

# 第一章 强化学习中的基本概念

本章介绍了强化学习（**Reinforcement Learning**）的基本概念。这些概念很重要，因为它们将在本书中广泛使用。我们首先用例子介绍这些概念，然后在马尔可夫决策过程的框架中对它们进行形式化。

## 1.1 网格世界实例

考虑一个如图1.2所示的例子，其中机器人在网格世界中移动。该机器人被称为智能体（**Agent**），可以在网格中的相邻单元之间移动。在每个时间步长，它只能占用一个单元格。白色格子可以进入，橙色格子被禁止进入。机器人想要到达一个目标格子。我们将在整本书中使用这样的网格世界示例，因为它们直观地说明了新的概念和算法。

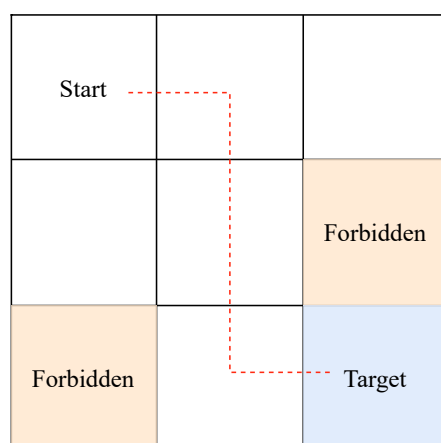


图1.2: 网格世界的例子在整本书中都有使用。

智能体的最终目标是找到一个“好”的策略（**Policy**），使其能够在从任何初始单元格开始时到达目标单元格。如何定义策略的“好”？这个想法是，智能体应该在不进入任何禁区、不走不必要的弯路或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，那么去规划到达目标格子的路径将是不重要的。如果智能体事先不知道有关环境的任何信息，那么任务就变得重要了。然后，智能体必须与环境交互，通过反复尝试找到一个好的策略。要做到这一点，本章其余部分介绍的概念是必要的。

## 1.2 状态和动作

要引入的第一个概念是状态（**State**），它描述了智能体相对于环境的状态。在网格世界示例中，状态对应于智能体的位置。因为有九个格子，所以也有九个状态。它们被索引为  $s_1, s_2, s_9$ ，如图1.3 (a) 所示。所有状态的集合称为状态空间（**State Space**），表示为  $\mathcal{S} = \{s_1, \dots, s_9\}$ 。

对于每种状态，智能体可以采取五种可能的动作（**Action**）：向上移动、向右移动、向下移动、向左移动和保持不变。这五个动作被表示为  $a_1, a_2, \dots, a_5$ （见图1.3（b））。所有动作（**Action**）的集合称为动作空间（**Action Space**），表示为  $\mathcal{A} = \{a_1, \dots, a_5\}$ 。不同的状态可以有不同的动作空间。例如考虑到在状态  $s_1$  取  $a_1$  或  $a_4$  将导致与边界的碰撞，我们可以将状态  $s_1$  的动作空间设置为  $\mathcal{A}(s_1) = \{a_2, a_3, a_5\}$ 。

在这本书中，我们考虑了最普遍的情况：对于所有  $i$ ， $\mathcal{A}(s_i) = \mathcal{A} = \{a_1, \dots, a_5\}$ 。

