第5章　模 态 逻 辑

我们经常需要推理事物的存在状态，以帮助我们做出决策。由于推理过程中常涉及不确定性，需要使用“必然”和“可能”等概念来表达不同的情况。模态逻辑 (Modal Logic) 提供了一种形式化语言，使我们能够对自然语言中的这些概念进行精确定义和严格推理。本章将首先介绍基本的模态逻辑、语言和语义，引入一系列重要的概念、形式推演系统和判定问题的一些经典结果。作为模态逻辑的扩展，结语的部分还将介绍时态逻辑和动态逻辑。

5.1　自动驾驶中的模态逻辑推理

假设一辆自动驾驶汽车正行驶在高速公路上，系统需要判断是否可以安全继续行驶。定义以下命题：

- p：前方道路畅通；
- o：前方有障碍物；
- s：传感器数据可靠 (即所有传感器都正常工作)；
- d：车辆可以安全行驶。

引入模态逻辑的两个基本算子：

- $\Box p$：“必然 p”，即所有情况下道路都畅通；
- $\Diamond p$：“可能 p”，即在至少一种情况下道路畅通。

推理规则　假设以下逻辑规则成立：

$\Box p \to d$　(如果前方必然畅通，则车辆可以安全行驶)　(a)

$\Box o \to \neg d$　(如果前方必然有障碍物，则车辆不能安全行驶)　(b)

$\Diamond p \wedge \Diamond o$　(在某些情况下前方畅通，而在某些情况下有障碍物，存在不确定性)　(c)

$\Box s \to (\Box p \vee \Box o)$　(如果传感器必然可靠，则可以判断前方是否畅通)　(d)

以下考虑不同情境下的推理。

理想情况，所有传感器正常工作：由于 $\Box s$ 为真 (所有传感器都可靠)，系统根据传感器信息得到 $\Box p$。由推理规则 (a) $\Box p \to d$ 可得 d，而且这个结论是必然的，即 $\Box d$。结论：

车辆必然可以安全行驶。

部分传感器失效，存在不确定性：例如，前方有雾，激光雷达无法检测远处的情况，但摄像头仍能部分探测。这意味着 $\neg\Box s$ (传感器数据不必然可靠)。由于部分传感器仍可工作，系统只能得出 $\Diamond p$ (道路可能畅通)。但同时，系统可能也会得出 $\Diamond o$ (道路可能有障碍物)。由于 $\Diamond p \wedge \Diamond o$，系统无法确定 $\Box d$ (无法确保必然安全行驶)。这时，需要调整驾驶策略：车辆应减速或依靠额外数据 (如云端地图、V2X 车联网信息) 进一步做决策。

前方有障碍物，但不确定是否可移动：假设汽车检测到前方可能有一辆静止的车，但无法判断它是否会移动。这可以表示为

$$\Diamond o \wedge \neg\Box o \quad \text{(可能有障碍物，但不必然有障碍物)} \tag{e}$$

进一步推理：如果 $\Box o$ (必然有障碍物)，则根据 (b) 得出 $\neg d$，即不能安全行驶。如果 $\Diamond\neg o$ (障碍物可能会移开)，则仍有 $\Diamond d$，即可能可以继续行驶。这时，需要调整驾驶策略：系统可以选择短暂停止，或尝试换道，而不是直接继续前行。

如上所示，这种模态逻辑推理方法可用于对不确定性进行推理：如在高速公路驾驶过程中，当传感器不确定前方情况时，系统可以动态调整决策 (如减速或变道)。或者在恶劣天气条件下，部分传感器失效，可帮助系统判断是否需要切换驾驶模式。如果车辆的传感器无法确定前方是否畅通，系统可以通过联网获取其他车辆或基础设施的信息，弥补感知信息的不确定性。

