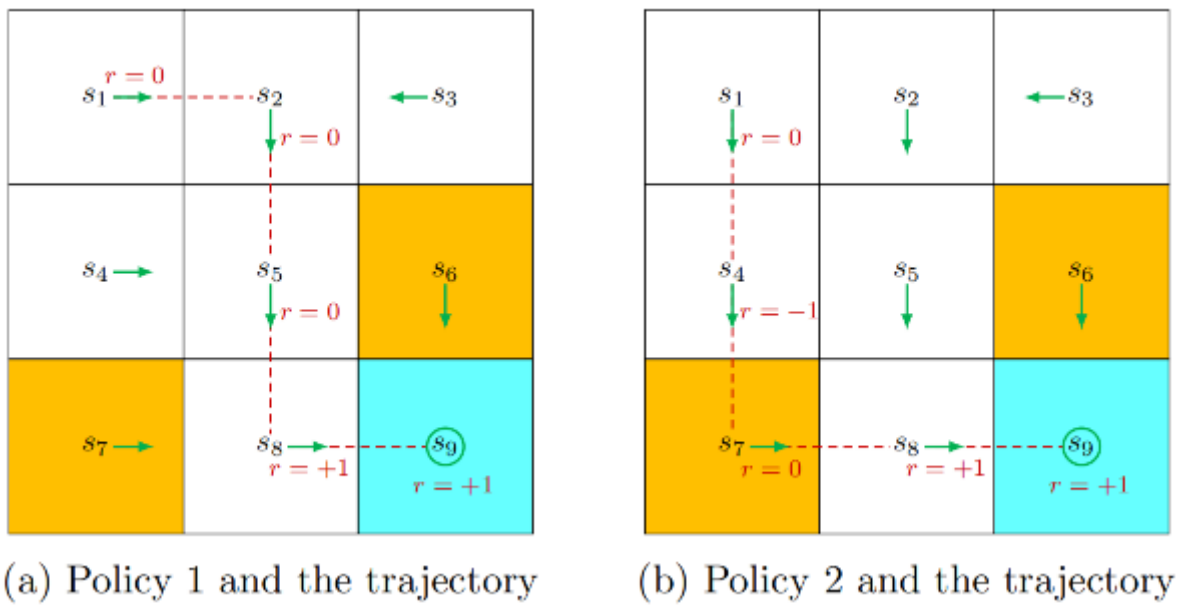
$$p(r=-1|s_1,a_1)=1, \quad p(r\neq -1|s_1,a_1)=0.$$

**表 1.3：获取奖励过程的表格表示。这里，该过程是确定性的。每个单元格指示了智能体在给定状态下采取某个动作后可以获得多少奖励。**

| | $a_1$(向上) | $a_2$(向右) | $a_3$(向下) | $a_4$(向左) | $a_5$（保持） |
|---|---|---|---|---|---|
| $s_1$ | $r_{\text{boundary}}$ | 0 | 0 | $r_{\text{boundary}}$ | 0 |
| $s_2$ | $r_{\text{boundary}}$ | 0 | 0 | 0 | 0 |
| $s_3$ | $r_{\text{boundary}}$ | $r_{\text{boundary}}$ | $r_{\text{forbidden}}$ | 0 | 0 |
| $s_4$ | 0 | 0 | $r_{\text{forbidden}}$ | $r_{\text{boundary}}$ | 0 |
| $s_5$ | 0 | $r_{\text{forbidden}}$ | 0 | 0 | 0 |
| $s_6$ | 0 | $r_{\text{boundary}}$ | $r_{\text{target}}$ | 0 | $r_{\text{forbidden}}$ |
| $s_7$ | 0 | 0 | $r_{\text{boundary}}$ | $r_{\text{boundary}}$ | $r_{\text{forbidden}}$ |
| $s_8$ | 0 | $r_{\text{target}}$ | $r_{\text{boundary}}$ | $r_{\text{forbidden}}$ | 0 |
| $s_9$ | $r_{\text{forbidden}}$ | $r_{\text{boundary}}$ | $r_{\text{boundary}}$ | 0 | $r_{\text{target}}$ |