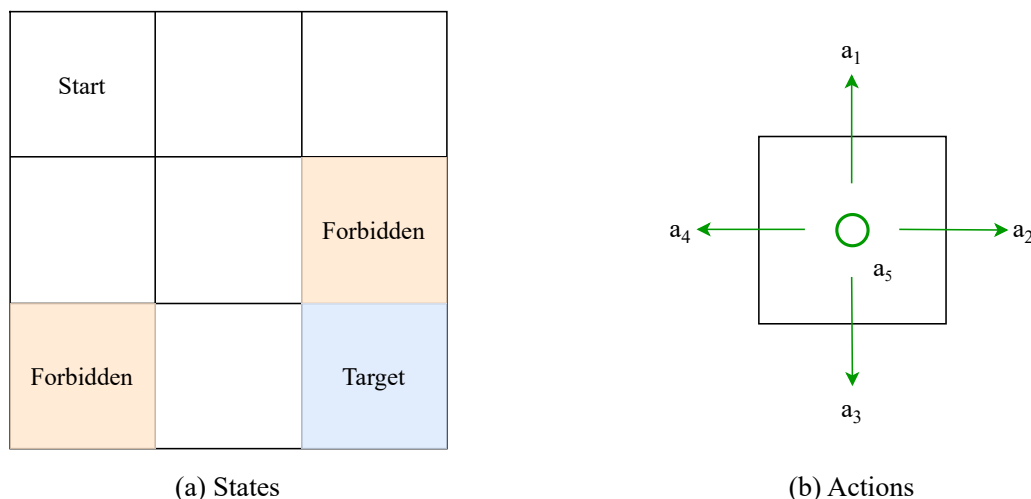


图1.3

## 1.3 状态转换

在执行动作时，智能体可能会从一种状态移动到另一种状态。这种过程被称为**状态转换（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），则智能体移动到状态  $s_2$ 。这样的过程可以表示为： $s_1 \xrightarrow{a_2} s_2$ 。

接下来我们将研究两个重要的例子。

- 当智能体试图超越边界时，下一个状态是什么，例如，在状态  $s_1$  采取动作  $a_1$ ？答案是，智能体将会被反弹，因为智能体不可能退出状态空间。因此，状态转换过程为  $s_1 \xrightarrow{a_1} s_1$ 。
- 当智能体试图进入被禁止的格子里时，下一个状态是什么，例如，在状态  $s_5$  采取动作  $a_2$ ？可能会遇到两种不同的情况。在第一个场景中，尽管  $s_6$  被禁止，但它仍然可以访问。在这种情况下，下一个状态是  $s_6$ ；因此，状态转换过程为  $s_5 \xrightarrow{a_2} s_6$ 。在第二种情况下， $s_6$  是不可访问的，因为它被墙包围了。在这种情况下，如果智能体试图向右移动，则它会被弹回到  $s_5$ ；因此，状态转换过程是  $s_5 \xrightarrow{a_2} s_5$ 。

我们应该考虑哪种情况？答案取决于**物理环境（Physical Environment）**。在这本书中，我们考虑了第一种情况，即被禁止的格子是可以进入的，但是踏入它们可能会受到惩罚。这种情况更为普遍和有趣。此外，由于我们正在考虑模拟任务，我们可以根据自己的喜好定义状态转换过程。在真实世界的应用中，状态转换过程由真实世界的动力学决定。

为每个状态及其关联的动作定义了状态转换过程。这个过程可以用表1.1中所示的表格来描述。在这个表中，每一行对应一个状态，每一列对应一个动作。每个格子表示智能体在相应状态下采取相应动作后要转换到的下一个状态。

	$a_1$ (向上)	$a_2$ (向右)	$a_3$ (向下)	$a_4$ (向左)	$a_6$ (不动)
$s_1$	$s_1$	$s_2$	$s_4$	$s_1$	$s_1$
$s_2$	$s_2$	$s_3$	$s_5$	$s_1$	$s_2$
$s_3$	$s_3$	$s_3$	$s_6$	$s_2$	$s_3$
$s_4$	$s_1$	$s_5$	$s_4$	$s_4$	$s_4$
$s_5$	$s_2$	$s_6$	$s_5$	$s_4$	$s_5$
$s_6$	$s_3$	$s_6$	$s_6$	$s_5$	$s_6$
$s_7$	$s_4$	$s_8$	$s_7$	$s_7$	$s_7$
$s_8$	$s_5$	$s_9$	$s_8$	$s_7$	$s_8$
$s_9$	$s_6$	$s_9$	$s_9$	$s_8$	$s_9$

表1.1：状态转换过程的表格形式。每个单元格表示智能体在某个状态下执行动作后要转换到的下一个状态。

从数学的角度讲，状态转换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：

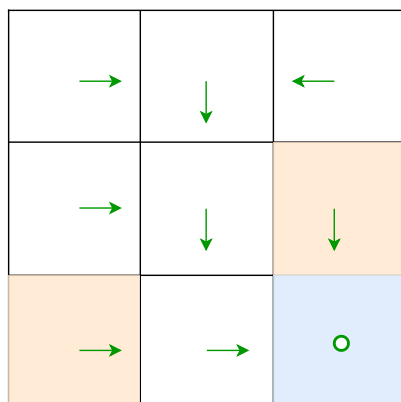
$$\begin{aligned}
 p(s_1 \mid s_1, a_2) &= 0, \\
 p(s_2 \mid s_1, a_2) &= 1, \\
 p(s_3 \mid s_1, a_2) &= 0, \\
 p(s_4 \mid s_1, a_2) &= 0, \\
 p(s_5 \mid s_1, a_2) &= 0,
 \end{aligned}$$