这个例子展示了模态语言在自动驾驶情境中如何用于表达不确定性并进行推理。事实上，围绕可能性和必然性的推理在日常生活中也非常普遍。

5.2 模态语言和可能世界语义

下面给出模态逻辑的语言。

定义 35 (模态逻辑的语言). *令 P 是原子命题的集合，p 表示任意的原子命题，模态逻辑的语言 L 由下面的规则定义：*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\varphi \leftrightarrow \varphi) \mid \Box\varphi \mid \Diamond\varphi$$

可以看出，在命题逻辑语言的基础上添加了两个模态算子。$\Box\varphi$ 读作“必然 φ”或者“φ 必然为真”。同样，$\Diamond\varphi$ 读作“可能 φ”或者“φ 可能为真”。事实上，这两个模态算子是可以互相定义的：$\Box\varphi \Leftrightarrow \neg\Diamond\neg\varphi$。

练习 31. *用模态逻辑的语言表示下面的句子。*

(1) *如果一个人工智能系统缺乏适当的训练数据，它就不可能做出准确的预测。*

(2) *为了使人工智能高效运行，拥有大量数据是必要的。*

(3) 如果一个人工智能存在偏见，它的决策不一定是公平的。

(4) 未来人工智能生成的内容可能会主导网络媒体。

(5) 如果计算能力持续提升，人工智能有可能实现类人推理。

使用可能世界语义来对“必然”和“可能”的概念做出精确的解释。基于可能世界的语义模型又叫作克里普克 (Kripke) 模型。

定义 36 (模态逻辑的模型). 一个克里普克模型 $\mathfrak{M}$ 是一个三元组 (S, R, V)，其中，

(a) S 是可能世界 (或可能状态) 的集合；

(b) R 是可能世界之间的二元可及关系；

(c) V 是赋值函数: $P \to \mathcal{P}(S)$。其中，$\mathcal{P}(S)$ 是 S 的幂集，$V(p)$ 是 p 为真的所有世界的集合。

可能世界之间的可及关系可以有多种解释。正如将在后续章节中看到的，这种关系既可以体现为认知上的可及性 (认知逻辑)，也可以代表一种道义上更理想的状态 (道义逻辑)，也可以是执行了一个程序使得状态发生了变化 (动态逻辑)，从而为其他逻辑语言的解释奠定基础。赋值函数则告诉我们哪些命题在哪些世界上为真。

例 14. 图 5-1 表示一个克里普克模型，其中 $W = \{w_1, w_2, w_3, w_4\}$，在图中用节点表示。可及关系用节点之间带箭头的线表示，我们有 w_1Rw_2, w_1Rw_3, w_2Rw_4, w_3Rw_3, w_3Rw_4。最后，$V(p) = \{w_1, w_3, w_4\}$, $V(q) = \{w_2, w_3\}$。注意，图中 w_3 有一个自反箭头，表示对自身可及。

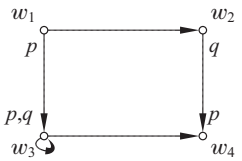


图 5-1　克里普克模型

当不考虑赋值函数时，就得到 $\mathfrak{F} = (S, R)$，称为框架 (Frame)。因为赋值函数可以有很多，所以基于同一个框架的模型也可能有很多。定义有效性概念的时候将会用到框架的概念。现在，一旦模型确定，就可以给出任意一个公式在模型的某个世界或状态上为真的条件。

定义 37 (真值条件). 给定一个模型 $\mathfrak{M}$ 和一个状态 $s \in S$，以下递归定义一个公式在 $\mathfrak{M}, s$ 上是真的：

$$\begin{aligned}
\mathfrak{M}, s \models p \quad & \text{当且仅当} \quad s \in V(p) \\
\mathfrak{M}, s \models \neg\varphi \quad & \text{当且仅当} \quad \text{并非} \mathfrak{M}, s \models \varphi \\
\mathfrak{M}, s \models \varphi \wedge \psi \quad & \text{当且仅当} \quad \mathfrak{M}, s \models \varphi \text{并且} \mathfrak{M}, s \models \psi \\
\mathfrak{M}, s \models \Box\varphi \quad & \text{当且仅当} \quad \text{所有的 } t, \text{若 } sRt, \text{则} \mathfrak{M}, t \models \varphi \\
\mathfrak{M}, s \models \Diamond\varphi \quad & \text{当且仅当} \quad \text{存在 } t, sRt \text{ 且} \mathfrak{M}, t \models \varphi
\end{aligned}$$

