第3章 确定性知识表示与推理

第 2 章讨论了理性 Agent 的概念及其典型结构。无论是模型反射型 Agent 还是目标驱动型 Agent 均需要知识以描述运行环境及应对策略。按照符号主义学派的观点，知识是一切智能行为的基础，要使机器具有智能，就必须使它拥有并可以使用知识。以目前的技术水平，计算机还不能直接理解人类用自然语言表示的知识。如果需要计算机具有知识，则首先需要将知识以计算机能够处理的方式表示出来，而计算机对知识的应用可以看成一个推理过程。本章将介绍知识表示方法和知识的推理方法。按照知识的确定化程度，知识表示和推理方法可分为确定性和不确定性两类。由于篇幅所限，本书仅讨论确定性知识的表示和推理。

3.1 确定性知识系统概述

3.1.1 确定性知识表示的概念

既然要讨论知识的表示，那么首先应该明确何为知识？我们将从知识的定义和知识的类型进行讨论。

1. 知识的定义

我们的日常生活中充满了各种知识，但要给知识下一个严格的定义并不容易。一种普遍的观点认为，知识是人们在改造客观世界的实践中积累起来的认识和经验。从信息处理的角度看，知识是对信息进行智能性加工所形成的对客观世界规律性的认识。所以，知识可以看成将多种信息关联在一起，并形成规律性认识的过程。从这种意义上讲，知识是由“信息”和“关联”两个基本要素构成的。

实现信息之间关联的形式可以有很多种，其中最常用的一种形式是“如果……则……”。这种知识称为“规则”，其反映了信息间的某种因果关系。规则性的知识表示较为符合人类的思维习惯，如：“如果今天不下雨，则我去跑步”“如果油箱没油了，则汽车不能开了”“如果肚子饿了，则去拿些饼干吃”等。

2. 知识的类型

按照不同的划分方法，知识可以分为不同的类型。

1）按知识的适用范围分类

按知识的适用范围，知识可分为常识性知识和领域性知识。

常识性知识是指普通的知识或一般的知识，是人们普遍知道的知识。如“雪是白色的”“太阳从东边升起”“天冷了要加衣服”等。

领域性知识是指面向某个具体领域的专业性知识，需要经过长期、专门的学习和训练才能较好掌握及应用的知识。如“飞机发动机的构造知识”“如何控制核电站的反应堆”“如何决策在股市中的投资”等。

2）按知识的作用与效果分类

根据知识的作用与效果，知识可分为陈述性知识、程序性知识和策略性知识。

陈述性知识也称为描述性知识，是能用言语进行直接陈述的知识，主要用于区别和辨别事物，回答“是什么”“为什么”等问题。如“王华爱祖国”“我是一名学生”“我出生并

成长于重庆，因此我是重庆人”等。

程序性知识即操作性知识，是描述在问题求解过程中所需要的操作、算法或行为等规律性的知识，表现为在信息处理活动中进行具体操作，主要回答“怎么做”的问题。如“菜谱中描述的红烧肉制作过程”“三阶魔方还原的方法”等。

策略性知识是运用陈述性知识和程序性知识去学习、记忆、解决问题的一般方法和技巧，可以看成知识的知识。如《三国演义》中描绘了孙策遇刺后留给孙权的遗言：“内事不决问张昭，外事不决问周瑜。”孙策说明了应该如何利用张昭和周瑜两人所具有的专家知识，这就是一种策略性知识。

3）按知识的确定性分类

按照知识的确定性，知识可分为确定性知识和不确定性知识。

确定性知识是可以给出其真值为“真”或“假”的知识。如“曹操是三国时期著名的政治家”“明朝万历皇帝在位 48 年”“在 19 世纪，人类设计的宇宙飞船已能够达到光速”等。上述三个确定性知识中，前两个为真，后一个为假。

不确定性知识是指难以直接给出真或假的知识，这一类知识具有“不确定”的特性，包括不完备性、不精确性和模糊性的知识等。不完备性知识是指在解决问题时不具备解决该问题所需要的全部知识。如自动驾驶汽车机器人 Agent，假定其不能实时获取道路交通流量和拥堵情况数据，则该 Agent 在进行最优路径规划时所具有的知识是不完备的。因此，该 Agent 选择的所谓“最优”路径可能发生拥堵。不精确性知识是指具有一定的概率为真、一定的概率为假的知识。如天气预报给出“明天上午下雨的概率为 30%”，那么我们无法回答明天上午到底是下雨还是不下雨。这样的知识就是不精确的知识。模糊性知识是指知识本身所描绘的概念没有确切的定义，在量上没有确定界限的知识。如“如果风大，则帆船的速度就快”。对于上述知识，我们人类理解起来没有任何问题。但该知识本身却存在模糊性。知识中的“风大”到底是指风有多大，是 7 级、9 级还是 11 级？而“帆船的速度就快”是指多快，40 千米/小时、60 千米/小时还是 80 千米/小时？另一个典型的例子是第 1 章中提到的“当车的速度快时，刹车的距离就长”的例子。

限于篇幅，本书仅讨论确定性知识的表示与推理。

3. *知识表示的概念*

知识的表示就是对知识的一种描述或者对知识的一组约定，一种计算机可以接受的用于描述知识的数据结构。换句话说，就是用一些约定的符号把知识编码成一组可以被计算机直接识别，并便于系统使用的数据结构。由此可知，知识表示不仅是为了把知识用某种机器可以直接识别的数据结构表示出来，更重要的是要能够方便系统正确地运用和管理知识。

通常，对知识表示的要求有以下四方面。

（1）表示能力。知识的表示能力是指能否正确、有效地将问题求解所需要的各种知识表示出来。这是对知识表示方法的最低要求。

（2）可利用性。知识的可利用性是指经过了知识表示处理后的知识能够被计算机直接使用，从而求得问题的解。该过程即为知识推理，将在 3.3 节中详细讨论。

(3) 可维护性。知识的可维护性是指在保证知识的一致性和完整性的前提下,对知识进行的增加、删除、修改等操作。也就是能方便地完成知识的定期更新。

(4) 可理解性。知识的可理解性是指所表示的知识应当符合人类的认知与思维习惯。人类能够很好地理解知识的内涵及计算机对该知识的使用过程。

3.1.2 确定性知识推理概述

从符号主义学派的观点来看,人工智能的推理研究就是要基于人类的思维机理,去实现机器的自动推理。机器完成自动推理包括推理的方法和推理的控制策略两个基本问题。下面先给出推理的概念,再介绍推理的分类。

1. 推理的概念

按照心理学的观点,推理是由具体事例归纳出一般规律,或者根据已有知识推出新的结论的思维过程。其中,比较典型的观点有以下两种。

1) 推理的结构观点

从推理的结构角度出发,认为推理由两个以上判断组成。每个判断揭示的是概念之间的联系和关系,推理过程是一种对客观事物做出肯定或否定的思维活动。例如:

判断 1:凡是体育专业的学生,身体素质都很好。

判断 2:李华是体育专业的学生。

由此可以得出结论:李华的身体素质很好。

2) 推理的过程观点

从过程的角度出发,可以认为推理是在给定信息和已有知识的基础上进行的一系列加工操作。基于此,推理可以形式化地表示为如下映射:

$$y = f(x, k) \tag{3-1}$$

式中:x 为推理时输入的信息;k 为推理时可用的领域知识和特殊案例;f 为可用的一系列操作;y 为推理过程所得到的结论。

推理的过程就是映射 f 将输入的信息与领域知识(或特殊案例)进行综合,并得出推理结论的过程。

3) 推理的机器实现

推理过程是由推理机完成的。推理机是指系统中用来实现推理的那段程序。根据推理所用知识以及推理方式和推理方法的不同,推理机的构造也有所不同,这将在 3.3 节中详细讨论。

2. 推理的分类

按照推理的逻辑基础,常用的推理方法可分为演绎推理和归纳推理。

1) 演绎推理

演绎推理是一种由一般到个别的推理方法,其从已知的一般性知识出发推理出蕴涵在这些知识中的适合某种个别情况的结论,其核心是“三段论”。常用的三段论由大前提、小前提和结论三部分组成。该推理方法可以追溯到古希腊哲学家亚里士多德提出的苏格拉底三段论。

例 3.1 苏格拉底三段论。

所有人都是必死的(大前提)。

苏格拉底是人(小前提)。

苏格拉底是必死的(结论)。

由例 3.1 可知,“所有人都是必死的”是一般性的论断,称为大前提;而“苏格拉底是人”讲的是苏格拉底的属性,较“所有人都是必死的”这个论断的范畴要小,因而称为小前提。从大前提和小前提两个条件可以推导出结论“苏格拉底是必死的”,于是完成了一般(所有人都必死)到个别(苏格拉底必死)的推理过程。这就是一个典型的演绎推理。

显然,在演绎推理中,大前提是由已知的一般性知识或推理过程得到的判断;小前提是关于某种具体情况或某个具体实例的判断;结论是由大前提推出的,并且适合于小前提的判断。

2) 归纳推理

归纳推理是一种由个别到一般的推理方法。该方法试图从一类事物的大量特殊事例出发推理出该类事物的一般性结论。数学归纳法就是归纳推理的典型例子,其通过对初始条件及递推关系的归纳,实现了数学证明的过程。归纳推理可分为完全归纳推理和不完全归纳推理。

在完全归纳推理中,可以枚举论域中所有对象进行考查,逐一验证这些对象是否具有某种属性,然后归纳推理出该类对象是否具有此属性。例如,某工厂的检验员对该工厂生产的产品逐一检验,以说明该厂的这批产品是合格的。

不完全归纳推理是指在进行归纳时主观或客观原因,只考查了全体对象的某个子集(采样而得到的部分对象)而得出的关于该对象的结论。例如,某工厂的检验员从该工厂生产的一批产品中挑选一部分进行检验,并说明该厂的这批产品是合格的。显然,不完全归纳推理得到的结果可能是错误的。抽检的产品全部合格,并不能说明整批产品就一定合格,但其至少是对该批产品质量的一个可信的估计。

3.2 确定性知识的表示

确定性知识是指能够明确给出真或假的知识。本节将重点介绍谓词逻辑表示法、产生式表示法和语义网络表示法等常用的知识表示方法。

3.2.1 谓词逻辑表示法

谓词逻辑表示法是一种基于数理逻辑的知识表示方式。人工智能中用到的数理逻辑包括一阶经典逻辑和一些非经典逻辑。本节讨论基于一阶经典逻辑的知识表示方法。

1. 谓词逻辑表示的基础知识

使用谓词逻辑表示知识,需要掌握的概念包括命题、论域、谓词、连词、量词、谓词公式等。

1) 命题的概念

讲到命题,首先需要明确以下几个相关概念。

断言：一个陈述句称为一个断言。

命题：凡有真假意义的断言称为命题。例如，“我是一名信息工程专业的学生”“今年冬天很冷”“王华爱祖国”等都是命题。

命题的意义称为真值，真值仅有“真”和“假”两种情况。命题的真值为真，记为 T(True)；命题的真值为假，则记为 F(False)。

2）论域与谓词

论域：由所讨论对象的全体构成的非空集合。论域中的元素称为个体。因此，论域有时也称为**个体域**。例如，有理数的论域就是所有有理数组成的集合，每一个有理数就是该论域中的一个个体。

在谓词逻辑中，谓词被用来表示命题。谓词可分为个体和谓词名两部分。个体是命题中的主语，是可以独立存在的实体。谓词名是命题的谓语，是用来刻画个体词的性质、状态或与其他个体关系的词。

例 3.2 命题的谓词表示方法举例。

命题“李明是一名信息工程专业的学生”可用谓词表示为

$$\text{INF\_STUDENT(Li Ming)}$$

命题“王华在跑步”可用谓词表示为

$$\text{RUNNING(Wang Hua)}$$

命题“李明和王华是同学”可用谓词表示为

$$\text{CLASSMATE(Li Ming, Wang Hua)}$$

在例 3.2 中，Li Ming、Wang Hua 等均是个体，是独立存在的实体，在命题中是主语。INF_STUDENT、RUNNING、CLASSMATE 就是谓词名，在命题中是谓语。INF_STUDENT(x)表示个体 x 是信息工程专业的学生，表达个体 x 具有“信息工程系学生”的性质；RUNNING(x)表示个体 x 正在跑步，表达个体 x 的状态是“正在奔跑”；CLASSMATE(x, y)表示个体 x 和个体 y 是同学，表达了两个个体之间的关系是“同学”。

通常，谓词名用大写英文字母表示，个体用小写英文字母表示。

谓词可形式化定义如下：

谓词：设 D 为论域，$P: D^n \to \{\text{True}, \text{False}\}$ 是一个映射，其中，$D^n = \{(x_1, x_2, \cdots, x_n) \mid x_1, x_2, \cdots, x_n \in D\}$，则称 P 是一个 n 元谓词($n=1,2,\cdots$)，记为 $P(x_1, x_2, \cdots, x_n)$，其中 $x_1, x_2, \cdots x_n$ 为个体。

在谓词中，个体可以是常量、变量或函数。例如，“$x<3$”可以用谓词表示为 LESS(x,3)，式中，x 是变量，3 是常量，它们都是谓词 LESS 的个体。又如，“李明的哥哥是医生”可以用谓词表示为 DOCTOR(old_brother(LiMing))，其中，old_brother(LiMing)代表李明的哥哥，是一个函数。DOCTOR(x)是谓词，表示个体“x 是医生”这个命题。

从形式上看，谓词和函数极为类似，两者却有本质的区别。下面给出函数的定义。

函数：设 D 为论域，$f: D^n \to D$ 是一个映射，则称 f 为论域 D 上的是一个 n 元函数($n=1,2,\cdots$)，记为 $f(x_1, x_2, \cdots, x_n)$，其中，$x_1, x_2, \cdots, x_n$ 为个体。

从谓词与函数的定义可知，谓词和函数最大的区别在于，谓词的真值是 True 或 False，而函数无真值可言，其值是论域中的某个个体。谓词实现的映射是将论域中个体（单个或多个）映射到真值，而函数是将论域中个体（单个或多个）映射为论域中某个个体。在谓词逻辑中函数不能单独使用，它必须嵌入谓词之中。