这表明，当在状态 $s_1$ 取动作 $a_2$ 时，智能体移动到 $s_2$ 的概率为1，而移动到其他状态的概率为0。因此，在状态 $s_1$ 采取动作 $a_2$ 肯定会导致智能体发生状态转换到 $s_2$ 。

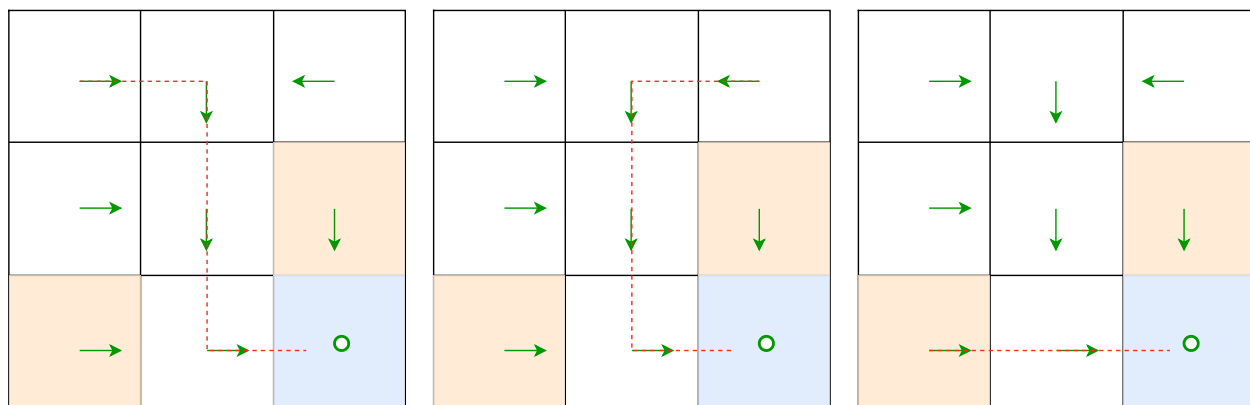
尽管它是直观的，但表格表示只能描述**确定性状态转换（Deterministic State Transitions）**。一般来说，状态转换可以是**随机的（Stochastic）**，并且必须用**条件概率分布（Conditional Probability Distributions）**来描述。例如，当在网格上施加随机阵风时，如果在 $s_1$ 处采取动作 $a_2$ ，则可能将智能体吹到 $s_5$ 而不是 $s_2$ 。在这种情况下，我们有 $p(s_5 \mid s_1, a_2) > 0$ 。然而，为了简单起见，我们在本书中只考虑了网格世界中S的确定性状态转换示例。

## 1.4 策略

策略（**Policy**）告诉智能体在每个状态下要采取的动作是什么。直观地说，策略可以被描述为箭头（见图1.4（a））。在策略之后，智能体可以生成从初始状态开始的轨迹（参见图1.4（b））。



(a) A deterministic policy



(b) Trajectories obtained from the policy

图1.4：由箭头和从不同初始状态开始获得的一些轨迹表示的策略。

从数学上讲，策略可以用条件概率来描述。将图1.4中的策略表示为  $\pi(a \mid s)$ ，这是为每个状态定义的条件概率分布函数。例如， $s_1$  的策略为：

$$\pi(a_1 \mid s_1) = 0$$

$$\pi(a_2 \mid s_1) = 1$$

$$\pi(a_3 \mid s_1) = 0$$

$$\pi(a_4 \mid s_1) = 0$$

$$\pi(a_5 \mid s_1) = 0$$

这指示在状态  $s_1$  采取动作  $a_2$  的概率是 1，并且采取其他动作的概率是 0。

上述策略具有确定性。一般来说，策略可能是随机的。例如，图1.5中所示的策略是随机的：在状态  $s_1$ ，智能体可能会采取向右或向下的行动。采取这两个动作的概率相同（均为 0.5）。在这种情况下， $s_1$  的策略为：