这表明，当在 $s_1$ 采取 $a_1$ 时，智能体确定地获得 $r=-1$。在这个例子中，奖励过程是确定性的。一般而言，它可以是随机的。例如，如果一个学生努力学习，他或她可能会收到正向奖励（例如，考试分数更高），但奖励的具体数值可能是不确定的。

## 1.6 轨迹、回报和回合




**(a) 策略 1 及其轨迹. (b) 策略 2 及其轨迹**

**图 1.6：遵循两个策略获得的轨迹。轨迹由红色虚线表示。**

**轨迹（trajectory）**是一条状态-动作-奖励链。例如，给定图 1.6(a) 所示的策略，智能体可以沿着如下轨迹移动：

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9.$$

该轨迹的**回报（return）**定义为沿轨迹收集的所有奖励的总和：

$$\text{return} = 0 + 0 + 0 + 1 = 1. \tag{1.1}$$

回报也被称为**总奖励（total rewards）**或**累积奖励（cumulative rewards）**。

回报可用于 **评估（evaluate） 策略**。例如，我们可以通过比较图 1.6 中两个策略的回报来评估它们。具体而言，从 $s_1$ 开始，左侧策略获得的回报如上计算为 1。对于右侧策略，从 $s_1$ 开始，生成的轨迹如下：

$$s_1 \xrightarrow[r=0]{a_3} s_4 \xrightarrow[r=-1]{a_3} s_7 \xrightarrow[r=0]{a_2} s_8 \xrightarrow[r=+1]{a_2} s_9.$$

相应的回报是

$$\text{return} = 0 - 1 + 0 + 1 = 0. \tag{1.2}$$

(1.1) 和 (1.2) 中的回报表明，左侧策略优于右侧策略，因为其回报更大。这一数学结论与直觉一致，即右侧策略较差，因为它经过了一个禁止单元格。

回报由**即时奖励（immediate reward）***和***未来奖励（future rewards）**组成。

这里，即时奖励是在初始状态采取动作后获得的奖励；

未来奖励是指离开初始状态后获得的奖励。

有可能即时奖励是负的，而未来奖励是正的。因此，采取哪些动作应由回报（即总奖励）决定，而不是由即时奖励决定，以避免短视的决策。

(1.1) 中的回报是为有限长度的轨迹定义的。回报也可以为无限长的轨迹定义。例如，图 1.6 中的轨迹在到达 $s_9$ 后停止。由于策略在 $s_9$ 定义良好，过程不必在智能体到达 $s_9$ 后停止。我们可以设计一个策略，使智能体在到达 $s_9$ 后保持不动。那么，该策略将生成以下无限长的轨迹：

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9 \xrightarrow[r=1]{a_5} s_9 \xrightarrow[r=1]{a_5} s_9 \ldots$$

沿着这条轨迹的奖励的直接求和是

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \cdots = \infty,$$

这不幸发散了。因此，我们需要为无限长的轨迹引入 **折扣回报（discounted return）** 概念。具体来说，折扣回报是折扣奖励的总和：

$$\text{discounted return} = 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \ldots, \tag{1.3}$$

其中 $\gamma \in (0, 1)$ 被称为折扣率（discount rate）。当 $\gamma \in (0, 1)$ 时，(1.3) 的值可以计算如下

$$\text{discounted return} = \gamma^3(1 + \gamma + \gamma^2 + \ldots) = \gamma^3 \frac{1}{1-\gamma}.$$

引入折扣率之所以有用，主要有以下原因。首先，它消除了**停止准则（stop criterion）**，允许存在无限长的轨迹。其次，折扣率可用于调整对近期奖励或远期未来奖励的重视程度。具体来说，如果 $\gamma$ 接近 0，那么智能体将更重视在不久的将来获得的奖励。由此产生的策略将是***短视的（short-sighted）***。如果 $\gamma$ 接近 1，那么智能体将更重视远期的未来奖励。由此产生的策略是***远视的（far-sighted）***，并且敢于为了在未来获得更多奖励而承担在近期获得负奖励的风险。这些观点将在 3.5 节中进行论证。