3）连接词

在谓词逻辑中，连接词是用来连接简单命题，并由简单命题构成复合命题的逻辑运算符号。谓词逻辑表示法中，常用的连接词有以下 5 个。

$\neg$：称为“非”或者“否定”。它表示对其后面的命题的否定，使该命题的真值与原来相反。如果命题 P 为真，则 $\neg P$ 为假，反之亦然。

$\vee$：称为“析取”。它表示所连接的两个命题之间具有“或”的关系。例如，对于命题 P 或 Q，$P \vee Q$ 读作“P 析取 Q”，表示命题 P 或 Q 其中任意一个为真，则 $P \vee Q$ 为真。

$\wedge$：称为“合取”。它表示所连接的两个命题之间具有“与”的关系。例如，对于命题 P 或 Q，$P \wedge Q$ 读作“P 合取 Q”，表示命题 P 和 Q 必须同时为真，则 $P \wedge Q$ 为真。

$\rightarrow$：称为“条件”或“蕴涵”。它表示“若……则……”的语义。例如，对于命题 P 或 Q，$P \rightarrow Q$ 读作 P 蕴涵 Q，表示 P 是 Q 的逻辑前提，而 Q 是 P 的逻辑结论。

$\leftrightarrow$：称为“双条件”或“等价于”。它表示“当且仅当”的语义。例如，对于命题 P 或 Q，$P \rightarrow Q$ 读作“P 等价于 Q”，表示 P 当且仅当 Q。

在谓词公式中，连接词的优先级由高到低依次是 $\neg$、$\wedge$、$\vee$、$\rightarrow$、$\leftrightarrow$。

命题逻辑可看成谓词逻辑的一种特殊形式，所以命题公式是谓词公式的一种特殊情况，也可用连接词把单个命题连接起来，构成命题公式。例如，$P \wedge Q$、$\neg P \vee \neg Q$、$\neg (P \rightarrow Q) \wedge (P \vee Q)$ 等都是命题公式。

4）量词

量词是由量词符号和被其量化的变元组成的表达式，用来对谓词中的个体做出量的规定。谓词逻辑表示方法中常用的量词有两个。

$\forall$：称为全称量词，表示“所有的”或“任意一个”。例如，对于 $(\forall x)P(x)$，当且仅当对论域中的所有个体 x，$P(x)$ 都为真，则 $(\forall x)P(x)$ 才为真，否则 $(\forall x)P(x)$ 为假。

$\exists$：称为存在量词，表示“某一个”或“存在一个”。例如，对于 $(\exists x)P(x)$，只要论域中存在一个个体 x，使得 $P(x)$ 为真，则 $(\exists x)P(x)$ 才为真。当且仅当论域中所有个体 x，$P(x)$ 均为假，则 $(\exists x)P(x)$ 为假。

随着量词的引入，就有辖域、约束变元和自由变元的概念。

辖域：位于量词后面的单个谓词或者用“()”括起来的合式公式称为该量词的辖域。

约束变元：辖域内与量词中同名的变元称为约束变元。

自由变元：辖域外或不受量词约束的变元称为自由变元。

例 3.3 谓词公式中的辖域、约束变元与自由变元。

(1) $(\exists x)\big(P(x,y) \rightarrow Q(y)\big) \wedge H(x,y)$。

(2) $(\forall x)\Big(\big(P(x,y) \rightarrow Q(y)\big) \wedge H(x,y)\Big)$。

在例 3.3(1)中，$P(x,y)$中的变元 x 为约束变元，而 $H(x,y)$中 x 为自由变元，所有的变元 y 均是自由变元。而在例 3.3(2)中，所有的变元 x 均为约束变元，而 y 均是自由变元。

2. 谓词逻辑知识表示

谓词逻辑具有丰富的知识表示能力。当用谓词逻辑表示知识时，首先需要定义谓词，再用连接词或量词把这些谓词连接起来，形成一个谓词公式，从而表示相关的知识。

例 3.4 用谓词逻辑表示知识“所有的整数不是奇数就是偶数”。

解： 首先定义谓词。

$I(x)$：x 是整数。

$E(x)$：x 是偶数。

$O(x)$：x 是奇数。

基于上述谓词定义，该知识可表示为

$$(\forall x)(I(x) \rightarrow E(x) \vee O(x))$$

上述谓词公式可以解读为：对于论域中任意个体 x，如果 x 是整数，则其蕴涵的逻辑结论是要么 x 是奇数，要么 x 是偶数。

例 3.5 用谓词逻辑表示知识“所有的将军都有自己的士兵”。

解： 首先定义谓词。

GENERAL(x)表示 x 是将军。

SOLDIER(y)表示 y 是士兵。

COMMAND(x,y)表示 x 指挥 y。

基于上述谓词定义，该知识可表示为

$$(\forall x)(\exists y)(\text{GENERAL}(x) \rightarrow \text{SOLDIER}(y) \wedge \text{COMMAND}(x,y))$$

上述谓词公式可以解读为：对于论域中任意个体 x，如果 x 是将军，则论域中一定存在个体 y，y 是士兵，且将军 x 指挥士兵 y。

例 3.6 用谓词逻辑表示知识“小李住在一栋红色的房子里”。

解： 首先定义谓词。

LIVE(x,y)表示 x 住在 y。

COLOR(y,z)表示 y 的颜色是 z。

上述谓词中，个体 x 的论域是人类集合，个体 y 的论域是各式各样的房屋集合，个体 z 的论域是各种各样的颜色集合。

基于上述谓词定义，该知识可表示为

$$\text{LIVE}(\text{Li},\text{House1}) \wedge \text{COLOR}(\text{House1},\text{Red})$$

上述谓词公式可以解读为：对于个体 Li(小李)住在一栋房屋 House1 中，且房屋 House1 的颜色是 Red(红色)的。

例 3.7 用谓词逻辑表示如下知识：

“王宏是计算机专业的一名学生”。

“王宏和李明是同班同学”。

“凡是计算机专业的学生都喜欢编程序”。

解：首先定义谓词。

CSD(x)表示个体 x 是计算机专业的一名学生。

CM(x,y)表示个体 x 和 y 是同班同学。

LIKE(x,z)表示个体 x 喜欢 z。

基于上述谓词定义，该知识可表示为

$$\text{CSD(Wang Hong)}$$

$$\text{CM(Wang Hong,Li Ming)}$$

$$(\forall x)(\text{CSD}(x) \rightarrow \text{LIKE}(x,\text{Programming}))$$

3. 谓词逻辑知识表示的经典问题

上面讨论了一阶谓词逻辑的基础和逻辑知识表示方法，本节讨论一个经典的使用谓词逻辑表示法的例子。

例 3.8 机器人搬箱子问题。

在一个房间里，c 处有一个机器人，a 和 b 处各有一张桌子，分别称为桌 a 和桌 b，桌 a 上有一箱子，如图 3.1 所示。要求机器人从 c 处出发把箱子从桌 a 上拿到桌 b 上，再回到 c 处。试用谓词逻辑来描述机器人的行动过程。

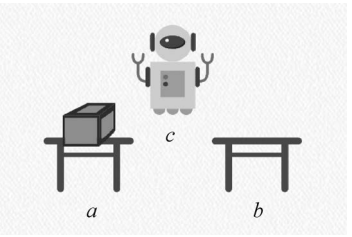

图 3.1 机器人搬箱子问题

解：在该例子中，不仅需要使用谓词来表示机器人、箱子、桌子的位置和状态，还需要表示机器人的操作。因此，需要定义谓词如下：

TABLE(x)：x 是桌子。

EMPTY(y)：y 手中是空的。

AT(y,z)：y 在 z 处。

HOLDS(y,w)：y 拿着 w。

ON(w,x)：w 在 x 桌面上。

其中，x 的论域是{a,b}，y 的论域是{robot}，z 的论域是{a,b,c}，w 的论域是{box}。

问题的初始状态可表示如下：

AT(robot,c)

EMPTY(robot)

ON(box,a)

TABLE(a)

TABLE(b)

问题的目标状态如下：

AT(robot,c)

EMPTY(robot)

ON(box,b)

TABLE(a)

TABLE(b)

机器人行动的目标是把问题的初始状态转换为目标状态，需要完成一系列的操作。机器人的每个操作可分为条件和动作两部分。条件部分用来说明执行该操作必须具备的前提条件，动作部分说明该操作是如何改变问题状态的。条件部分可用谓词公式来表示，动作部分通过在执行该操作前的问题状态中删去或增加相应的谓词来实现。在本问题中，机器人需要执行以下 3 个操作：

GOTO(g,h)：机器人从 g 处走到 h 处。

PICKUP(g)：机器人在 g 处拿起盒子。

SETDOWN(g)：机器人在 g 处放下盒子。

g 和 h 的论域是{a,b,c}。

这三个谓词操作对应的前提条件和动作如下：

GOTO(g,h)

条件：AT(robot,g)。

动作：删除表 AT(robot,g)，添加表 AT(robot,h)。

PICKUP(g)

条件：ON(box,g)，TABLE(g)，AT(robot,g)，EMPTY(robot)。

动作：删除表 EMPTY(robot)，ON(box,g)，添加表 HOLDS(robot,box)。

SETDOWN(g)

条件：HOLDS(robot,box)，TABLE(g)，AT(robot,g)。

动作：删除表 HOLDS(robot,box)，添加表 EMPTY(robot)，ON(box,g)。

机器人在执行每个操作之前，都需要检查当前状态是否可以满足该操作的前提条件。如果满足，就执行相应的操作；否则，检查下一个操作所要求的前提条件。该过程类似于产生式推理中的事实和规则前提匹配，将在 3.3.1 节中详细讨论。

作为谓词逻辑知识表示方法的应用举例，图 3.2 给出了机器人移盒子问题的求解过程，该过程实际上是一个搜索过程，其目的是找到一条从初始状态到目标状态的转换路径。搜索策略将在第 4 章详细讨论，这里讨论的重点是谓词逻辑知识表示方法。

4. 谓词逻辑表示法的特性

谓词逻辑知识表示的主要特点是建立在一阶逻辑的基础上，并利用逻辑运算方法研究推理的规律，即条件与结论之间的蕴涵关系。逻辑表示法的主要优点如下。

(1) 自然性。谓词逻辑是一种接近于自然语言的形式语言系统，用谓词逻辑表示出的知识与人类的认知类似，也符合人类思维习惯。

(2) 明确性。谓词逻辑表示法对如何由简单陈述句构造复杂陈述句的方法有明确规定，连接词、量词的用法与含义非常明确。逻辑表示法表示的知识可以按照一种标准的方法去解释它，不会导致歧义。

(3) 精确性。谓词逻辑是一种二值逻辑，其谓词公式的真值只有“True”和“False”。因此，谓词逻辑常用来表示精确的知识。

(4) 灵活性。逻辑表示法把知识表示和知识处理的程序有效地分开，处理谓词逻辑

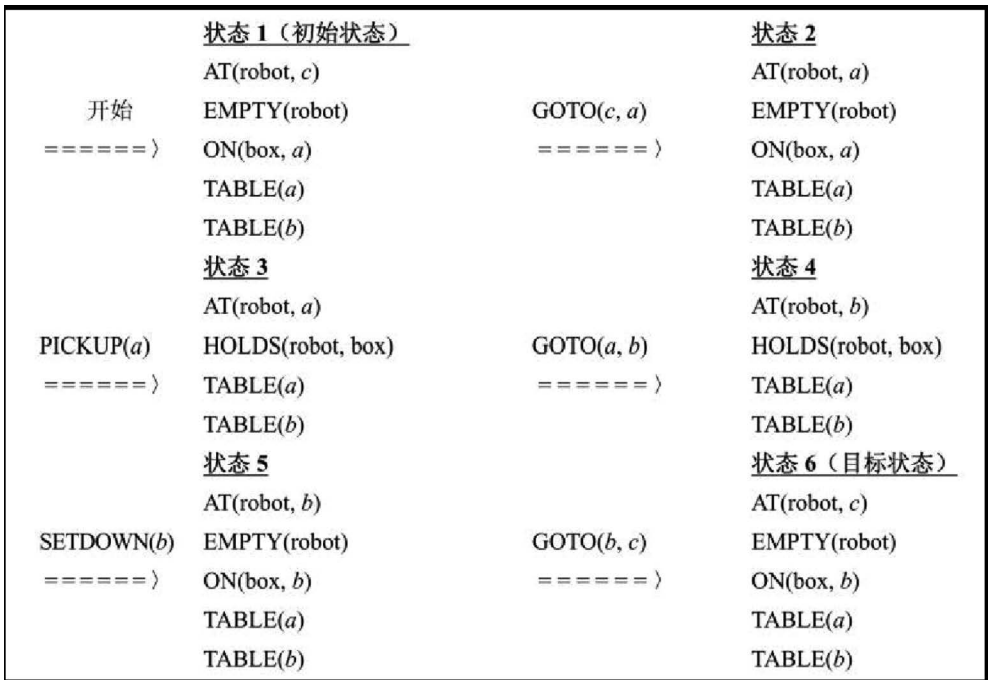

图 3.2　用谓词逻辑表示的机器人搬箱子问题的求解过程

知识程序的变化不会影响知识本身的谓词逻辑表示。

（5）可维护性。在谓词逻辑表示法中，不同的知识用单独的谓词进行表示，各条知识之间本质上是独立的。因此，添加、删除、修改知识的工作比较容易进行。

逻辑表示法也存在以下不足。

（1）知识表示能力弱。逻辑表示法只能表示确定性知识，而不能表示非确定性知识，如不精确、不完备及模糊性知识。实际上，人类的大部分知识都不同程度地具有某种不确定性，这就使得逻辑表示法表示知识的范围和能力受到了一定限制。另外，逻辑表示法难以表示过程性知识和策略性的知识。

（2）组合爆炸明显。由于谓词逻辑法不同知识之间的关系无法明确表达，从问题初始状态到目标状态的推理路径并不明确，需要大量地试探和盲目地推理。因此，当系统知识量较大时，容易发生组合爆炸。