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

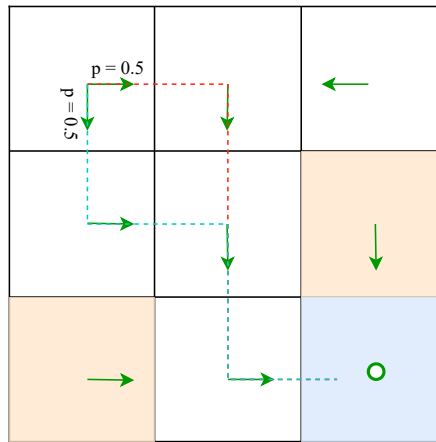


图1.5：随机策略。在状态 $s_1$ ，智能体可以以0.5的相等概率向右或向下移动。

由条件概率表示的策略可以存储为表。例如，表1.2代表了图1.5中描述的随机策略。第  $i$  行和第  $j$  列中的条目（**Entry**）是表示在第  $i$  个状态下采取第  $j$  个动作的概率。这种表示称为表格表示（**Tabular Representation**）。我们将在第 8 章中介绍另一种将策略表示为参数化函数（**Parameterized Functions**）的方法。

	$a_1$ (向上)	$a_2$ (向右)	$a_3$ (向下)	$a_4$ (向左)	$a_6$ (不动)
$s_1$	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

表1.2：策略的表格表示。每个条目指的是在某个状态下采取相应动作的概率。

## 1.5 奖励

奖励（**Reward**）是强化学习中最独特的概念之一。在某个状态下执行动作后，智能体从环境中获得奖励作为反馈，奖励表示为  $r$ 。奖励是状态  $s$  和动作  $a$  的函数。因此，它也被表示为  $r(s, a)$ 。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学习的策略有不同的影响。一般来说，奖励为正，代表我们鼓励智能体所采取的动作。如果奖励为负，代表我们会阻止智能体所采取的行动。

在网格世界示例中，奖励设计如下：

- 如果智能体试图退出边界，则让  $r_{boundary} = -1$ 。
- 如果智能体试图进入一个被禁止的单元格，则让  $r_{forbidden} = -1$ 。
- 如果智能体达到目标状态，则让  $r_{target} = +1$ 。
- 否则，智能体将获得  $r_{other} = 0$  的奖励。

应特别注意：目标状态  $s_9$ 。在智能体到达  $s_9$  之后，奖励过程不必终止。如果代理在  $s_9$  采取动作  $a_5$ ，则下一状态再次是  $s_9$ ，并且奖励是  $r_{target} = +1$ 。如果代理采取行动  $a_2$ ，则下一个状态也是  $s_9$ ，但奖励是  $r_{boundary} = -1$ 。

奖励可以被解释为一个人机界面（**Human-machine Interface**），通过它，我们可以引导智能体按照我们的预期行事。例如，通过上面设计的奖励，我们可以预期智能体倾向于避免离开边界或避免进入禁区。设计适当的奖励是强化学习的重要一步。然而，这一步骤对于复杂的任务来说是非平凡的（**Nontrivial**），因为它可能需要用户很好地理解给定的问题。尽管如此，它可能仍然比用其他需要专业背景或对给定问题有深入理解的方法来解决问题容易得多。

执行动作后获得奖励的过程可以直观地表示为表格，如表1.3所示。表中的每一行对应一个状态，每一列对应一个动作。表中每个单元格中的值表示在一个状态下通过采取相应的动作可以获得的奖励。

	$a_1$ (向上)	$a_2$ (向右)	$a_3$ (向下)	$a_4$ (向左)	$a_6$ (不动)
$s_1$	$r_{boundary}$	0	0	$r_{boundary}$	0
$s_2$	$r_{boundary}$	0	0	0	0
$s_3$	$r_{boundary}$	$r_{boundary}$	$r_{forbidden}$	0	0
$s_4$	0	0	$r_{forbidden}$	$r_{boundary}$	0
$s_5$	0	$r_{forbidden}$	0	0	0
$s_6$	0	$r_{boundary}$	$r_{target}$	0	$r_{forbidden}$
$s_7$	0	0	$r_{boundary}$	$r_{boundary}$	$r_{forbidden}$
$s_8$	0	$r_{target}$	$r_{boundary}$	$r_{forbidden}$	0
$s_9$	$r_{forbidden}$	$r_{boundary}$	$r_{boundary}$	0	$r_{target}$