其余形式公式的真值条件，这里省略，读者容易补充。

练习 32. 考虑例 14 中的模型，判断下面公式分别在 4 个世界上的真假。

(1) $\Box p$

(2) $\Diamond q$

(3) $\Box\Box q$

(4) $\Box(p \to q)$

(5) $\Diamond q \to \Box p$

5.3 互模拟、有效性

给定两个模态逻辑的模型 $\mathfrak{M}$ 和 $\mathfrak{M}'$，若 $\mathfrak{M}$ 中的状态 s 和 $\mathfrak{M}'$ 中的状态 s' 满足相同的模态公式集，则它们是模态等价的，记作 $s \rightsquigarrow s'$。

与模态等价密切相关的一个概念是模型之间的“互模拟”(Bisimulation)。

定义 38 (互模拟). 设 $\mathfrak{M} = (S, R, V)$, $\mathfrak{M}' = (S', R', V')$ 为两个模型。称一个非空的二元关系 $Z \subseteq S \times S'$ 是两个模型之间的互模拟，(记作 Z: $\mathfrak{M} \underline{\leftrightarrow} \mathfrak{M}'$)，如果 Z 满足下面的条件：

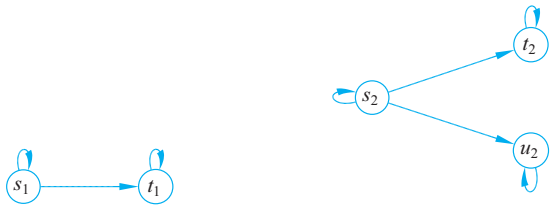
(1) 若 sZs'，则 s, s' 满足相同的命题变元；

(2) 如果 sZs' 并且 sRt，那么 $\mathfrak{M}'$ 中存在 t' 满足 $s'R't'$ 并且 tZt'；

(3) 如果 sZs' 并且 $s'R't'$，那么 $\mathfrak{M}$ 中存在 t 满足 sRt 并且 tZt'。

当 Z 是连接 $\mathfrak{M}$ 中的 s 和 $\mathfrak{M}'$ 中的 s' 互模拟时，可以说 s 和 s' 互模拟，记作 $\mathfrak{M}, s \underline{\leftrightarrow} \mathfrak{M}', s'$。当 $\mathfrak{M}$ 和 $\mathfrak{M}'$ 在上下文中清楚时，简写为 $s \underline{\leftrightarrow} s'$。

例 15. 验证图5-2中的两个模型是互模拟的。



(a) $\mathfrak{M}_1$ (b) $\mathfrak{M}_2$

图 5-2 互模拟的两个模型

类似于一阶逻辑的同构概念，互模拟是用来描述两个模态模型之间关系的概念。同时，互模拟的概念也揭示了模态语言的局限性，即使用模态语言无法区分两个互模拟的模型。

以下是关于互模拟的一个性质，可以通过对公式结构的归纳证明得到。

命题 1 (互模拟的不变性). 对两个模型 $\mathfrak{M}, \mathfrak{M}'$ 之间的任一互模拟关系 Z，如果 sZt, 则 s, t 是模态等价的。

那么，反过来，模态等价是否蕴涵存在互模拟关系？下面介绍模态逻辑中的经典结果，轩尼诗-米尔纳 (Hennessy-Milner) 定理。为此，先定义“像有穷”(Image-Finite) 的模型。如果对一个模型 $\mathfrak{M}$ 中的每个状态 s、关系 R，集合 $\{t \mid sRt\}$ 是有穷的，称该模型是像有穷的。