（3）系统效率低。谓词逻辑表示法的推理过程是根据形式逻辑进行的，把推理演算与知识含义截然分开，抛弃了表达内容中所含有的语义信息，往往使推理过程冗长，降低了系统效率。

3.2.2　产生式表示法

产生式这一术语是 1943 年由美国数学家波斯特（E. Post）首次提出并使用的。到 1972 年，纽厄尔和西蒙在研究人类的认知模型中开发了基于规则的产生式系统。目前，产生式表示法已成为人工智能中应用最多的一种知识表示方式，尤其是在专家系统方

面，许多成功的专家系统采用产生式表示方法。本节重点讨论确定性知识的产生式表示方法，产生式推理部分将在 3.3.1 节讨论。

1. 产生式表示法中的事实

在产生式表示法中，事实可看成断言一个语言变量的值或断言多个语言变量之间关系的陈述句。其中，语言变量的值或语言变量之间的关系可以是数字，也可以是一个描述性的词等。

例如：陈述句“雪是白的”，其中“雪”是语言变量，“白”是语言变量的值；“该船的最高航速是 30 节/小时”中“该船的最高航速”是语言变量，“30 节/小时”是语言变量的值。又如：陈述句“王华是张老师的学生”，“王华”和“张老师”均是语言变量，而“某人为某人的学生”是两个语义变量之间的关系。所以，在产生式表示法中确定性知识可由两类三元组表示。

第一类：(Object，Attribute，Value)

其中：Object 表示对象；Attribute 表示属性；Value 表示值。

如对于陈述句“雪是白的”，Object＝“雪”，Attribute＝“颜色”，Value＝“白色”；陈述句“该船的最高航速是 30 节/小时”，Object＝“该船”，Attribute＝“最高航速”，Value＝“30 节/小时”。

第二类：(Object1，Object2，Relationship)

其中：Object1 和 Object2 分别表示两个不同的对象；Relationship 用来描述两个对象之间的关系。

如对于陈述句“王华是张老师的学生”，Object1＝“王华”，Object2＝“张老师”，Relationship＝“学生-教师关系”。

2. 产生式规则

产生式规则用于描述事物间的因果关系，通常用 IF…THEN…(如果……则……)的形式表达。产生式规则可简称为规则。典型的产生式规则由规则前项(前提条件)和规则后项(结论或动作)两部分组成。规则前项是该规则可否使用的前提条件，由单个事实或多个事实的逻辑组合构成；规则后项是一组逻辑结论或动作，指出当规则前项被满足时，应该得出的逻辑结论或应该执行的相关动作。

产生式规则的形式化描述如下。

<规则> ::＝＜规则前项>→＜规则后项>

<规则前项> ::＝<简单条件> or <复合条件>

<规则后项> ::＝<事实> or <动作>

<复合条件> ::＝(<简单条件> and <复合条件>) or (<简单条件> or <复合条件>) or (<复合条件> and <简单条件>) or (<复合条件> and <简单条件>)

<动作> ::＝<动作名>(<个体 1 >, <个体 2 >, …)

例 3.9 产生式规则举例。

(1) IF “汽车没油了” THEN “汽车无法行驶”。

(2) IF“汽车没油了”THEN“给汽车加油”。

(3) IF“当前季节是秋季”and“天空乌云密布”and“天气预报未来有雨”THEN“出门带上雨伞”。

(4) IF(“天空正在下大雨”or“天空正在下大雪”) and“家里没有雨具”THEN“请尽量别出门”。

在上述4个例子中,规则(1)和规则(2)的规则前项均是简单条件,规则(3)和规则(4)的规则前项均为复合条件。规则(1)与规则(3)的规则后项是逻辑结论,规则(2)和规则(4)的规则后项则是应当采取的动作。

3. 产生式表示法的特性

产生式表示法的主要优点如下。

(1) 自然性。产生式表示法用“IF…THEN…”的形式表示知识,这种表示形式与人类的判断性知识基本一致,非常直观,易于人类理解。人类专家也能够较好地用“IF…THEN…”结构表达自己的专业知识。

(2) 易维护性。每条产生式规则都是一个独立的知识单元,各产生式规则之间不存在相互调用关系,增加、删除、修改某一条规则不会对已有规则产生影响。

(3) 适用性。产生式表示除用来表示确定性知识,稍做变形(如在规则后项中加入概率或是置信度等)就可用来表示不确定性知识。如专家系统MYCIN用了450条带有置信因子的规则,用概率推理的方式诊断血液病的类型,其诊断结果的准确性高于了初级医生。

(4) 灵活性。产生式表示法把知识表示和知识处理的程序有效地分开,其不必关心推理程序的实现细节。推理程序的完善与升级也不会影响根据产生式规则推导出的逻辑结论。

产生式表示法的主要缺点如下。

(1) 组合爆炸明显。由于各规则之间的联系无法明确表达,从问题初始状态到目标状态的推理路径并不明确,需要大量地试探和盲目地推理。因此,当系统知识量较大时,容易发生组合爆炸。

(2) 推理效率较低。基于产生式表示法的问题求解过程实际上是事实与产生式规则反复进行“匹配—冲突消解—执行”的过程,即先用规则前提与数据库中的已知事实进行匹配,再从规则库中选择可用规则,当有多条规则可用时,还需要按一定策略进行“冲突消解”,然后才能执行选中的规则。这样的执行方式将导致数据库和规则库的多次遍历及重复冲突消解,执行的效率较低。

(3) 不便于表示结构性知识。产生式表示中的知识均以“IF…THEN…”方式表达,两条逻辑性很强的知识之间却不能建立明确的联系。如例3.9中,“IF 汽车没油了THEN 汽车无法行驶”和“IF 汽车没油了 THEN 给汽车加油”明显是针对同一个事实的结论和操作,但在规则库中,它们却完全独立,相互没有联系。此外,对于传递性的知识和层次性的知识,产生式表示法也很难将其以自然的方式来表示。

3.2.3 语义网络表示法

产生式表示法不便于表示结构性知识，那么有没有善于表达结构知识的方法呢？本节介绍的语义网络表示法就很适合表达结构性知识。语义网络是奎利恩(J. R. Quillian)于1968年提出的一种心理学模型，后来奎利恩又把它用于知识表示。1972年，西蒙在自然语言理解系统中也采用了语义网络表示法。目前，语义网络已成为应用较多的一种知识表示方法。

1. 语义网络的概念

语义网络是一种用实体及其语义关系来表达知识的有向图。其中：节点代表实体，表示各种事物、概念、情况、属性、状态、事件、动作等；边代表语义关系，表示它所连接的两个实体之间的语义联系。在语义网络中，每一个节点和边都必须带有标志，这些标志用来说明它所代表的实体和语义。

在语义网络表示法中，最基本的单元称为语义基元，一个语义基元可用三元组(节点1，节点2，边)来描述。例如，若用实体1、实体2分别表示三元组中的节点1和节点2，用R表示实体1与实体2之间的语义联系，则它对应的语义基元如图3.3所示。

例3.10 用语义网络表示“西瓜是一种瓜果”。

由题意可知，该知识表达的是“西瓜”和“瓜果”之间的语义联系，这种语义联系为“是一种”表达了一种集合包含关系。因此，在语义网络中，边被标记为“是一种”，表达的是子类和父类的关系，如图3.4所示。

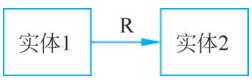

图3.3 语义网络中语义基元的结构

图3.4 “西瓜是一种瓜果”的语义网络表示

显然，语义网络可以表示“如果……则……”形式的产生式规则，只需要将规则前项和规则后项分别表示为两个实体，然后把边的语义联系表示为“逻辑结论”即可。

从功能上讲，语义网络可以描述任何事物间的任意关系。通常，这种描述需要把很多基本语义关系联系到一起来实现。常用的一些基本语义关系如下。

1) 实例关系

实例关系体现的是“具体与抽象”的概念，用来描述“一个事物是另外一个事物的具体例子”。其语义标志为ISA，即“is a”的简写形式，含义为“是一个”。

例3.11 用语义网络表示“Thomas Jefferson is a person”“孙悟空是一只猴子”。

解： 上述知识用语义网络表示如图3.5所示。

图3.5 “实例关系”的语义网络表示举例

2）分类关系

分类关系体现的是子类和父类的概念，也称为泛化关系，用来描述一个实体是另一个实体的成员，其语义标志为 AKO，即“a kind of”的缩写，其含义为“是一种”。例 3.10 就是典型的分类关系，其可重新表示为如图 3.6 所示。

图 3.6 “分类关系”的语义网络表示举例

3）成员关系

成员关系表达的是“个体与集体”的概念，用来描述“一个实体是另外一个实体中的成员”。其语义标志为 AMO（a member of），含义为“是一员”。

例 3.12 用语义网络表示“王华是共青团员”“独立团隶属于 115 师”。

解： 上述知识用语义网络表示如图 3.7 所示。

图 3.7 “成员关系”的语义网络表示举例

4）属性关系

属性关系是指实体与其行为、能力、状态、特征等属性之间的关系。由于不同事物的属性不同，因此属性关系可以有很多种。例如：

Have，含义是“有”，表示一个实体具有另一个实体所描述的属性。

Can，含义是“能”“会”，表示一个实体能做另一个实体所描述的事情。

Age，含义是“年龄”，表示一个实体是另一个实体在年龄方面的属性。

表示属性的关系不胜枚举，上述例子只是最常见的一些。

例 3.13 用语义网络表示“李红今年 21 岁”“这只猫的颜色是白色的，有尾巴，能够爬树”。

解： 上述知识用语义网络表示如图 3.8 所示。

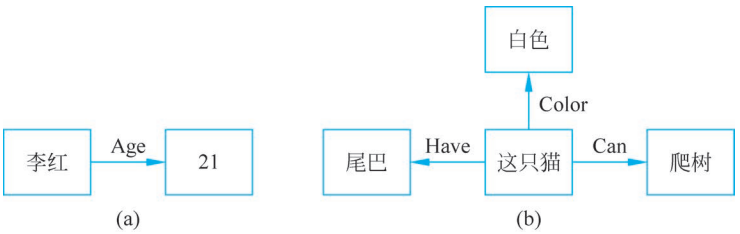

图 3.8 “属性关系”的语义网络表示举例

5）包含关系

包含关系也称为聚类关系，具有组织或结构特征的“部分与整体”之间的关系。常用的包含关系有 part-of，含义为“是一部分”，表示一个事物是另一个事物的一部分。

例 3.14 用语义网络表示“手掌是人体的一部分”“窗户是房屋的一部分”。

解： 上述知识用语义网络表示如图 3.9 所示。

图 3.9 “包含关系”的语义网络表示举例

6）时间关系

时间关系是常见的事件关联关系之一，其表示不同事件在发生时间上的先后关系。常用的时间关系有 Before 和 After。Before 的含义为“在前”，表示一个事件在另一个事件之前发生；After 的含义为“在后”，表示一个事件在另一个事件之后发生。

例 3.15 用语义网络表示“北京奥运会在伦敦奥运会之前”“香港回归后澳门也回归了”。

解： 上述知识用语义网络表示如图 3.10 所示。

图 3.10 “时间关系”的语义网络表示举例

7）空间关系

空间关系又称为位置关系，是指不同事物在空间位置方面的关系。常用的空间关系有：

Located-on，含义为“在上”，表示某一实体在另一实体之上。

Located-at，含义为“在”，表示某一实体所在的位置。

Located-under，含义为“在下”，表示某一实体在另一实体之下。

Located-inside，含义为“在内”，表示某一实体在另一实体之内。

Located-outside，含义为“在外”，表示某一实体在另一实体之外。

例 3.16 用语义网络表示“咖啡杯在桌子上”“国防科技大学位于长沙市”“牛顿在苹果树下”“游泳馆在体育中心内”“美国国会大厦不在伊利诺伊州，其位于华盛顿特区”。

解： 上述知识用语义网络表示如图 3.11 所示。

8）相近关系

相近关系是指不同事物在形状、内容、位置等方面相似或接近。常用的相近关系有：

Similar-to，表示某一事物与另一事物相似。

Near-to，表示某一事物与另一事物在空间位置上接近。

例 3.17 用语义网络表示“狗长得像狼”“电影院的旁边是糖果店”。

解： 上述知识用语义网络表示如图 3.12 所示。

9）因果关系

因果关系表示某件事情的发生而导致另一件事情发生的原因-结果关系，其类似于产生式表示法中的知识表示。其表示方式为 If-then，表示两个事物之间存在“如果……

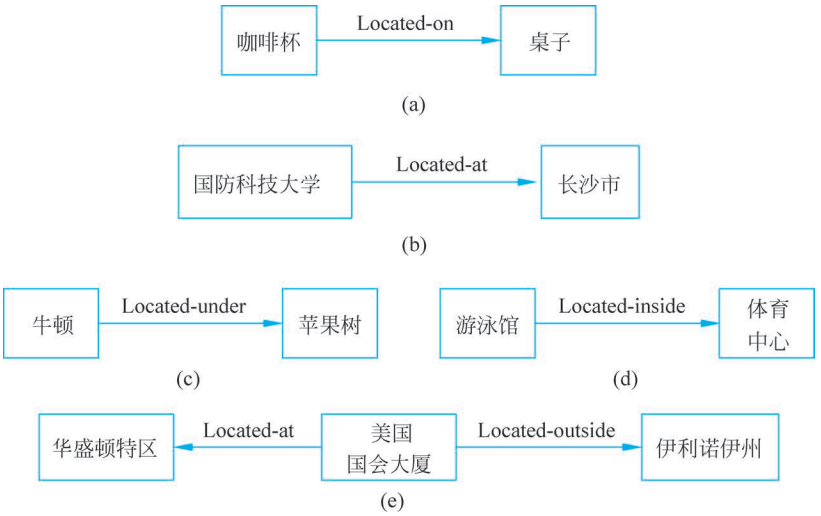

图 3.11 “空间关系”的语义网络表示举例

图 3.12 “相近关系”的语义网络表示举例

则……”的关系。

例 3.18 用语义网络表示“如果天气晴,我就去踢球”。

解:上述知识用语义网络表示如图 3.13 所示。

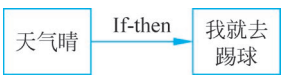

图 3.13 “因果关系”的语义网络表示举例

2. 知识的语义网络表示

1) 用语义网络表示一元关系

一元关系是指可以用一元谓词 $P(x)$ 表示的关系。其中,个体 x 为实体,谓词 P 说明实体的性质、属性等。一元关系描述的是一些最简单、最直观的事物或概念,常用“是”“有”“会”“能”等语义关系来说明。