表1.3: 获得奖励过程的表格表示。在这里这个过程是确定的。每个单元格表示在给定状态下智能体采取相应的动作后可以获得多少奖励。

初学者可能会问的一个问题如下：如果给出奖励表，我们可以通过简单地选择奖励最大的动作来找到好的策略吗？答案是否定的。这是因为这些奖励是在采取动作后可以获得的即时奖励。为了确定一个好的策略，我们必须考虑长远来的看获得的总奖励（更多信息，请参阅第1.6节）。即时奖励最大的行动可能不会带来最大的总奖励。

尽管直观，但表格表示只能描述确定性奖励过程。一种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

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

这表明，当在状态  $s_1$  取  $a_1$  时，智能体确定地获得  $r = -1$  的奖励。在这个例子中，奖励过程是确定性的。一般来说，它可以是随机的。例如，如果学生努力学习，他或她将获得积极的奖励（例如，考试成绩更高），但奖励的具体价值可能并不确定。

## 1.6 轨迹、回报和回合

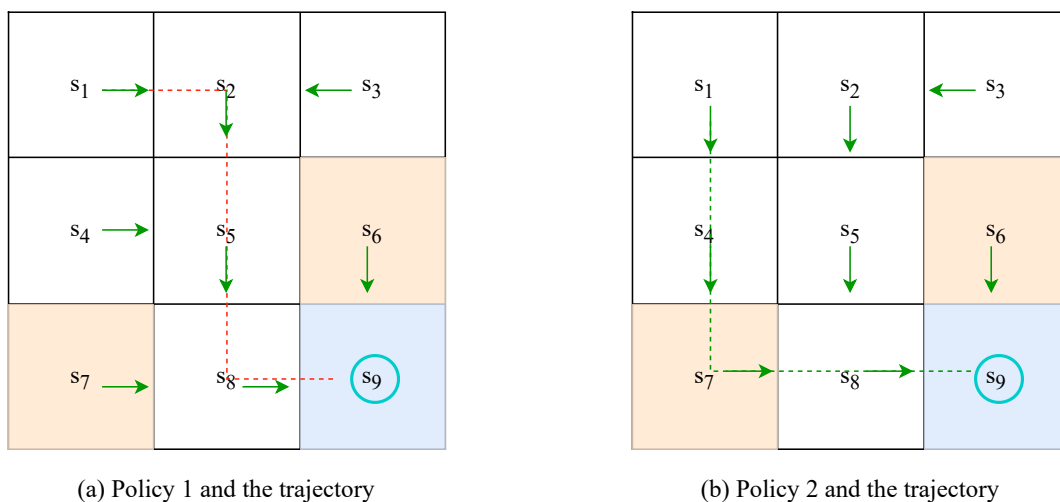


图1.6: 通过以下两种策略获得的轨迹。轨迹由红色虚线表示。

轨迹（**Trajectory**）是一个状态-动作-奖励链（**State-Action-Reward Chain**）。例如，给定图1.6 (a) 中所示的策略，如果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**）被定义为沿着该轨迹收集的所有奖励的总和：

$$\mathbf{return} = 0 + 0 + 0 + 1 = 1 \quad (1.1)$$

回报（**Return**）也称为总奖励（**Total Rewards**）或累计奖励（**Cumulative Rewards**）。



回报可用于策略评估。例如，我们可以通过比较图1.6中的两个策略的回报来评估它们。特别地，从  $s_1$  开始，左侧的策略获得的回报是如上计算的1。对于右侧的策略，从  $s_1$  开始，生成以下轨迹：

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

相应的回报是：

$$\mathbf{return} = 0 - 1 + 0 + 1 = 0 \quad (1.2)$$

(1.1)和(1.2)中的回报表明，左侧策略比右侧策略好，因为它的回报更大。这一数学结论与直觉一致，即右侧策略更糟糕，因为它通过了一个禁区。

回报包括即时奖励（**Immediate Reward**）和未来奖励（**Future Rewards**）。这里，即时奖励是在初始状态下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。有可能眼前的回报是负的，而未来的回报是正的。因此，应该根据回报（即总奖励）而不是即时奖励来决定采取哪些动作，以避免短视的决定。

(1.1)中的回报是有限长度的轨迹定义的。回报也可以定义为无限长的轨迹。例如，图1.6中的轨迹在到达  $s_9$  后停止。由于策略是为  $s_9$  定义的，所以在智能体到达  $s_9$  后，过程不必停止。我们可以设计一个策略，使智能体在到达  $s_9$  后保持不变。然后，该策略将产生以下无限长的轨迹：