定理 18 (轩尼诗-米尔纳定理). 给定两个像有穷的模型 $\mathfrak{M}=(S,R,V),\mathfrak{M}'=(S',R',V')$，有 $s\underline{\leftrightarrow}s'$ 当且仅当 $s \rightleftsquigarrow s'$。

该定理的证明中最主要的想法是将模态等价关系直接定义为互模拟。细节可以参考Blackburn et al. (2001)的论文。

对于每一个模型 $(\mathfrak{M},s)$，总存在一个最小的与之互模拟的模型 $(\mathfrak{N},s)$，称为它的“互模拟收缩”(Contraction)。这个收缩模型可以看作原模型中模态信息的最简表达形式。给定一个模型，可以通过寻找其互模拟收缩模型对其进行简化。在模型检测的理论与实践中，互模拟是一项关键技术，因为它能够在保留原模型信息的前提下，显著减少需要分析的模型大小。这也正是为什么在模型验证之前，通常会利用互模拟 (或其他行为等价关系的概念) 对模型进行简化。

下面给出模态公式的有效性定义。

定义 39 (模型上普遍为真). 若一个公式 φ 在模型 $\mathfrak{M}$ 的所有世界上都为真，我们称该公式在模型 $\mathfrak{M}$ 上是普遍为真的。

定义 40 (模型上可满足). 若模型中有一些世界使得公式 φ 为真，称该公式在模型上是可满足的。

上面的定义可以扩展到公式集。一个公式集 Σ 在一个模型 $\mathfrak{M}$ 的所有世界上为真，称该公式集在模型 $\mathfrak{M}$ 上是普遍为真的。若公式集在模型的一些世界上为真，称该公式集在模型上是可满足的。

定义 41 (有效性). 若一个公式 φ 在基于框架 $\mathfrak{F}$ 的任意模型中的世界 s 上都是真的，称公式 φ 在框架 $\mathfrak{F}$ 的世界 s 上是有效的，记作 $\mathfrak{F},s\models\varphi$。若公式 φ 在框架 $\mathfrak{F}$ 的任意世界上是有效的，称 φ 在框架 $\mathfrak{F}$ 上是有效的，记作 $\mathfrak{F}\models\varphi$。若一个公式 φ 在框架类 F 中的任一框架 $\mathfrak{F}$ 上是有效的，则公式在该框架类上是有效的，记作 $\mathsf{F}\models\varphi$。若一个公式 φ 在所有框架类上是有效的，称该公式是有效的，记作 $\models\varphi$。

根据框架中可及关系所满足的性质，可以定义框架类。满足自反性的框架构成自反框架类，满足传递性的框架构成传递框架类。这些框架的性质可以对应模态逻辑的公式，常见的公式和性质的对应可以见表5-1。

表 5-1 模态逻辑公式与框架性质的对应关系

模态逻辑公式	对应的框架性质
T: $\Box p\to p$	自反性 (每个状态可及自身)
4: $\Box p\to\Box\Box p$	传递性 (如果状态 x 可及 y，y 可及 z，则 x 可及 z)
D: $\Box p\to\Diamond p$	持续性 (每个状态至少存在一个可及状态)
B: $p\to\Box\Diamond p$	对称性 (如果 x 可及 y，则 y 也可及 x)
5: $\Diamond p\to\Box\Diamond p$	欧几里得性质 (如果 x 可及 y 且 x 可及 z，则 y 与 z 之间可及)

T 对应自反性，表明每个状态都能“看见”自己；**4** 对应传递性，强调状态之间的传递联系；**D** 确保每个状态都有至少一个后继状态；**B** 强调状态之间的双向可及；**5** 描述了一种欧几里得性质，即状态之间的广泛联系。

当一个公式集在一框架类 F 上是有效的，称它是该框架类的逻辑，记作 Λ_{F}。

例 16. 证明 $\Box p \to p$ 不是有效的，但是它对于自反的框架类是有效的。

证明. 要证明 $\Box p \to p$ 不是有效的，只需要找到一个模型，使得公式在该模型的某个世界上为假。考虑模型 $\mathfrak{M}$, 其中，$W = \{w\}$，R 为空关系，$V(p) = \emptyset$。有 $\mathfrak{M}, w \models \Box p$ 且 $\mathfrak{M}, w \nvDash p$，因此，$\mathfrak{M}, w \nvDash \Box p \to p$。这样构造的模型，又称为反模型。