从语义网络的结构看,其包括两个节点和一条有向边,描述的是两个节点之间的二元关系。那么,如何用它来描述一元关系呢?通常的做法是用节点 1 表示实体,用节点 2 表示实体的性质或属性等,用边表示节点 1 和节点 2 之间的语义关系。例如,实例关系(ISA)、属性关系(Have、Can、Age)等均是典型的一元关系表达。又如,用语义网络表示“Thomas Jefferson is a person”,则可以理解为“Thomas Jefferson”这个实体具有“person”的属性,表示方法见例 3.1。

例 3.19 用语义网络表示“动物能吃,能睡,能运动”。

解：上述知识用语义网络表示如图 3.14 所示。

在例 3.19 中，表示“动物”这个实体有“吃”、“睡”和“运动”的属性。可见，尽管语义网络描述的是两个节点之间的二元关系，但它同样可以方便地表示一元关系。

2）用语义网络表示二元关系

二元关系是指可用二元谓词 $P(x,y)$ 表示的关系，其中，个体 x 和 y 为实体，谓词 P 说明两个实体之间的关系。二元关系可以方便地用语义网络表示。分类关系、成员关系、包含关系、时间关系、空间关系、相近关系、因果关系等均是典型的二元关系。

有些关系看起来比较复杂，但可以较容易地分解成多个独立的二元关系或一元关系。对于这类问题，可先给出每个二元关系或一元关系的语义网络表示，再把它们关联到一起，得到问题的完整表示。

例 3.20 用语义网络表示“动物能吃，能睡，能运动”“狗是动物，有尾巴，能叫”“鱼是动物，有鳞，能游泳”。

解：上述知识用语义网络表示如图 3.15 所示。

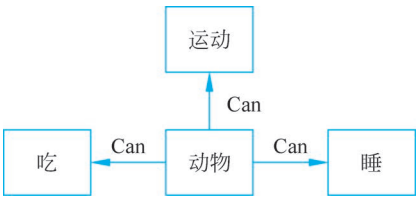

图 3.14 一元关系的语义网络表示举例

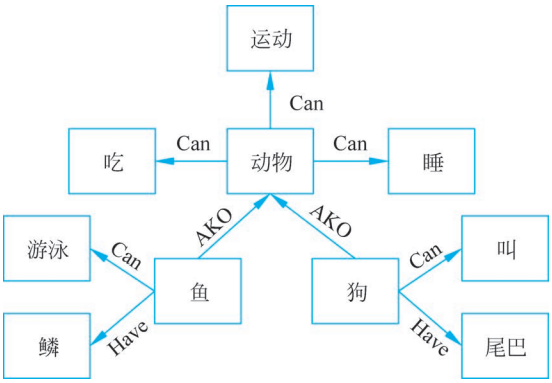

图 3.15 动物分类的语义网络

在例 3.20 中，“AKO”为二元关系，其余均为一元关系。多个一元关系和二元关系组合在一起构成了复杂的语义网络图，很好地表达了例 3.20 中的三个知识。

例 3.21 用语义网络表示“李韬是天河公司的设计师”“李韬是计算机学会成员，今年 32 岁”“天河公司是国企，位于德雅村”。

解：上述知识用语义网络表示如图 3.16 所示。

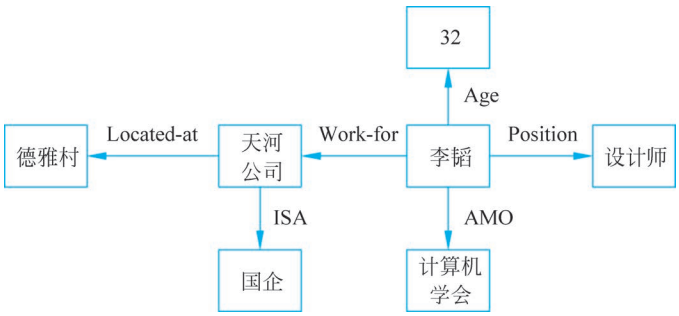

图 3.16 李韬相关信息的语义网络

例 3.22 用语义网络表示“熊伟的计算机是黑色，品牌是联想”“吴烨的计算机是银灰色，品牌是苹果”。

解： 在这个例子中，要表达的两条知识相对独立，但又有一定的联系。比如，两个人各自拥有一台计算机，但计算机作为抽象名词，可以看成是同一个概念。使该问题的表示更加一般化和便于扩充，可在语义网络中增加“计算机”这个抽象概念，并用计算机 1、计算机 2 分别代表熊伟和吴烨的计算机。因此，上述知识用语义网络表示如图 3.17 所示。

3）用语义网络表示多元关系

多元关系是指可用多元谓词 $P(x_1,x_2,\cdots,x_n)$ 表示的关系，其中，个体 $x_1,x_2,\cdots,x_n$ 为实体，谓词 P 说明这些实体之间的关系。在现实世界中，往往需要通过某种关系把多种事物联系起来，这就构成了一种多元关系。当用语义网络表示多元关系时，需要先将其转化为多个一元或二元关系，再把这些一元关系、二元关系组合起来实现对多元关系的表示。例 3.21 和例 3.22 均是由多元关系分解而成的多个一元、二元关系的组合。

3. *情况和动作的表示*

为了描述那些复杂的情况和动作，西蒙在他提出的表示方法中增加了情况节点和动作节点，即在语义网络中用一个节点来表示情况或动作。

1）情况（状态）的表示

用语义网络表示情况或状态时，需要设立一个情况节点。该节点有一组向外引出的边，用于指出各种可能的情况。

例 3.23 用语义网络表示“从去年夏天开始，张帆就拥有了自己的汽车”。

解： 在这个例子中，如果要强调“拥有”这样一个情况（状态），则可以把“所有权”设置为一个单独的节点。带“情况”节点的语义网络如图 3.18 所示。

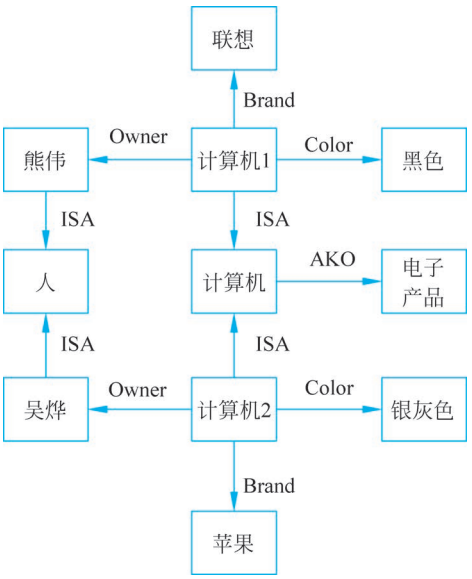

图 3.17 计算机相关信息的语义网络

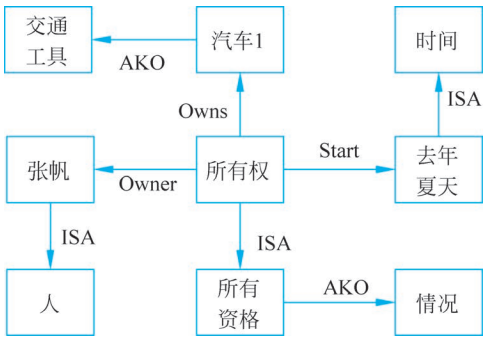

图 3.18 带“情况”节点的语义网络

这是一个相对复杂的语义网络，情况节点“所有权”位于该语义网络的中心，具有很多的属性。其“所有者”是张帆，“被拥有的物品”是汽车，然后用“去年夏天”这个节点表达了“所有权”这个情况的起始时间。

当然，如果并不强调“所有权”这个情况，“张帆拥有汽车”这样的知识也可以表示为如图 3.19 所示的语义网络。但值得注意的是，当没有“情况”节点时，难以表达张帆对汽车的“所有权”是从“去年夏天”这个时间开始的。

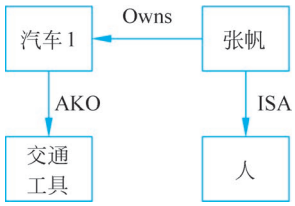

图 3.19　不带“情况”节点的语义网络

2）事件和动作的表示

与情况或状态时类似，用语义网络表示事件或动作时也需要建立一个单独的事件节点。事件节点也有一些向外引出的边，用于指出事件的主体和客体。

例 3.24　用语义网络表示“小王本学期一直担任学生会主席”。

解：在这个例子中，如果要强调“担任”这样一个动作，则可以把“担任”设置为一个单独的节点。带“动作”节点的语义网络如图 3.20 所示。

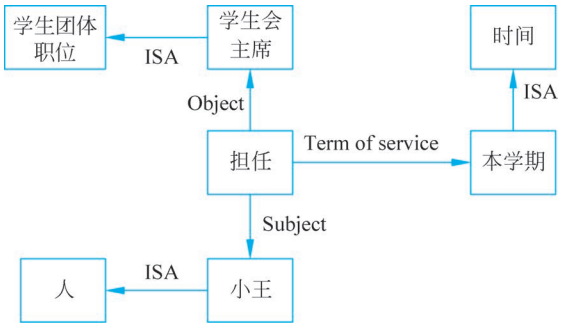

图 3.20　带“动作”节点的语义网络

图 3.20 中，动作节点“担任”的主体是“小王”，客体是“学生会主席”，代表小王“担任”了学生会主席。然后用“本学期”这个节点表达了“担任”这个情况的持续时间。

有时需要在语义网络中单独构建一个节点用以描述某一个事件。

例 3.25　用语义网络表示“王涵送给李敏一本书”。

解：在这个例子中，如果要强调“送给”这样一个动作，则可以构建带“动作”节点的语义网络，如图 3.21 所示。如果强调“送书”这个事件，则可以构建带“事件”节点的语义网络，如图 3.22 所示。

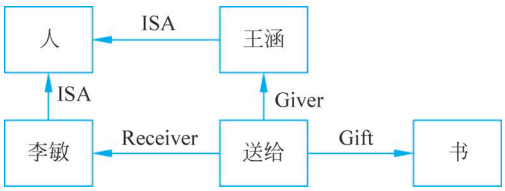

图 3.21　将“送给”设置为“动作”节点的语义网络

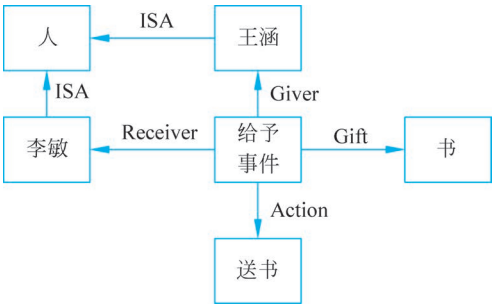

图 3.22　将“送书”设置为“事件”节点的语义网络

例 3.25 中，带“动作”节点的语义网络与带“事件”节点的语义网络表达的含义类似。但带“事件”节点的语义网络强调的是整个事件，动作“送书”是该事件的一个属性（表明书是“送”出去的，而不是“借”出去或是“卖”出去的）。

4. 语义网络的推理过程

语义网络的推理过程主要有继承和匹配两种。

继承是指把对事物的描述从抽象节点传递到具体节点的过程。通过继承可以得到当前节点更多的属性值，它通常是沿着 ISA、AKO 等边进行的。

匹配是指在语义网络中寻找与待求解问题相符的语义网络模式。

例 3.26　基于图 3.23 的语义网络，回答下列问题：

（1）“狗是否能运动？”

（2）“鱼是否能游泳？”

（3）“狗是否有鳞？”

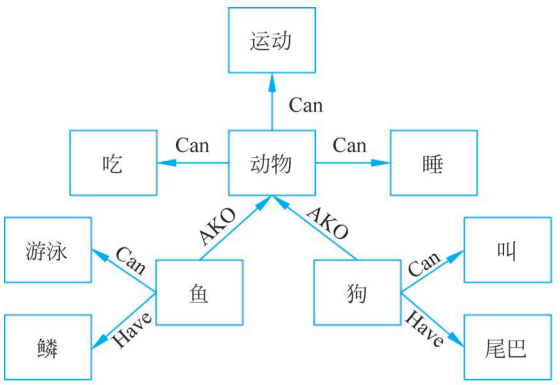

图 3.23　语义网络的推理示意

解：该语义网络与例 3.20 中的语义网络相同。从这个语义网络中可知，抽象节点“动物”是具象节点“鱼”和“狗”的统称，所以其由 AKO 边相连，那么“动物”所具有的“运动”能力可以被“鱼”和“狗”节点继承。于是，对于问题（1），回答是“狗能够运动”。对于问题（2），节点“鱼”与“游泳”节点通过 Can 边相连，表明“鱼”和“游泳”与问题中的“鱼是否能游泳”匹配，于是得到答案“鱼能游泳”。对于问题（3），由于节点“狗”和节点“鳞”没

有 Have 边相连，说明不能与问题匹配，因而得到答案“狗没有鳞”。继承和匹配就是语义网络知识应用的基本模式。

5. 语义网络表示法的特性

语义网络表示法的主要优点如下。

(1) 结构性。语义网络把事物的属性及事物间的各种语义联系显式地表示出来，是一种结构化的知识表示方法。在这种方法中具象节点可以继承抽象节点的属性，从而实现了信息的高效表达。

(2) 联想性。语义网络通过节点和边表达了事物之间的语义联系，体现出人类联想思维的过程。

(3) 自然性。语义网络实际上是一个在边上带有标志的有向图，可以直观地把知识表示出来，符合人们表达事物间关系的习惯，并且自然语言与语义网络之间的转换比较容易实现。