上述讨论中没有明确提到的一个重要概念是**回合（episode）**。当通过遵循策略与环境交互时，智能体可能会在某些终止状态（terminal states）停止。**由此产生的轨迹称为一个回合（episode）（或一次试验 trial）**。如果环境或策略是随机的，当从同一状态开始时，我们会获得不同的**episode**。然而，如果一切都是确定性的，当从同一状态开始时，我们将始终获得相同的**episode**。

一个**episode**通常被假设为一条有限的轨迹。具有**episode**的任务称为**episodic***任务（episodic tasks）*。然而，有些任务可能没有终止状态，这意味着与环境交互的过程永远不会结束。此类任务称为***连续性任务（continuing tasks）***。事实上，我们可以通过将**episodic**任务转换为连续性任务，以统一的数学方式处理回合制任务和连续性任务。为此，我们需要明确定义智能体到达终止状态后的过程。具体而言，在回合制任务中到达终止状态后，智能体可以通过以下两种方式继续采取行动。

- 第一，如果我们把终止状态视为一种特殊状态，我们可以专门设计其动作空间或状态转移，使智能体永远停留在该状态。这种状态称为**吸收态（absorbing states）**，意味着智能体一旦到达该状态就永远不会离开。例如，对于目标状态 $s_9$，我们可以指定 $\mathcal{A}(s_9) = \{a_5\}$ 或者设定 $\mathcal{A}(s_9) = \{a_1, \ldots, a_5\}$ 且对于所有 $i = 1, \ldots, 5$，都有 $p(s_9|s_9, a_i) = 1$。

- **第二，如果我们把终止状态视为普通状态**，我们可以简单地将其动作空间设置为与其他状态相同，**智能体可能会离开该状态并再次回来**。由于每次到达 $s_9$ 都可以获得 $r=1$ 的正奖励，**智能体最终将学会永远停留在 $s_9$ 以收集更多奖励**。值得注意的是，**当回合无限长且因停留在 $s_9$ 而获得的奖励为正时，必须使用折扣率来计算折扣回报以避免发散。**

在本书中，我们考虑第二种情况，即目标状态被视为动作空间为 $\mathcal{A}(s_9)=\{a_1,\dots,a_5\}$ 的普通状态。

## 1.7 马尔可夫决策过程

本章的前几节通过示例阐述了强化学习中的一些基本概念。本节将在马尔可夫决策过程（Markov decision processes, MDPs）的框架下，以更形式化的方式介绍这些概念。

MDP 是描述随机动力系统的一般框架。MDP 的关键要素列举如下。

- **集合（Sets）：**
  - 状态空间：所有状态的集合，记为 $\mathcal{S}$。
  - 动作空间：一组动作，记为 $\mathcal{A}(s)$，与每个状态 $s\in\mathcal{S}$ 相关联。
  - 奖励集：一组奖励，记为 $\mathcal{R}(s,a)$，与每个状态-动作对 $(s,a)$ 相关联。
- **模型（Model）：**
  - 状态转移概率：在状态 $s$ 下，当采取动作 $a$ 时，转移到状态 $s'$ 的概率为 $p(s'|s,a)$。对于任意 $(s,a)$，满足 $\sum_{s'\in\mathcal{S}}p(s'|s,a)=1$。
  - 奖励概率：在状态 $s$ 下，当采取动作 $a$ 时，获得奖励 $r$ 的概率为 $p(r|s,a)$。对于任意 $(s,a)$，满足 $\sum_{r\in\mathcal{R}(s,a)}p(r|s,a)=1$。
- **策略（Policy）：**
  - 在状态 $s$ 下，选择动作 $a$ 的概率为 $\pi(a|s)$。对于任意 $s\in\mathcal{S}$，满足 $\sum_{a\in\mathcal{A}(s)}\pi(a|s)=1$。
- **马尔可夫性质（Markov property）：**
  - 马尔可夫性质是指随机过程的无记忆性（memoryless property）。从数学上讲，它的意思是

$$
\begin{aligned}
p(s_{t+1}|s_t,a_t,s_{t-1},a_{t-1},\dots,s_0,a_0)&=p(s_{t+1}|s_t,a_t),\\
p(r_{t+1}|s_t,a_t,s_{t-1},a_{t-1},\dots,s_0,a_0)&=p(r_{t+1}|s_t,a_t),
\end{aligned}
\tag{1.4}
$$