令 (W, R) 为一自反框架，即对于任意 $w \in W$ 都有 wRw 成立。任取基于该框架的模型 $\mathfrak{M} = (W, R, V)$，并任取一个世界 $w \in W$。为了证明 $\mathfrak{M}, w \models \Box p \to p$，假设 $\mathfrak{M}, w \models \Box p$。由 $\Box p$ 的语义可知：对于所有世界 $v \in W$，若 wRv，则 $\mathfrak{M}, v \models p$。由于框架是自反的，故 wRw 成立。从而得 $\mathfrak{M}, w \models p$。由此可见，在任何基于自反框架的模型中，$\Box p \to p$ 都普遍为真。故该公式对自反的框架类是有效的。 □

练习 33. 证明以下命题。

(1) $\Box p \to \Box\Box p$ 不是有效的，但对于传递的框架类是有效的。

(2) $p \to \Box\Diamond p$ 不是有效的，但对于对称的框架类是有效的。

(3) $\Box(p \to q) \to (\Box p \to \Box q)$ 在所有框架类上是有效的。

5.4 公理系统

在任意的框架类上有效的公式构成了模态逻辑 **K**，通常被称为最小的正规模态逻辑, 其希尔伯特式的公理化系统定义如下。注意，希尔伯特式的公理化系统跟自然演绎系统不同，它是由一些普遍有效的公理和推演规则构成的。在模态逻辑中，普遍的公理对于不同的系统有所不同。为了强调这一点，在本章选择了希尔伯特式的公理化系统。

定义 42 (公理系统 **K**). 模态逻辑 **K** 由下面的公理和推理规则构成：

(1) 所有命题逻辑重言式的特例；

(2) $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$；

– 分离规则 (MP)：从 $\varphi \to \psi$ 和 φ，推出 ψ；

– 必然化规则 (Nec)：从 φ，推出 $\Box\varphi$。

下面给出系统 **K** 中的定理证明。

例 17. 给出下面定理的证明：$\vdash_K \Box(p \wedge q) \to (\Box p \wedge \Box q)$。

证明. (1) $(p \wedge q) \to p$ (重言式)

(2) $\Box((p \wedge q) \to p)$ ((1); 必然化规则 Nec)

(3) $\Box((p \wedge q) \to p) \to (\Box(p \wedge q) \to \Box p)$ (K 公理实例)

(4) $\Box(p \wedge q) \to \Box p$ ((2),(3); 分离规则 MP)

(5) $(p \wedge q) \to q$ (重言式)

(6) $\Box((p \wedge q) \to q)$ ((5); 必然化规则 Nec)

(7) $\Box((p \wedge q) \to q) \to (\Box(p \wedge q) \to \Box q)$ (K 公理实例)

(8) $\Box(p \wedge q) \to \Box q$ ((6),(7); 分离规则 MP)

(9) $(\Box(p \wedge q) \to \Box p) \to ((\Box(p \wedge q) \to \Box q) \to (\Box(p \wedge q) \to (\Box p \wedge \Box q)))$ (重言式)

(10) $(\Box(p \wedge q) \to \Box q) \to (\Box(p \wedge q) \to (\Box p \wedge \Box q))$ ((4),(9); MP)

(11) $\Box(p \wedge q) \to (\Box p \wedge \Box q)$ ((8),(10); MP)