语义网络表示法也存在一定的缺点。

(1) 非严格性。语义网络没有像谓词那样严格的形式表示体系，一个给定语义网络的含义完全依赖处理程序对它进行的解释，通过语义网络实现的推理不能保证其正确性。

(2) 复杂性。语义网络表示知识的手段是多种多样的，虽然对其表示带来了灵活性，但由于表示形式的不一致，也使得对它的处理增加了复杂性。

3.3 确定性知识推理

3.2 节介绍了 3 种常用的知识表示方法。知识表示是为了应用，知识的应用是推理。本节主要针对确定性知识，讨论基于规则的产生式推理、基于标准逻辑的自然演绎推理及基于鲁滨逊归结原理的归结演绎推理。

3.3.1 产生式推理

通常，人们把利用产生式知识表示方法所进行的推理称为产生式推理，把由此所形成的系统称为产生式系统。产生式规则在 3.2.2 节中已经进行了介绍。产生式规则形式如：IF…THEN…的形式，如：

IF	the ‘fuel tank’ is empty
THEN	the car is dead

又如：

IF	the season is autumn
AND	the sky is cloudy
AND	the forecast is drizzle
THEN	the advice is ‘take an umbrella’

20 世纪 70 年代中期，纽厄尔和西蒙等设计了一个产生式系统模型，奠定了现代专家系统的基础。这个产生式模型基于如下思想：人类应用他们的知识(以产生式规则表达)解决一个特定的以特征信息为代表的问题。

典型的产生式系统基本结构如图 3.24 所示，包括综合数据库(Database)、知识库(Knowledge Base)和推理机(Inference Engine)三部分组成。

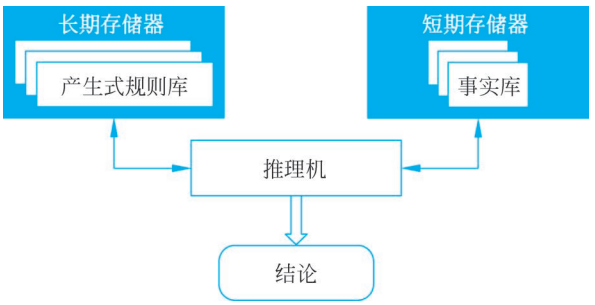

图 3.24　产生式系统基本结构

综合数据库也称为事实库，是用来存放与求解问题有关的各种当前信息，如问题的初始状态、输入的事实、推理得到的中间结论及最终结论等的数据结构。可见，数据库是存储用户输入事实及推理结论的部件，而事实和推理结论随着推理场景的变化而变化，因此这种存储是短期(Short-term Memory)的。

知识库又称为规则库，是用来存放与求解问题有关的所有产生式规则的集合，包含了将问题从初始状态转换成目标状态所需要的所有变换规则。这些产生式规则描述了问题领域中的一般性知识。可见，知识库是存储产生式规则集合的部件，而描述问题领域知识的产生式规则不会经常改变，因此这种存储是长期(Long-term Memory)的。

推理机又称为控制系统，由一组程序构成，用来控制整个产生式系统的推理过程。推理机主要进行以下五件事情。

(1) 选择匹配：按一定策略从知识库中选择规则(规则前项)与综合数据库中的已知事实进行匹配。

(2) 冲突消解：对匹配成功的规则，按照某种策略从中选出一条规则执行。

(3) 执行操作：对所执行的规则，若其后项为一个或多个结论，则把这些结论加入综合数据库；若其后项为一个或多个操作，则执行这些操作。

(4) 终止推理：检查综合数据库中是否包含有目标，若有，则停止推理。

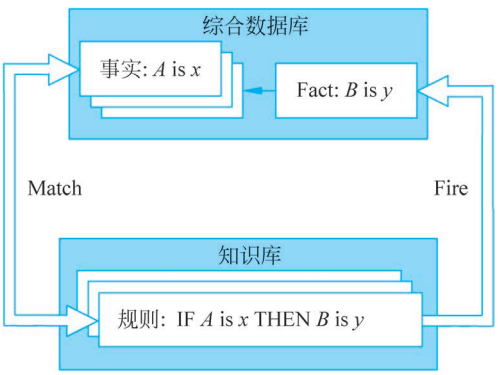

图 3.25　产生式系统基本推理过程

(5) 路径解释：在问题求解过程中记住应用过的规则序列，以便最终能够给出问题的解的路径。

在产生式系统基本推理过程中，当规则库中某条规则的前提可以和综合数据库中的已知事实相匹配时，该规则被激活，由它推出的结论将作为新的事实放入数据库，成为后面推理的已知事实，如图 3.25 所示。

现代专家系统可以看成产生式系统的

演变与发展，专家系统基本结构如图 3.26 所示。

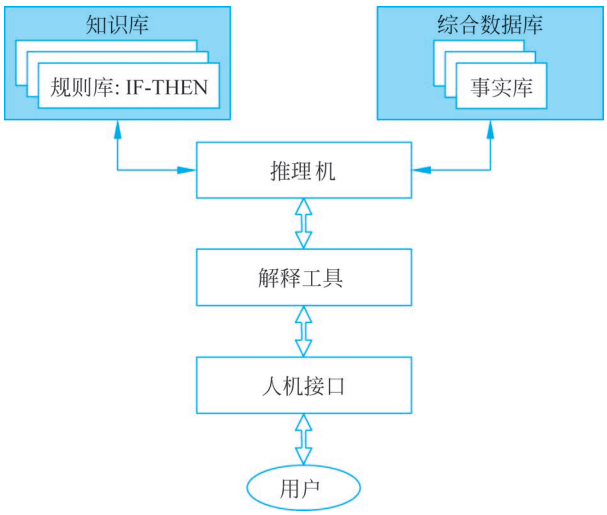

图 3.26　典型的专家系统基本结构

除了综合数据库、知识库和推理机以外，专家系统还包含解释工具（Explanation Facilities）、人机接口（User Interface）等部件。解释工具为用户提供推理机结论的相关解释。例如，诊断血液病的专家系统会向用户解释专家系统是如何判断该病人是患了某种血液病的推理过程。用户接口可以是图形界面的，也可以是问答系统等基于文字界面的，还可以是基于语音交互的，用于用户输入事实，并获得专家系统得出的结论和解释。

产生式推理具有一定的方向性，按照推理的控制方向，产生式推理可分为正向、逆向和混合 3 种方式。

1. 产生式的正向推理

正向推理是一种从已知事实出发正向使用推理规则的推理方式，也称为数据驱动推理或前向链推理。在正向推理中，用户需要事先提供一组初始证据，并将其放入综合数据库；推理开始后，推理机根据综合数据库中的已有事实到知识库中寻找当前可用规则，形成一个规则集合，然后按照冲突消解策略从该规则集合中选择一条知识进行推理，并将新推出的事实加入综合数据库，作为后面继续推理时可用的已知事实；如此重复这一过程，直到求出所需要的解或知识库中再无可用知识为止。

例 3.27　产生式系统正向推理过程。

如图 3.27 所示，初始条件下，综合数据库中有事实 A、B、C、D 和 E。而知识库中有 5 条规则。这时，开始正向推理。在正向推理过程中，所有能激发的规则都让其激发，于是事实 A 与规则 3 的前项匹配，规则 3 激发，生成新的事实 X，事实 C 与规则 4 的前项匹配，规则 4 激发，生成新的事实 L，这时没有规则可以被激发，正向推理第一轮完毕，将新生成的事实 X 和 L 存入综合数据库。由于事实库中同时存在 X、B、E 三个事实，规则 2 激发，生成新的事实 Y，这时没有规则可以被激发，正向推理第二轮完毕，将新生成的事实 Y 存入综合数据库。由于事实库中同时存在 Y、D 两个事实，规则 1 激发，生成新的事实

Z。这时没有规则可以被激发，正向推理第三轮完毕，将新生成的事实 Z 存入综合数据库。这时知识库中所有规则均不能再被激发，正向推理过程结束。

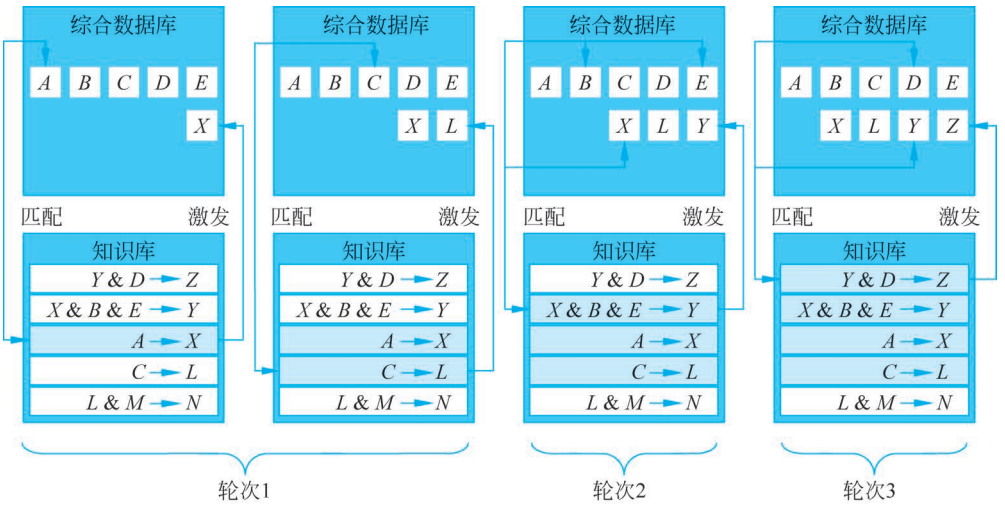

图 3.27　产生式系统正向推理举例

显然，在正向推理过程中并没有明确的推理目的，对推理将得出的结论也没有预知。在每一轮推理过程中能被激发的规则均被激发，从而得到新的事实，并存入综合数据库。如果需要的结论出现在综合数据库中，或者没有新的规则可以被激发，则推理结束。在正向推理过程中，某些规则的激发对于最终的推导结果没有帮助。一个真正的产生式系统（专家系统）可能有上百条规则，可能有很多被激发并产生了新的事实，但其与最终的目标毫无关系。如在例 3.27 中假设真正需要的结论是 Z，则规则 4 的激发对最终的结论没有支撑。

因此，如果要推理出一个明确的事实（求证某个结论），则正向推理技术的效率不太理想。

2. 产生式的逆向推理

逆向推理是一种以某个假设目标作为出发点的推理方法，也称为目标驱动推理或逆向链推理。与正向推理相比，逆向推理多了一个规则栈，暂时用于存放待激发的规则。在逆向推理中，产生式系统会设定一个目标，推理机试图找到证据来证明它。首先，推理机搜索知识库，寻找包含期望解决方案的规则，也就是说那些规则后项（“THEN”部分）包含目标结论的规则。如果找到这样的规则，而且它的 IF 部分和数据库中数据匹配，则激发该规则，目标得到证明，但这样的情况非常少。更多的情况是综合数据库中没有直接与该规则匹配的事实。这时产生式系统就对该规则进行压栈（存入规则栈中），并建立新的目标，即该被压栈规则的前项（“IF”部分），称为子目标，继续搜索知识库中能否证明该子目标的规则。推理机反复执行上述过程，直到目标被证明或知识库中没有可以证明子目标的规则为止。

例 3.28　产生式系统逆向推理过程（设定目标为证明事实“Z”成立），如图 3.28 所示。

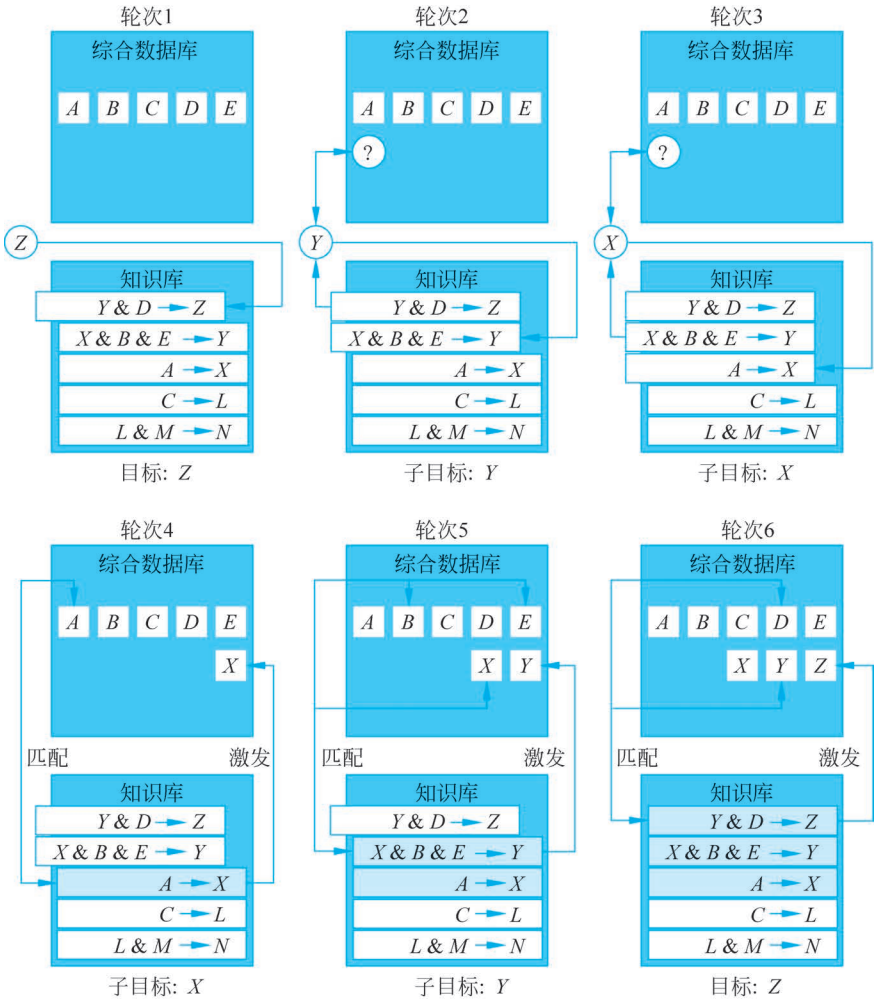

图 3.28　产生式系统逆向推理举例(设定目标为证明事实"Z"成立)