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

沿着这条轨迹的奖励的直接总和是：

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

不幸的是出现了分歧。因此，我们必须介绍无限长轨迹的折扣回报（**Discounted Return**）概念。特别是，折扣回报是折扣奖励的总和：

$$\mathbf{discounted return} = 0 + \gamma \times 0 + \gamma^2 \times 0 + \gamma^3 \times 1 + \gamma^4 \times 1 + \gamma^5 \times 1 + \cdots, \quad (1.3)$$

其中  $\gamma \in (0, 1)$  称为折扣率（**Discount Rate**），当  $\gamma \in (0, 1)$  时，(1.3)的值可以计算为：

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

引入折扣率是有用的，原因如下。首先，它取消了如何停止的标准，并允许存在无限长的轨迹。其次，折扣率可以用来调整近期奖励或远期奖励的重视程度。特别是，如果  $\gamma$  接近 0，那么智能体会更加重视在近期获得的奖励。由此产生的策略是短视的。如果  $\gamma$  接近 1，那么智能体会更加强调未来的奖励。由此产生的策略目光远大，敢于面对将来获得负面奖励的情况。这些要点将在第3.5节中进行说明。



上述讨论中没有明确提到的一个重要概念是回合（**Episode**）。当通过策略与环境交互时，智能体可能会在某些终态（**Terminal States**）下停止。由此产生的轨迹被称为一个回合（**Episode**）（或尝试（**Trial**））。如果环境或策略是随机的，当从同一状态开始时，我们会获得不同的回合。然而，如果一切都是确定性的，那么当从相同的状态开始时，我们总是获得相同的回合。

一个回合通常被认为是一个有限的轨迹。带回合的任务被称为回合制任务（**Episodic Tasks**）。然而，有些任务可能没有终态，这意味着与环境交互的过程永远不会结束。这种任务称为连续任务（**Continuing Tasks**）。事实上，我们可以通过统一的数学方式将回合制任务处理为连续任务。要做到这一点，我们需要很好地定义智能体到达终态后的过程。具体来说，在情景任务中达到终态后，智能体可以通过以下两种方式继续采取行动。

- 首先，如果我们将终端状态视为一种特殊状态，我们可以专门设计它的动作空间或状态转换，使智能体永远处于这种状态。这种状态被称为吸收态（**Absorbing States**），这意味着智能体一旦到达一个状态就永远不会离开。例如。对于目标状态  $s_g$ ，我们可以指定  $\mathcal{A}(s_g) = \{a_5\}$ ，或者对所有的  $i = 1, 2, \dots, 5$ ， $\mathcal{A}(s_g) = \{a_1, a_2, \dots, a_5\}$ ，其中  $p(s_g | s_g, a_i) = 1$ 。
- 第二，如果我们将终态视为正常状态，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返回。由于每次达到  $s_g$  都可以获得  $r=1$  的正奖励，因此智能体最终将学会永远留在  $s_g$  以收集更多奖励。值得注意的是，当一回合是无限长的，且停留在  $s_g$  的奖励是正的时，必须使用折扣率来计算折扣回报，以避免差异。

在这本书中，我们考虑第二种情况，其中目标状态被视为动作空间为  $\mathcal{A}(s_g) = \{a_1, \dots, a_5\}$  的正常状态。

## 1.7 马尔可夫决策过程

本章的前几节通过例子说明了强化学习中的一些基本概念。本节在马尔可夫决策过程（**Markov Decision Processes, MDPs**）的框架下以更正式的方式介绍了这些概念。

**MDP** 是描述随机动力系统的通用框架。**MDP** 的关键成分如下所示。

- 元素：
  - 状态空间：所有状态的集合，表示为  $\mathcal{S}$ 。
  - 动作空间：一组动作，表示为  $\mathcal{A}(s), s \in \mathcal{S}$ 。
  - 奖励集：一组奖励，表示为  $\mathcal{R}(s, a)$ ，与每个状态动作对  $(s, a)$  相关。
- 模型：
  - 状态转换概率：在状态  $s$ ，当采取动作  $a$  时，转换到状态  $s'$  的概率为  $p(s' | s, a)$ 。