□

练习 34. 给出下面定理的证明。

(1) $(\Box p \wedge \Box q) \to \Box(p \wedge q)$。

(2) $\Diamond(p \vee q) \leftrightarrow (\Diamond p \vee \Diamond q)$。

(3) $(\Diamond p \wedge \Box(p \to q)) \to \Diamond q$。

K 之所以被称为最小的正规模态逻辑，因为它是关于框架类推理的正规系统中最弱的系统。通过添加其他的公理，可以获得其他更强的模态逻辑系统。在系统 **K** 的基础上添加 T 和 4，得到公理系统 **S4**；进一步添加 B 作为公理，就得到公理系统 **S5**。**S5** 通常定义在等价关系的框架类上，也就是自反、对称和传递的框架。事实上，在自反的前提下，加入欧几里得公理 (5：$\Diamond p \to \Box\Diamond p$) 可以推出对称性和传递性。因此，**S5** 可以通过在系统 **KT** 的基础上添加 5 来公理化，而不需要单独引入对称性公理 (B)。所以，我们也经常看到表述 **S5** = **KT5**。系统 **S4** 和 **S5** 在认知逻辑中是常见的系统将在下一章继续讨论。

5.5 元定理

定理 19 (可靠性和完全性). 对任意的模态公式 φ, $\vdash_K \varphi$ 当且仅当 $\models \varphi$。

证明思路与一阶逻辑的完全性证明类似，类似的方法也适用于上面提到的其他模态逻辑系统。关于其中的细节，可参考文献 Blackburn et al. (2001) 和 van Benthem (2010)。

在模态逻辑中，通常的判定问题具体可以如下表述为。

- 模型检测：给定一个具体的克里普克模型 $\mathfrak{M}$ (通常有穷) 和一个公式 φ，判断 $\mathfrak{M}, w \models \varphi$ 是否成立 (或对所有 w 都成立，依不同定义而定)。
- 可满足性：给定一个模态公式 φ, 是否存在一个克里普克模型 $\mathfrak{M}$ 和世界 w，使得 $\mathfrak{M}, w \models \varphi$？换句话说，$\varphi$ 能不能在某个模型的某个状态下被满足？
- 有效性：给定模态公式 φ，是否在所有的克里普克模型、所有世界上都为真？这等价

于 $\neg\varphi$ 是否不可满足。

下面是一些经典的结果。

定理 20 (可满足性). 在单一模态算子的正规模态逻辑 **K** 中，公式的可满足性问题是 PSPACE 完全的。此外，系统 **T**、**S4** 中公式的可满足性问题仍然是 PSPACE 完全的。**S5** 的可满足性问题是 NP 完全的。

证明思路 这个结果最早由Ladner (1977) 证明，主要思路如下。

- 上界：使用表列法 (Tableau) 或过滤法 (Filtration) 在多项式空间内完成可满足性的判定，从而归入 PSPACE。
- 下界：将某些典型的 PSPACE 困难问题 (如量化布尔公式的某些变形) 多项式时间还原至模态逻辑可满足性，由此获得 PSPACE 复杂度。

特别地，在可及关系上添加如自反 (**T**) 、传递 (**S4**) 并不会降低问题复杂度，可满足性依旧是 PSPACE 完全的。对 **S5** 来说，因为它的可及关系是一个等价关系 (自反、传递和对称)，也就是每个世界都可以访问每个世界。这使得 **S5** 的模型可以非常紧凑，能够有效降低复杂度。对这些系统的可满足性问题的讨论，可参见文献 Blackburn et al. (2001)、van Benthem (2010) 和 Ladner (1977)。

模型检测在计算机科学中广泛应用，例如，针对并发系统、分布式系统、硬件设计进行自动验证。

定理 21 (模型检测). 令 $\mathfrak{M} = (W, R, V)$ 为克里普克模型，其中 $|W| = n$，$R \subseteq W \times W$ 为可及关系，V 为命题变元的赋值函数。给定基本模态逻辑 **K** (或其扩展系统如 **T**、**S4**、**S5**) 中的公式 φ，则以下判定问题都可在关于 $|W| + |R|$ 与 $|\varphi|$ 的多项式时间内求解 (P 完全的)。