在例 3.28 中,推理机会首先搜索综合数据库和知识库,发现综合数据库中当前不包含事实 Z,但知识库中规则 1 的结论部分包含事实 Z。则推理机将规则 1 压栈,然后将规则 1 的前项 $Y\&D$ 作为子目标。然后,推理机会首先搜索综合数据库和知识库,发现事实 D 在综合数据库中,但事实 Y 不在,且规则 2 的规则后项包含事实 Y,则规则 2 压栈,且将规则 2 的前项 $X\&B\&E$ 作为新的子目标。接下来推理机发现在综合数据库中存在事实 B 和 E,但事实 X 不存在。但在知识库中,规则 3 的规则后项包含 X,则将规则 3 压栈,规则 3 的前项 A 作为新的子目标。接下来,推理机搜索综合数据库,发现事实 A 在事实库中。于是,规则 3 弹栈并激发,得到事实 X,并将 X 插入综合数据库中。规则 2 弹栈并激发,得到事实 Y,并将 Y 插入综合数据库中。规则 1 弹栈并激发,得到事实 Z,逆向推理过程成功退出。可见,逆向推理的目的性更强,可以避免激发对证明目标没有贡献的规则。例如,在例 3.27 中正向推理中被激发的规则 4 没有在逆向推理中被激发。

3. 产生式的混合推理

正向推理的特点是直观，无明确目标。可见，正向推理更适合于推理目标不明确，需要通过推理进行广泛“探索”的场景。但正向推理可能会激发很多与最终结论不相关的规则，导致其推理效率低。逆向推理从已知为真的事实出发，直接运用经明确的目标，通过“假设-证明”的思路从结论倒推出需要的条件，其效率较正向推理更高。在现实场景中，假设目标却非常难以选择。

由以上讨论可知，正向推理和逆向推理都有各自的适用场景和优缺点。当问题较复杂时，单独使用其中的哪一种都会影响到推理效率。为了更好地发挥这两种算法各自的长处，取长补短，可以将它们结合起来使用。这种把正向推理和逆向推理结合起来进行的推理称为混合推理。

混合推理有多种具体的实现方法，可以采用先正向推理后逆向推理的方法，也可以采用先逆向推理后正向推理的方法，还可以采用随机选择正向和逆向推理的方法。现代的专家系统中通常采用混合推理的方法。

3.3.2 自然演绎推理

自然演绎推理是从一组已知为真的事实出发，直接运用经典逻辑中的推理规则推出结论的过程。

1. 等价式

设 P 和 Q 是 D 上的两个谓词公式，若对 D 上的任意解释，P 和 Q 都有相同的真值，则称 P 和 Q 在 D 上是等价的。如果 D 是任意非空论域，则称 P 和 Q 是等价的，记作 $P \Leftrightarrow Q$。

谓词公式的一个解释是指对谓词公式中各个变元的一次真值指派，即指定各变元的真值为“真”或为“假”。

常用的等价式如下。

双重否定律：$\neg\neg P \Leftrightarrow P$

交换律：$(P \vee Q) \Leftrightarrow (Q \vee P)$，$(P \wedge Q) \Leftrightarrow (Q \wedge P)$

结合律：$(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$

$(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$

分配律：$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$

摩根定律：$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$

$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$

吸收律：$P \vee (P \wedge Q) \Leftrightarrow P$，$P \wedge (P \vee Q) \Leftrightarrow P$

补余律：$P \vee \neg P \Leftrightarrow T$，$P \wedge \neg P \Leftrightarrow F$

连词化归律：$P \rightarrow Q \Leftrightarrow \neg P \vee Q$

$P \leftrightarrow Q \Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P)$

$P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$

量词转换律：$\neg(\exists x)P \Leftrightarrow (\forall x)(\neg P)$

$\neg(\forall x)P \Leftrightarrow (\exists x)(\neg P)$

量词分配律：　　$(\forall x)(P \wedge Q) \Leftrightarrow (\forall x)P \wedge (\forall x)Q$

$(\exists x)(P \vee Q) \Leftrightarrow (\exists x)P \vee (\exists x)Q$

2. 永真蕴涵式

谓词公式 P 和 Q，如果 $P \rightarrow Q$ 永真，则称 P 永真蕴涵 Q，且称 Q 为 P 的逻辑结论，P 为 Q 的前提，记为 $P \Rightarrow Q$。

常用的永真蕴涵式如下。

化简式：　　$P \wedge Q \Rightarrow P, P \wedge Q \Rightarrow Q$

附加式：　　$P \Rightarrow (P \vee Q), Q \Rightarrow (P \vee Q)$

析取三段论：　　$\neg P, P \vee Q \Rightarrow Q$

假言推理：　　$P, P \rightarrow Q \Rightarrow Q$

拒取式：　　$\neg Q, P \rightarrow Q \Rightarrow \neg P$

假言三段论：　　$P \rightarrow Q, Q \rightarrow R \Rightarrow P \rightarrow R$

二难推理：　　$P \vee Q, P \rightarrow R, Q \rightarrow R \Rightarrow R$

全称固化：　　$(\forall x)P(x) \Rightarrow P(y)$

其中：y 是论域中的任一个体，依此可消去谓词公式中的全称量词。

存在固化：　　$(\exists x)P(x) \Rightarrow P(y)$

其中：y 是论域中某一个可以使 $P(y)$ 为真的个体，依此可消去谓词公式中的存在量词。

3. 置换与合一

1) 置换

在不同谓词公式中往往会出现多个谓词的谓词名相同但个体不同的情况，此时推理过程是不能直接进行匹配的。因此，需要先进行变元的替换。例如，对谓词公式

$$P(a) \quad 和 \quad P(x) \rightarrow Q(x)$$

式中，$P(a)$ 与 $P(x)$ 的谓词名相同，但个体不同，不能直接进行推理。首先需要找到项 a 对变元 x 的替换，使 $P(a)$ 和 $P(x)$ 不仅谓词名相同，而且个体相同。这种利用项对变元进行替换的操作称为置换(Substitution)。其形式化定义如下。

置换是形如 $\{t_1/x_1, t_2/x_2, \cdots, t_n/x_n\}$ 的有限集合，其中：$t_1, t_2, \cdots, t_n$ 是项；$x_1, x_2, \cdots, x_n$ 是互不相同的变元；$t_i/x_i (i=1,2,\cdots,n)$ 表示用 t_i 替换 x_i，并且要求 t_i 不能与 x_i 相同，x_i 不能循环地出现在另一个 t_i 中。

例如，$\{a/x, b/y, g(c)/z\}$ 是一个置换，但是 $\{f(x)/y, g(y)/x\}$ 不是一个置换，原因是它在 x 与 y 之间出现了循环置换现象。引入置换的目的本来是要将某些变元用其他变元、常量或函数来替换，使其不在公式中出现。但在 $\{f(x)/y, g(y)/x\}$ 中，$f(x)$ 用来置换 y，而 $g(y)$ 又用来替换 x，这样导致循环替换，既不能消去 x 又不能消去 y，因此它不是一个置换。

通常，置换可用希腊字母 θ、α、β、σ 等来表示。

例示：设 $\theta=\{t_1/x_1, t_2/x_2, \cdots, t_n/x_n\}$ 是一个置换，F 是一个谓词公式，把公式 F 中出现的所有 x_i 换成 $t_i (i=1,2,\cdots,n)$，得到一个新的公式 G，称 G 为 F 在置换 θ 下的例示，记作 $G=F \cdot \theta$。

一个谓词公式的任何例示都是该公式的逻辑结论。

2）合一

合一(Unifier)可以简单地理解为利用置换使两个或多个谓词的谓词名和个体均一致，以方便后续运算操作。其形式定义如下。

合一：设有公式集 $F=\{F_1,F_2,\cdots,F_n\}$，若存在一个置换 θ，使得 $F_1\cdot\theta=F_2\cdot\theta=\cdots=F_n\cdot\theta$，则称 θ 是 F 的一个合一，称 $F_1,F_2,\cdots,F_n$ 是可合一的。合一一般不唯一。

最一般合一：设 σ 是公式集 F 的一个合一，如果对任一个合一 θ 都存在一个代换 λ，使得 $\theta=\sigma\cdot\lambda$，则称 σ 是一个最一般合一。最一般合一是唯一的。

例 3.29 证明 $\theta=\{a/x,g(a)/y,f(g(a))/z\}$ 是公式集 $F=\{P(g(x),y,f(y)),P(g(a),g(x),z)\}$ 的一个合一。

证明：记 $F_1=P(g(x),y,f(y)),F_2=P(g(a),g(x),z)$

则

$$F_1\cdot\theta=P(g(a),g(a),f(g(a)))$$

且

$$F_2\cdot\theta=P(g(a),g(a),f(g(a)))$$

于是有 $F_1\cdot\theta=F_2\cdot\theta$，所以按合一的定义可知，$\theta$ 是公式集 F 的一个合一，且公式集 F 是可合一的。

4. 自然演绎推理的应用

下面将通过例子讨论如何使用自然演绎推理中的等价式、永真蕴含式及置换、合一方法完成从前提到逻辑结论的证明过程。

例 3.30 设已知如下事实：

$P,Q,Q\rightarrow R,P\rightarrow S,P\wedge R\rightarrow T,S\wedge T\rightarrow A$

求证：A 为真。

证明：$Q,Q\rightarrow R\Rightarrow R$ （假言推理）

$P,R\Rightarrow P\wedge R$ （引入合取词）

$P\wedge R,P\wedge R\rightarrow T\Rightarrow T$ （假言推理）

$P,P\rightarrow S\Rightarrow S$ （假言推理）

$S,T\Rightarrow S\wedge T$ （引入合取词）

$S,S\wedge T\rightarrow A\Rightarrow A$ （假言推理）

原命题得证。

例 3.31 设已知如下事实：

(1) 只要是能开花的绿色植物，小雪就喜欢。

(2) 花卉市场 A 卖的所有绿色植物都是能开花的。

(3) 三角梅在花卉市场 A 售卖。

求证：小雪喜欢三角梅。

证明：首先，定义谓词，即

Bloom(x)：x 是能开花的绿色植物。

Like(y,z)：y 喜欢 z。

Market_A(k)：k 在花卉市场 A 售卖。

依据题意，上述事实可用谓词逻辑表示如下：

$$(\forall x)(\text{Bloom}(x) \rightarrow \text{Like}(\text{Xue},x))$$

$$(\forall x)(\text{Market\_}A(x) \rightarrow \text{Bloom}(x))$$

$$\text{Market\_}A(\text{Tri\_Plum})$$

其中，x 的论域为所有绿色植物，Xue 表示小雪，Tri_Plum 表示三角梅。

利用自然演绎推理方法可得

$(\forall x)(\text{Market\_}A(x) \rightarrow \text{Bloom}(x)) \Rightarrow \text{Market\_}A(y) \rightarrow \text{Bloom}(y)$

（全称固化）

$\text{Market\_}A(\text{Tri\_Plum}), \text{Market\_}A(y) \rightarrow \text{Bloom}(y) \Rightarrow \text{Bloom}(\text{Tri\_Plum})$

（假言推理，置换{Tri_Plum/y}）

$(\forall x)(\text{Bloom}(x) \rightarrow \text{Like}(\text{Xue},x)) \Rightarrow \text{Bloom}(z) \rightarrow \text{Like}(\text{Xue},z)$

（全称固化）

$\text{Bloom}(\text{Tri\_Plum}), \text{Bloom}(z) \rightarrow \text{Like}(\text{Xue},z) \Rightarrow \text{Like}(\text{Xue},\text{Tri\_Plum})$

（假言推理，置换{Tri_Plum/z}）

由谓词定义可知，Like(Xue,Tri_Plum)表示小雪喜欢三角梅，原命题得证。

自然演绎推理的优点：定理证明过程自然，易于理解，有丰富的推理规则可用；推理过程符合人类的思维习惯，因此具有很好的可解释性。

自然演绎推理的主要缺点：在推理方向的选择上无明确指示，试探性和技巧性很强；在问题规模较大时容易产生组合爆炸；推理过程中得到的中间结论一般按指数规律递增。因此，自然演绎推理对于求解复杂问题的推理不利，甚至难以实现。

3.3.3 归结演绎推理

自然演绎推理方法在推理方向的选择上技巧性很强。如果每一个推理方向都探索，又容易导致组合爆炸。那么，有没有一种方法能够将逻辑推理的过程转换为逻辑计算的过程呢？逻辑计算并不需要太多技巧，只需要按照运算法则计算即可，因而更适合交给计算机完成。本节将介绍的归结演绎推理就是这样的方法。

归结演绎推理是一种基于鲁滨逊归结原理的机器推理技术。鲁滨逊归结原理也称为消解原理，是鲁滨逊于 1965 年提出的一种基于逻辑的“反证法”，其能够机械化地证明定理。归结演绎推理基本思想是把永真性的证明转化为不可满足性的证明，即证明 $P \rightarrow Q$ 永真，只要证明 $P \wedge \neg Q$ 为不可满足即可（$\neg(P \rightarrow Q) \Rightarrow \neg(\neg P \vee Q) \Rightarrow P \wedge \neg Q$，连词化归律，摩根定律，双重否定律）。

1. 永真性和不可满足性

为了后面推理需要，首先介绍谓词公式的永真性、可满足性与不可满足性的概念。

永真性：如果谓词公式 P 对非空论域 D 上的任一解释都取得真值 T，则称 P 在 D

上是永真的；如果 P 在任何非空论域上均是永真的，则称 P 永真。

由于谓词公式的一个解释是指对谓词公式中各个变元的一次真值指派，因此要利用该定义去判定一个谓词公式为永真，就必须枚举每个非空论域中的每个解释，并逐一进行判断。当解释的个数有限时，尽管枚举工作量大，公式的永真性毕竟还可以判定；当解释个数无限时，其永真性就很难通过枚举的方法判定。

可满足性：对于谓词公式 P，如果至少存在论域 D 上的一个解释，使公式 P 在此解释下的真值为 T，则称公式 P 在 D 上是可满足的。谓词公式的可满足性也称为相容性。