- 奖励概率：在状态  $s$ ，当采取动作  $a$  时，获得奖励  $r$  的概率为  $p(r | s, a)$ 。  
他认为：  $\sum_{r \in \mathcal{R}(s,a)} p(r | s, a) = 1, \forall (s,a)$
- 策略：在状态  $s$ ，选择动作  $a$  的概率为  $\pi(a | s)$ 。他认为：  
 $\sum_{a \in \mathcal{A}(s)} \pi(a | s) = 1, \forall s \in \mathcal{S}$
- 马尔可夫性质：马尔可夫性质（**Markov Property**）是指随机过程的无记忆性质。从数学上讲，这意味着

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

其中  $t$  表示当前时间戳， $t+1$  表示下一时间戳。等式(1.4)表明，下一个状态或奖励仅取决于当前状态和动作，并且与之前的状态和动作无关。马尔可夫性质对于推导 MDP 的基本贝尔曼方程很重要，如下一章所示。

这里，所有  $(s, a)$  的  $p(s' | s, a)$  和  $p(r | s, a)$ ，这个叫做模型（**Model**）。该模型可以是平稳的，也可以是非平稳的（或者换句话说，是时不变的或时变的）。平稳模型不会随时间变化；非平稳模型可能随时间变化。例如，在网格世界的例子中，如果禁区有时会弹出或消失，则模型是非平稳的。在这本书中，我们只考虑平稳模型。

人们可能听说过马尔可夫过程。MDP 和 MP 之间的区别是什么？答案是，一旦 MDP 中的策略被固定，MDP 就会退化为 MP。例如，图1.7中的网格世界示例可以抽象为马尔可夫过程。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，并且状态的数量是有限的或可计数的，那么它也被称为马尔可夫链<sup>1</sup>。在这本书中，当上下文清楚时，术语“马尔可夫过程”和“马尔可夫链”可以互换使用。此外，本书主要考虑有限的 MDP，其中状态和动作的数量是有限的。这是应该完全理解的最简单的情况。

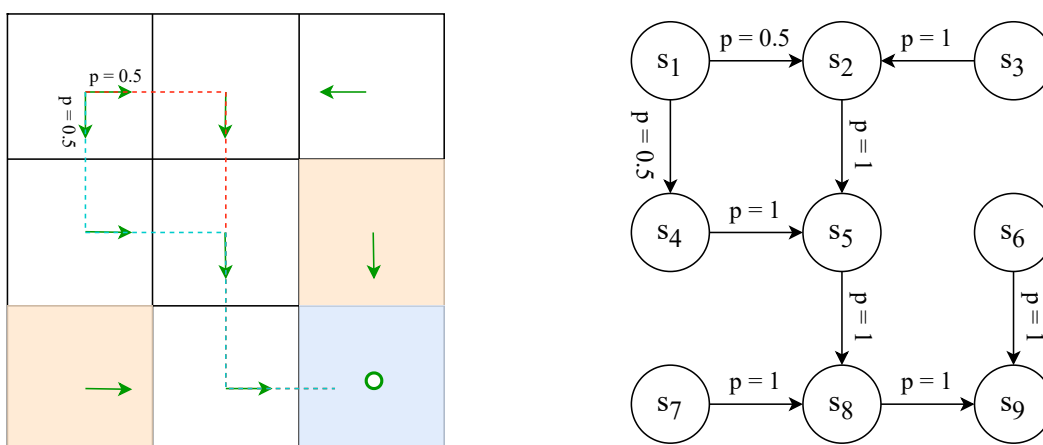


图1.7：作为马尔可夫过程的网格世界示例的抽象。这里，圆圈表示状态，带箭头的链接表示状态转换。

最后，强化学习可以被描述为一个智能体-环境交互过程。智能体是一个决策者，可以感知其状态、维护策略并执行动作。智能体之外的一切都被视为环境。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。在智能体决定采取动作之后，致动器执行这样的决定。然后，智能体的状态将被改变，并且可以获得奖励。通过使用解释器，代理可以解释新状态和奖励。因此，可以形成闭环。

## 1.8 总结

本章介绍了将在本书剩余部分广泛使用的基本概念。我们使用直观的网络世界示例来演示这些概念，然后在 MDP 框架中对它们进行形式化。有关 MDP 的更多信息，读者可以参见 [1 2](#)。

---

1. M. Pinsky and S. Karlin, An introduction to stochastic modeling (3rd Edition). [↩](#) [↩](#)

2. M. L. Puterman, Markov decision processes: Discrete stochastic dynamic programming. John Wiley & Sons, 2014. [↩](#)