  - 其中 $t$ 代表当前时间步，$t+1$ 代表下一个时间步。方程 (1.4) 表明，下一个状态或奖励仅取决于当前的状态和动作，而与之前的状态和动作无关。马尔可夫性质对于推导 MDP 的基本贝尔曼方程（Bellman equation）非常重要，如下一章所示。

在这里，所有 $(s,a)$ 的 $p(s'|s,a)$ 和 $p(r|s,a)$ 被称为**模型（model）**或 **动力学（dynamics）**。模型可以是*平稳的（stationary）*或*非平稳的（nonstationary）*（换句话说，时不变的或时变的）。平稳模型不会随时间变化；非平稳模型可能会随时间变化。例如，在网格世界示例中，如果一个禁入区域有时突然出现或消失，那么模型就是非平稳的。在本书中，我们只考虑平稳模型。

---

## 1.8 小结

大家可能听说过马尔可夫过程（Markov processes, MPs）。MDP 和 MP 之间有什么区别？**答案是，一旦 MDP 中的策略固定下来，MDP 就退化为 MP**。例如，图 1.7 中的网格世界示例可以抽象为一个马尔可夫过程。在随机过程的文献中，如果马尔可夫过程是一个离散时间过程且状态数量有限或可数，它也被称为马尔可夫链（Markov chain）。在本书中，当上下文清晰时，术语“马尔可夫过程”和“马尔可夫链”可以互换使用。此外，本书主要考虑*有限 MDP（finite MDPs）*，即状态和动作的数量是有限的。这是应该被充分理解的最简单的情况。