不可满足性：如果谓词公式 P 对非空论域 D 上的任一解释，都取真值 F，则称 P 在 D 上是不可满足的；如果 P 在任何非空论域上均是不可满足的，则称 P 不可满足。不可满足性又称为永假性或不相容性。

2. 谓词公式的范式

谓词公式的范式即是谓词公式的标准形式。在归结演绎推理中往往需要将谓词公式转换为与之等价的范式。根据量词在公式中出现的情况，可将谓词公式的范式分为以下两种。

1）前束范式

设 F 为一谓词公式，如果其中的所有量词均非否定地出现在公式的最前面，且它们的辖域为整个公式，则称 F 为前束范式。

前束范式的一般形式为

$$(Q_1x_1)\cdots(Q_2x_2)M(x_1,x_2,\cdots,x_n)$$

其中：$Q_i(i=1,2,\cdots,m)$为前缀，是一个由全称量词或存在量词组成的量词串；$M(x_1,x_2,\cdots,x_n)$为母式，是一个不包含任何量词的谓词公式。

2）Skolem 范式

如果前束范式中所有的存在量词都在全称量词之前，则称这种形式的谓词公式为 Skolem 范式。相对于前束范式，Skolem 范式提出了更严格的要求。

例 3.32 前束范式和 Skolem 范式举例。

$(\exists x)(\forall y)(\exists z)(P(x,y)\vee Q(z)\wedge R(y,z))$是前束范式，但不是 Skolem 范式。

$(\exists x)(\exists z)(\forall y)(P(x,y)\vee Q(z)\wedge R(y,z))$是前束范式，也是 Skolem 范式。

3. 子句集及其化简

采用归结演绎推理方法会将谓词公式转化与之对应的子句集，归结演绎推理是在子句集上进行的，因此在讨论归结演绎推理方法之前，介绍子句集的有关概念及化简方法。

1）子句及子句集

首先介绍子句和子句集的相关概念。

文字：原子谓词公式及其否定统称为文字。

例如：$P(x)$，$\neg P(x)$，$Q(x)$，$\neg Q(x)$均是文字。

子句：任何文字的析取式称为子句。

例如：$P(x)\vee Q(y)$，$R(x,y)\vee\neg S(y)\vee T(z)$都是子句。

空子句：不包含任何文字的子句称为空子句。由于空子句不含有任何文字，也就不能被任何解释所满足，因此空子句是永假的，不可满足的。空子句一般被记为□或 NIL。

子句集：由子句或空子句所构成的集合称为子句集。

2）子句集及其化简

任何一个谓词公式都可以通过应用等价关系及推理规则将其转化成相应的子句集。其化简步骤如下。

（1）消去连接词“→”和“↔”。

反复使用连词化归律

$$P \to Q \Leftrightarrow \neg P \vee Q \quad 及 \quad P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$$

即可消去谓词公式中的连接词“→”和“↔”。

如：$(\forall x)\big((\forall y)P(x,y) \to \neg(\forall y)(Q(x,y) \to R(y))\big)$

可等价转化为

$$(\forall x)\big(\neg(\forall y)P(x,y) \vee \neg(\forall y)(\neg Q(x,y) \vee R(y))\big)$$

（2）减少否定符号的辖域。

反复使用双重否定律

$$\neg\neg P \Leftrightarrow P$$

摩根定律

$$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q \text{ 及 } \neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$$

量词转换律

$$\neg(\exists x)P \Leftrightarrow (\forall x)(\neg P) \text{ 及 } \neg(\forall x)P \Leftrightarrow (\exists x)(\neg P)$$

将每个否定符号“¬”移到紧靠谓词的位置，使得每个否定符号最多只作用于一个谓词上。

则经步骤(2)化简后的谓词公式可转化为

$$(\forall x)\big((\exists y)\neg P(x,y) \vee (\exists y)(Q(x,y) \wedge \neg R(y))\big)$$

（3）对变元标准化。

变元标准化要求不同量词约束的变元有不同的名字。具体做法是在一个量词的辖域内把谓词公式中受该量词约束的变元全部用另一个没有出现过的任意变元代替。

经步骤(3)化简后的谓词公式可转化为

$$(\forall x)\big((\exists y)\neg P(x,y) \vee (\exists z)(Q(x,z) \wedge \neg R(z))\big)$$

（4）化为前束范式。

经过步骤(3)的变元标准化后，所有约束变元的名字都不同，因此可以把量词移动到谓词公式的最左边，从而形成前束范式。注意，在移动量词的过程中，不能改变量词的相对顺序。

经步骤(4)化简后的谓词公式可转化为

$$(\forall x)(\exists y)(\exists z)\big(\neg P(x,y) \vee (Q(x,z) \wedge \neg R(z))\big)$$

(5) 消去存在量词。

消去存在量词时,需要区分以下两种情况。

① 若存在量词不出现在全称量词的辖域内(它的左边没有全称量词),只要用一个新的个体常量替换受该存在量词约束的变元,就可消去该存在量词。

如:$(\exists y)(\exists z)\big(\neg P(x,y)\vee(Q(x,z)\wedge\neg R(z))\big)$通过消去存在量词,可转换为

$$\neg P(x,a)\vee(Q(y,b)\wedge\neg R(b))$$

② 若存在量词位于一个或多个全称量词的辖域内,即形如:

$$(\forall x_1)\cdots(\forall x_n)(\exists y)P(x_1,\cdots,x_n,y)$$

则不能直接消去存在量词,而需要用 Skolem 函数 $g(x_1,\cdots,x_n)$替换受该存在量词约束的变元,再消去该存在量词。下面给出 Skolem 函数的定义。

如果存在量词在全称量词的辖域内,如$(\forall x)(\exists y)P(x,y)$变量 y 的取值依赖于变量 x 的取值,令这种依赖关系由函数 $g(x)$来表示,它把每个 x 值映像到存在的那个 y,这个函数称为 Skolem 函数。

经步骤(4)化简后的谓词公式要消去存在量词时,需要用到 Skolem 函数,其可转换为

$$(\forall x)\big(\neg P(x,f(x))\vee\big(Q(x,g(x))\wedge\neg R(g(x))\big)\big)$$

(6) 化为 Skolem 标准形。

Skolem 标准型的一般形式是$(\forall x_1)\cdots(\forall x_n)M(x_1,\cdots,x_n)$,其中,$M(x_1,\cdots,x_n)$是 Skolem 标准形的母式,由子句的合取构成。

对步骤(5)已经去掉存在量词的前束范式的母式应用分配律

$$P\vee(Q\wedge R)\Leftrightarrow(P\vee Q)\wedge(P\vee R)$$

可以得到 Skolem 标准形的谓词公式。

将上述经步骤(5)化简后的谓词公式示例化为 Skolem 标准形,可得

$$(\forall x)\Big(\big(\neg P(x,f(x))\vee Q(x,g(x))\big)\wedge\big(\neg P(x,f(x))\vee\neg R(g(x))\big)\Big)$$

(7) 消去全称量词。

谓词公式化为 Skolem 标准形后,其母式中的全部变元均受全称量词的约束,并且全称量词的次序已无关紧要,因此可以省去全称量词。但认为剩下的母式的变元仍是被全称量词量化的。

例如,经步骤(6)化为 Skolem 标准形的谓词公式示例,去掉全称量词后得

$$\big(\neg P(x,f(x))\vee Q(x,g(x))\big)\wedge\big(\neg P(x,f(x))\vee\neg R(g(x))\big)$$

(8) 消去合取词。

去掉步骤(7)化简后的谓词公式中的所有合取词,把母式用子句集的形式表示出来。这里的表示也只是认为换一种表示形式,子句集中的子句仍然存在合取关系。其中,子句集中的每个元素就是一个子句。

例如，步骤(8)所得公式的子句集中包含以下两个子句：

$$\neg P(x,f(x)) \vee Q(x,g(x))$$

$$\neg P(x,f(x)) \vee \neg R(g(x))$$

(9) 更换变元名称。

对子句集中的某些变元重新命名，使任意两个子句中不出现相同的变元名。由于每个子句都对应着谓词公式中的一个合取元，并且所有变元都由全称量词量化的，因此任意两个不同子句的变元之间实际上不存在任何关系。这样，更换变元名是不会影响公式的真值的。

例如，对步骤(8)所得公式，可把第二个子句集中的变元名 x 更换为 y，得到如下子句集：

$$\neg P(x,f(x)) \vee Q(x,g(x))$$

$$\neg P(y,f(y)) \vee \neg R(g(y))$$

3) 子句集的应用

通过子句集化简的 9 个步骤，可以将任何谓词公式转化为与之相对应的标准子句集。由于在消去存在量词时所用的 Skolem 函数可以不同，因此化简后的标准子句集是不唯一的。从形式上看，以子句集的方式表达的谓词公式更简单、更明确。那么原始谓词公式和与之对应的子句集有何联系呢？简单地说，当原谓词公式为不可满足(永假)时，其标准子句集则一定是不可满足的，即 Skolem 化并不影响原谓词公式的永假性。这个结论很重要，是归结原理的主要依据，可用定理的形式来描述。

定理 3.1 设有谓词公式 F，其标准子句集为 S，则 F 为不可满足的充要条件是 S 为不可满足的。

上述定理的证明思路是证明从谓词公式到标准子句集的 9 个转化步骤均不改变原谓词公式的不可满足性。限于篇幅，具体的证明步骤本书不做介绍。

由定理 3.1 可知，要证明一个谓词公式是不可满足的，只要证明其相应的标准子句集是不可满足的。证明一个子句集的不可满足性可由“鲁滨逊归结原理”来解决。

4. 鲁滨逊归结原理

鲁滨逊归结原理是通过对子句集中的子句做逐次归结来证明子句集的不可满足性的，是对定理自动证明的一个重大突破。

1) 基本思想

由谓词公式转换为子句集的方法可知，在子句集中子句之间是合取关系。只要有一个子句不可满足，整个子句集就是不可满足的。此外，前面已经指出空子句是不可满足的。因此，一个子句集中如果包含空子句，此子句集就一定是不可满足的。

鲁滨逊归结原理就是基于上述认识提出的，其基本思想如下。

(1) 将所有的已知条件化简为子句集 S_1。

(2) 将欲证明的逻辑结论否定，并化简为子句集 S_2。

(3) 将 S_1 和 S_2 合并，得到扩展的子句集 S，然后设法检验子句集 S 是否含有空子句，若含有空子句，则表明 S 是不可满足的；否则，转步骤(4)。

(4) 若扩展的子句集 S 中不含有空子句，则继续使用归结法，在子句集中选择合适的子句进行归结，直至导出空子句或不能继续归结为止。

如果能够通过归结方法在子句集中导出空子句，则说明子句集是不可满足的，进而说明逻辑结论的否定是不可满足的，从而证明了逻辑结论的永真性。

鲁滨逊归结原理可分为命题逻辑归结原理和谓词逻辑归结原理。下面将分别进行介绍。

2) 命题逻辑归结原理

(1) 命题逻辑的归结式。

首先给出归结式的相关定义。

互补文字：若 P 是原子谓词公式，则称 P 与 $\neg P$ 为互补文字。

命题逻辑归结、归结式与亲本子句：设 C_1 和 C_2 是子句集中的任意两个子句，如果 C_1 中的文字 L_1 与 C_2 中的文字 L_2 互补，那么可从 C_1 和 C_2 中分别消去 L_1 和 L_2，并将 C_1 和 C_2 中余下的部分按**析取关系**构成一个新的子句 C_{12}，则称这一过程为命题逻辑归结，称 C_{12} 为 C_1 和 C_2 的归结式，称 C_1 和 C_2 为 C_{12} 的亲本子句。

例 3.33 设 $C_1=P\vee Q$，$C_2=\neg P\vee R\vee S$，求 C_1 和 C_2 的归结式 C_{12}。

解：由归结的定义可知，互补的文字是 $L_1=P$ 和 $L_2=\neg P$。在归结式中消去互补文字，并按析取关系构成新的子句得 $C_{12}=Q\vee R\vee S$。

例 3.34 设 $C_1=P$，$C_2=\neg P$，求 C_1 和 C_2 的归结式 C_{12}。

解：互补的文字是 $L_1=P$ 和 $L_2=\neg P$，则根据归结过程得到归结式为 $C_{12}=\mathrm{NIL}$。说明 C_1 和 C_2 的归结式为空子句。可见 C_1 和 C_2 组成的子句集为不可满足的。

例 3.35 设 $C_1=P\vee\neg Q$，$C_2=\neg P$，$C_3=Q$，求 C_1、C_2 和 C_3 的归结式 C_{123}。

解：需求解三个子句的归结结果，先对其中两个子句进行归结，再将归结结果和第三个子句进行归结。先对 C_1 和 C_2 归结可以得到 $C_{12}=\neg Q$，再对 C_{12} 与 C_3 归结得到 $C_{123}=\mathrm{NIL}$。也可以先对 C_1 和 C_3 归结得到 $C_{13}=P$，再对 C_{13} 与 C_2 归结同样得到 $C_{123}=\mathrm{NIL}$。可见，对子句集进行归结的顺序并不影响归结结果。子句集的归结顺序可以不唯一，结果却是唯一的。归结过程表示为如图 3.29 所示的树状结构，称为子句集的归结树。

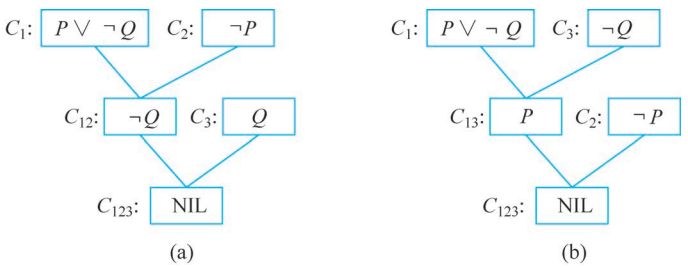

图 3.29 归结过程的树状结构表达

(2) 命题逻辑归结式的性质。

对命题逻辑归结式的性质可用如下定理、推论描述。

定理 3.2 归结式 C_{12} 是其亲本子句 C_1 和 C_2 的逻辑结论。

证明：由于 C_1 和 C_2 是 C_{12} 的亲本子句，则可设 $C_1=P\vee C_1'$，$C_2=\neg P\vee C_2'$，且 C_1 和 C_2 关于解释 I 为真。则只需证明 $C_{12}=C_1'\vee C_2'$ 关于解释 I 也为真。