(1) 点模型检测：判断对某个固定世界 $w \in W$，是否有 $\mathfrak{M}, w \models \varphi$ 成立。

(2) 全局模型检测：判断在模型 $\mathfrak{M}$ 中是否所有世界都满足 φ，即 $\forall w \in W : \mathfrak{M}, w \models \varphi$。

更具体地，存在算法能在 $O\big((n + |R|) \cdot |\varphi|\big)$ 或其他多项式量级的时间内完成此判定。

证明思路 可以采用标记算法 (Labelling Algorithm)：将公式 φ 逐步分解为子公式，对每个子公式在模型的每个世界上标记其真值，反复利用 $\Box$ 与 $\Diamond$ 的语义规则 (即对后继可及世界的检查) 更新标记。该过程可在多项式时间内完成，得到每个世界对所有子公式的真值信息，从而可判断 $\mathfrak{M}, w \models \varphi$ 是否成立。

在系统 **T**、**S4**、**S5** 等中，可及关系具备自反、传递、对称或欧几里得性等性质，但这不会实质增加局部检查的复杂度，因而同样可以在类似的多项式时间算法中处理。该定理为在实践中进行可扩展的自动验证 (如程序或系统模型检测) 提供了坚实理论支持。

时态逻辑是对模态逻辑的进一步扩展，通常也能在多项式或指数级时间内做模型检测。这也是它在工业界被广泛用于“程序性质自动验证”的一个重要原因。

定理22 (有效性)**.** 在单一模态算子的正规模态逻辑 **K** 中，公式的有效性问题是 PSPACE 完全的。此外，系统 **T**、**S4** 中公式的有效性问题仍然是 PSPACE 完全的。**S5** 的有效性问题是 co-NP 完全的。

证明细节，参考 Ladner (1977) 的论文。

注记2. 可满足性关心的是“有没有某个模型可让公式为真”，这对“任意”模型都要考虑，故算法复杂度更高，在 PSPACE 范围。模型检测是给定一个具体模型，通常其复杂度与可满足性不处于相同量级。

注记3. 我们知道，一阶逻辑的有效性问题为不可判定，即不存在一个算法可以在有限时间内对所有一阶公式给出正确的“有效/无效”答案。一阶逻辑的可满足性同样不可判定。模态逻辑只有可能和必然两个算子，它们虽然也有量化的意义，但其作用是局部的，与一阶逻辑的全局量化有着本质的区别。模态逻辑在理论上是可判定的：我们可以给出一个算法，对任意模态公式进行有限步骤的构造或搜索，最终输出“可满足”或“不可满足”。这对很多需要自动推理的计算机应用来说是个好消息。

5.6　结语：时态逻辑、动态逻辑等扩展

在本章中探讨了模态逻辑的语言和语义，并介绍了公理系统、元定理和判定问题的经典结果。这为后续的模态逻辑的扩展奠定了基础。在结束本章之际，将简要介绍与模态逻辑及其应用紧密相关的两个方向：模型检测中的时态逻辑、程序执行语境下的动态逻辑。此外，将引入下一章的认知逻辑。

5.6.1　时态逻辑

时态逻辑主要用于描述系统状态随时间的演化。在模型检测中，时态逻辑是核心工具之一。通过引入时间维度，可以精确表达诸如“最终会发生”、“始终满足”或“直到某一条件成立”等性质，从而对系统的行为进行形式化描述和检测。常见的时态逻辑系统如下。

- LTL (Linear Temporal Logic，线性时态逻辑)：LTL 聚焦于线性时间模型，以序列方式刻画状态的演化。
- CTL (Computation Tree Logic，计算树逻辑)：在 CTL 中，状态公式和路径公式结合使用，可以对分支结构进行精细刻画。
- CTL*：作为 CTL 的推广，CTL* 允许更加灵活的时间和路径量化，使得表达能力更强。