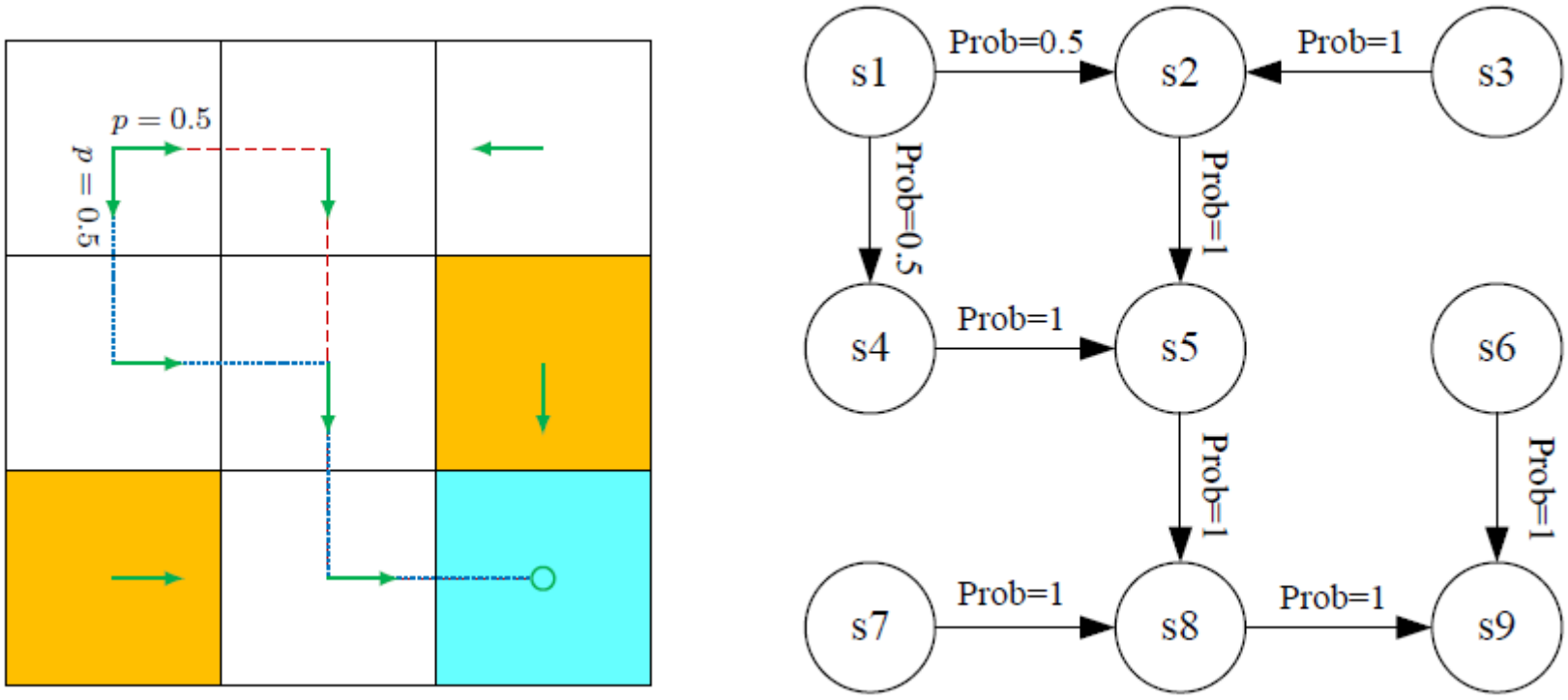



**图 1.7：网格世界示例抽象为马尔可夫过程。这里，圆圈代表状态，带箭头的连线代表状态转移。**

最后，强化学习可以描述为一个智能体与环境交互的过程。**智能体（agent）***是一个决策者，***它可以感知自身的状态、维护策略并执行动作。智能体之外的一切都被视为环境（environment）**。在网格世界示例中，智能体和环境分别对应于机器人和网格世界。在智能体决定采取一个动作后，执行器会执行该决策。然后，智能体的状态会发生改变，并获得一个奖励。通过使用解释器，智能体可以解释新的状态和奖励。从而形成一个闭环。

## 1.9 问答

- 问：我们可以将所有奖励都设置为负数或正数吗？
- 答： 在本章中，我们提到正奖励会鼓励智能体采取某个动作，而负奖励会阻止智能体采取该动作。事实上，决定鼓励还是抑制的是**相对奖励值（relative reward values）**，而不是**绝对值（absolute values）**。

更具体地说，本章中我们设定了 $r_{\text{boundary}} = -1, r_{\text{forbidden}} = -1, r_{\text{target}} = +1$ 以及 $r_{\text{other}} = 0$。我们也可以在不改变最终最优策略的情况下，给所有这些值加上一个公共数值。例如，我们可以给所有奖励加上 $-2$，从而得到 $r_{\text{boundary}} = -3, r_{\text{forbidden}} = -3, r_{\text{target}} = -1$ 以及 $r_{\text{other}} = -2$。**虽然所有奖励都是负的，但最终的最优策略保持不变。这是因为最优策略对奖励的仿射变换（affine transformations）具有不变性。**详情将在 3.5 节中给出。

- **问：奖励是下一个状态的函数吗？**
- **答：** 我们提到过，奖励 $r$ 仅取决于 $s$ 和 $a$，而不取决于下一个状态 $s'$。然而，这可能有些反直觉，因为在许多情况下，正是下一个状态决定了奖励。例如，当下一个状态是目标状态时，奖励为正。因此，一个自然而然的问题是，奖励是否应该取决于下一个状态。这个问题的数学重述是，我们是否应该使用 $p(r|s,a,s')$（其中 $s'$ 是下一个状态）而不是 $p(r|s,a)$。答案是，$r$ 确实取决于 $s, a$ 和 $s'$。然而，由于 $s'$ 也取决于 $s$ 和 $a$，我们可以等价地将 $r$ 写成 $s$ 和 $a$ 的函数：
  - $$p(r|s,a) = \sum_{s'} p(r|s,a,s')p(s'|s,a)$$
- 这样，就可以很容易地建立第 2 章所示的贝尔曼方程（Bellman equation）。