这些逻辑都建立在克里普克结构的基础上，利用状态与状态之间的可及关系描述自动

验证系统的各项性质，为复杂系统的可靠性提供了理论支持。

LTL 的基本假设是时间是单一线性序列。换句话说，LTL 关注的是单一路径上的状态演化，而不区分不同分支。LTL 中常用的算子如下。

- $\mathsf{X}\varphi$：下一个状态 φ 成立。
- $\mathsf{F}\varphi$：在未来某个时刻 φ 会成立 (最终成立)。
- $\mathsf{G}\varphi$：在所有未来时刻 φ 始终成立。
- $\varphi\,\mathsf{U}\,\psi$：$\varphi$ 一直成立，直到 ψ 成立为止。

LTL 假定系统行为可以用一条单一的“时间线”描述，因此非常适合于描述单个执行轨迹上的行为。其语法和语义较为直观，但在表达需要考虑多个分支的性质时可能不如 CTL 或 CTL* 灵活。关于时态逻辑的著作很多，不在这里赘述。感兴趣的读者可以参考 Prior (1967) 的先驱性工作，和计算机领域影响很大的 Pnueli (1977) 的工作。

5.6.2 命题动态逻辑

命题动态逻辑 (Propositional Dynamic Logic，PDL) 是一种专门用于描述和分析程序行为的模态逻辑。它的核心思想是将程序的执行过程抽象为状态之间的转换，并利用模态算子描述程序执行后的性质。

具体而言，在 PDL 中，每个程序 α 对应一个二元关系 R_α，定义在一个克里普克结构的状态集合 S 上。形式化地说，如果 $(s,t) \in R_\alpha$，则表示在状态 s 执行程序 α 后，系统可能转移到状态 t。这种抽象将程序执行转化为状态之间的“跳转”，从而为形式化分析提供了直观的图模型。PDL 采用模态算子刻画程序执行后的性质，主要有以下两种表达方式。

- $[\alpha]\varphi$ 表示：在当前状态下，如果程序 α 可执行，则对于所有通过 α 转移可达的状态，命题 φ 都成立。这表达了“必然性”，即程序执行后无论如何演化，φ 都得满足。
- $\langle\alpha\rangle\varphi$ 表示：存在一条程序 α 的执行路径，使得执行后达到的某个状态满足 φ。这表达了“可能性”，说明至少有一种执行方式可以达到满足 φ 的状态。

为了描述复杂的程序行为，PDL 允许利用基本程序构造更复杂的表达式，其常见的算子如下。

- 序列组合：若 α 和 β 是两个程序，则 $\alpha;\beta$ 表示先执行 α 再执行 β。对应的状态转换关系是 $R_{\alpha;\beta} = R_\alpha \circ R_\beta$，即状态 s 经过 α 后转移到某状态，再经过 β 转移到目标状态。
- 非确定性选择：$\alpha \cup \beta$ 表示在执行时可非确定性地选择执行 α 或 β 中的任一者。这反映了程序设计中的分支和选择行为。
- 迭代：α^* 表示程序 α 的零次或多次重复执行，涵盖了循环结构和递归调用等情况。

这些算子大幅增强了 PDL 的表达能力，使其能够描述复杂的程序控制流程。

从模型的角度看，PDL 中的状态转移常常以加标转换系统 (Labeled Transition System) 刻画。该系统由一组状态及标记了程序的转移边构成，其中每条边上的标记就是相应的程序。这种图模型直观地展示了所有可能的程序执行路径，因此在形式化验证、程序正确性证明以及模型检测等领域中发挥着重要作用。借助 PDL，可以形式化地证明程序在所有可能的执行路径上都能满足期望的性质，或者通过寻找反例来识别程序中的潜在错误。这为自动化工具的开发提供了理论支持，并在实际的软件工程和系统设计中得到了广泛应用。感兴趣的读者可以进一步参考文献 Harel et al. (2000)。

从下一章开始，我们介绍认知逻辑 (Epistemic Logic)，其主要表示和推理智能体的知识和信念，其核心在于将模态算子 $\Box$ 解释为“智能体知道 φ”。我们将深入研究认识逻辑的语法、语义及其在知识表示和多智能体系统中的应用，从而更好地理解认知状态与信息交流的形式化模型。