显然，对于解释 I，P 和 $\neg P$ 中必有一个为真，一个为假。

若 P 为假，则必有 C_1' 为真，不然会使 C_1 为假，这与前提假 C_1 为真矛盾，因此只能有 C_1' 为真。

同理，若 $\neg P$ 为假，则必有 C_2' 为真。

因此，必有 $C_{12}=C_1'\vee C_2'$ 关于解释 I 为真。

即 C_{12} 是其亲本子句 C_1 和 C_2 的逻辑结论。原命题得证。

这个定理是归结原理中很重要的一个定理，由它可得到以下两个推论。

推论 3.1 设 C_1 和 C_2 是子句集 S 中的两个子句，C_{12} 是 C_1 和 C_2 的归结式，若用 C_{12} 代替 C_1 和 C_2 后得到新的子句集 S_1，则由 S_1 的不可满足性可以推出原子句集 S 的不可满足性。即 S_1 是不可满足的$\Rightarrow S$ 是不可满足的。

推论 3.2 设 C_1 和 C_2 是子句集 S 中的两个子句，C_{12} 是 C_1 和 C_2 的归结式，若把 C_{12} 加入 S 中，得到新的子句集 S_2，则 S_2 和原子句集 S 的不可满足性是等价的。即 S_2 是不可满足的$\Leftrightarrow S$ 是不可满足的。

推论 3.1 和推论 3.2 的证明可利用不可满足性的定义和归结过程来完成，证明过程类似定理 3.2，本书从略。

上述两个推论告诉我们，为证明子句集 S 的不可满足性，只要对其中可进行归结的子句进行归结，并把归结式加入子句集 S 中，或者用归结式来代替它的亲本子句，然后对新的子句集证明其不可满足性就可以。如果通过归结能得到空子句，根据空子句的不可满足性，即可得到原子句集 S 是不可满足的结论。

在命题逻辑中，这种不可满足性的推理过程可用如下定理描述。

定理 3.3 子句集 S 是不可满足的，当且仅当存在一个从 S 到空子句的归结过程。

上述定理的证明需要用到海伯伦原理。本书限于篇幅，这里证明从略。需要指出，鲁滨逊归结原理仅对不可满足的子句集 S 适用，对可满足的子句集 S 是得不出任何结果的。

3）谓词逻辑归结原理

在谓词逻辑中，子句集中的谓词通常含有变元，因此不能像命题逻辑那样直接消去互补文字，而需要首先通过置换操作对待归结的两个谓词公式的互补文字进行合一后才能归结。

（1）谓词逻辑的归结式。

与命题逻辑类似，在谓词逻辑中也需要先定义归结的概念。

谓词逻辑归结、归结式与亲本子句：设 C_1 和 C_2 是子句集中没有公共变元的任意两个子句，L_1 和 L_2 分别是 C_1 和 C_2 中文字。如果 L_1 和 L_2 存在最一般合一 σ，使得 $L_1\sigma$ 与 $L_2\sigma$ 互补，则称

$$C_{12}=(\{C_1\sigma\}-\{L_1\sigma\})\cup(\{C_2\sigma\}-\{L_2\sigma\})$$

为 C_1 和 C_2 的归结式，称 C_1 和 C_2 为 C_{12} 的亲本子句，称这一过程为谓词逻辑的归结。

这里使用集合符号和集合的运算是为了说明问题方便。即先将子句 $C_i\sigma$ 和 $L_i\sigma$ 写成集合的形式，并在集合表示下做减法和并集运算，再写成子句集的形式。

可见，谓词逻辑归结的定义与命题逻辑归结极为类似，其区别如下。

第一，谓词逻辑中通常变元的名字不同，需要用合一将互补文字的变元统一后再进行消解。

第二，定义中要求 C_1 和 C_2 无公共变元。这是因为如果 C_1 和 C_2 存在公共变元，可能导致无法合一，从而不能消去互补文字。其实，在归结过程中 C_1 和 C_2 不能存在公共变元，已经在原始谓词公式化简为子句集的步骤(9)中保证了。因为步骤(9)的工作就是更换子句集中不同子句的变元名称。

例 3.36 设 $C_1=P(a)\vee Q(x)$，$C_2=\neg P(y)\vee R(y)\vee S(z)$，求 C_1 和 C_2 的归结式 C_{12}。

解： 合一后，可能互补的文字是 $L_1=P(a)$ 和 $L_2=\neg P(y)$。取 L_1 和 L_2 的合一 $\sigma=\{a/y\}$，则 $C_1\sigma=P(a)\vee Q(x)$，$C_2\sigma=\neg P(a)\vee R(a)\vee S(z)$。在归结式中消去互补文字，并按析取关系构成新的子句得 $C_{12}=Q(x)\vee R(a)\vee S(z)$。

例 3.37 设 $C_1=P(x)\vee P(f(a))$，$C_2=\neg P(y)\vee Q(y)\vee R(a)$，求 C_1 和 C_2 的归结式 C_{12}。

解： 在 C_1 中两个文字的谓词名均是 P，但变元不同。应首先在 C_1 内部进行一次合一，化简后再与 C_2 做归结。

取 $\sigma_1=\{f(a)/x\}$，则 $C_1\sigma_1=P(f(a))\vee P(f(a))=P(f(a))$。

显然，$L_1=P(f(a))$ 和 $L_2=\neg P(y)$，取 L_1 和 L_2 的合一 $\sigma_2=\{f(a)/y\}$，则 $C_1\sigma_1\sigma_2=P(f(a))$，$C_2\sigma_2=\neg P(f(a))\vee Q(f(a))\vee R(a)$。在归结式中消去互补文字，并按析取关系构成新的子句得 $C_{12}=Q(f(a))\vee R(a)$。

在例 3.37 中，对参加归结的某个子句，若其内部有可合一的文字，则在进行归结之前应先对这些文字进行合一。$C_1\sigma_1$ 称为 C_1 的因子。本应是 C_1 与 C_2 进行归结，但使用 σ_1 做完合一后变成了 C_1 的因子 $C_1\sigma_1$ 与 C_2 进行归结。此时，谓词逻辑归结的定义仍然适用，$C_1\sigma_1$ 和 C_2 的归结式依然记为 C_{12}。

更一般的情况是，对于两个谓词逻辑归结中的子句 C_1 和 C_2，C_1 与 C_2 的归结式、$C_1\sigma_1$ 与 C_2 的归结式、C_1 与 $C_2\sigma_2$ 的归结式以及 $C_1\sigma_1$ 与 $C_2\sigma_2$ 的归结式均可称为子句 C_1 和 C_2 的二元归结式，记为 C_{12}。

这里还需要强调几种不能归结的情况。

① 谓词不一致不能归结。如 $P(y)$ 与 $\neg Q(y)$ 不能归结。

② 常量之间不能归结。如 $Q(a)$ 与 $\neg Q(b)$ 不能通过置换 $\{a/b\}$ 完成归结。变量可以置换为常量完成归结，但常量之间不能置换。因为 $Q(a)$ 关于解释 I 的真值并不能保证与 $Q(b)$ 的真值相同。

③ 变量与其函数之间不能归结。如 $P(x)$ 与 $\neg P(f(x))$ 不同通过置换 $\{f(x)/x\}$ 完

成归结。道理与②相同。

④ 不能同时消去两个互补对，这可能导致结论不正确，如例 3.38。

例 3.38 设 $C_1=P(a)\vee Q(b)$，$C_2=\neg P(a)\vee \neg Q(b)$，求 C_1 和 C_2 的归结式 C_{12}。

解： C_1 与 C_2 是中存在两个互补对（$P(a)$ 与 $\neg P(a)$、$Q(b)$ 与 $\neg Q(b)$），若同时消去两个互补对，得到 NIL，说明 C_1 与 C_2 矛盾。实际上，从谓词公式的逻辑含义上看，C_1 与 C_2 并不矛盾。所以，C_1 与 C_2 不能归结，NIL 不是 C_1 和 C_2 的二元归结式。

（2）谓词逻辑归结式的性质。

对谓词逻辑，定理 3.2、推论 3.1、推论 3.2 和定理 3.3 仍然适用，这里不再赘述。若通过谓词逻辑归结法将子句集归结出空子句，则可证明原谓词公式否定的不可满足性。

5. *归结演绎推理的方法*

无论是对于命题逻辑还是对于谓词逻辑，归结演绎推理的方法都极为类似。归结演绎推理是一种命题自动证明方法，其基本思想是反证：将要证明的结论加否定，并证明其不可满足性。具体方法如下。

假设 F 为已知的前提条件，G 为欲证明的结论，且 F 和 G 都是公式集的形式。根据反证法“G 为 F 的逻辑结论，当且仅当 $F\wedge\neg G$ 是不可满足的”，可把已知 F 证明 G 为真的问题转化为证明 $F\wedge\neg G$ 为不可满足的问题。再根据定理 3.1 和定理 3.2，在不可满足的意义上公式集 $F\wedge\neg G$ 与其子句集是等价的，又可把 $F\wedge\neg G$ 在公式集上的不可满足问题转化为子句集上的不可满足问题。这样就可用归结原理进行定理的证明。

应用归结原理证明定理的过程称为归结反演。已知 F，证明 G 为真的归结反演过程如下。

（1）否定目标公式 G，得 $\neg G$。

（2）把 $\neg G$ 并入公式集 F 中，得到 $\{F,\neg G\}$。

（3）把 $\{F,\neg G\}$ 化为子句集 S。

（4）应用归结原理对子句集 S 中的子句进行归结，并把每次得到的归结式并入 S 中。如此反复，若出现空子句，则停止归结，此时就证明了 G 为真。

下面举例说明归结反演过程。

例 3.39 设已知的公式集为 $\{P,(P\wedge Q)\rightarrow R,(S\vee T)\rightarrow Q,T\}$，求证结论 R。

解： 假设结论 R 为假，将 $\neg R$ 加入公式集，并化为子句集，得到

$$S=\{P,\neg P\vee\neg Q\vee R,\neg S\vee Q,\neg T\vee Q,T,\neg R\}$$

归结过程的树状表达如图 3.30 所示。

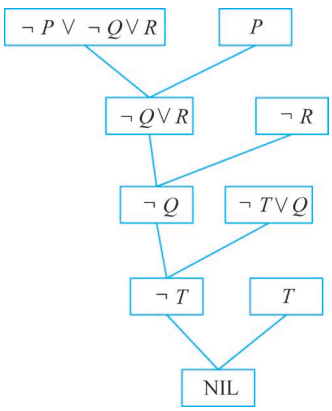

图 3.30 归结过程的树状表达

在该树中，由于根部出现空子句，因此命题 R 得到证明。

这个归结证明过程为：开始假设子句集 S 中的所有子句均为真，即原公式集为真，$\neg R$ 也为真；然后利用归结原理，对子句集中含有互补文字的子句进行归结，并把

所得的归结式并入子句集中；重复这一过程，最后归结出了空子句。于是，可知子句集 S 是不可满足的，即开始时假设 $\neg R$ 为真是错误的，这就证明了 R 为真。

例 3.40 已知：

$$F:(\forall x)\big((\exists y)(A(x,y)\wedge B(y))\rightarrow(\exists z)(C(z)\wedge D(x,z))\big)$$

$$G:\neg(\exists x)C(x)\rightarrow(\forall x)(\forall y)(A(x,y)\rightarrow\neg B(y))$$

求证：G 是 F 的逻辑结论。

证明：先把 G 否定并放入 F 中，得到$\{F,\neg G\}$；再将$\{F,\neg G\}$化简为子句集，得到

① $\neg A(x,y)\vee\neg B(y)\vee C(g(x))$

② $\neg A(u,v)\vee\neg B(v)\vee D(u,g(u))$

③ $\neg C(k)$

④ $A(m,n)$

⑤ $B(j)$

其中，子句①和②由 F 化简得到，③、④和⑤由 $\neg G$ 化简得到。归结过程如图 3.31 所示。

因此，G 是 F 的逻辑结论。

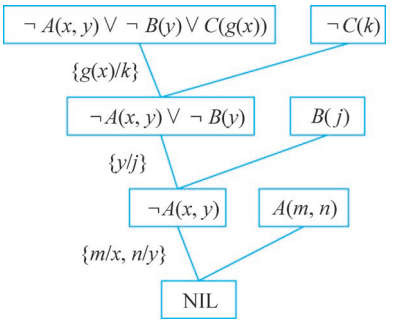

图 3.31 归结过程的树状表达

例 3.41 快乐学生问题。设任何能通过毕业考试且能获奖的人都是快乐的；任何认真学习或运气好的人都能通过毕业考试；小云不认真学习，但运气好；任何运气好的人都能获奖。求证小云是快乐的。

证明：定义谓词，即

Pass(x)：x 能通过毕业考试。

Win(x)：x 能获奖。

Lucky(x)：x 是幸运的。

Study(x)：x 是认真学习的。

Happy(x)：x 是快乐的。

基于上述谓词，由题意可知，前提条件可表达为

① $(\forall x)\big((\text{Pass}(x)\wedge\text{Win}(x))\rightarrow\text{Happy}(x)\big)$

② $(\forall x)\big((\text{Study}(x)\vee\text{Lucky}(x))\rightarrow\text{Pass}(x)\big)$

③ $\neg\text{Study}(\text{Yun})\wedge\text{Lucky}(\text{Yun})$

④ $(\forall x)(\text{Lucky}(x)\rightarrow\text{Win}(x))$

结论可表达为

$$\text{Happy}(\text{Yun})$$

按照归结原理的证明思路，将结论否定，加入前提条件公式集合中，并化简为子句集，可得

⑤ $\neg\text{Pass}(x)\vee\neg\text{Win}(x)\vee\text{Happy}(x)$

⑥ ¬Study(y) ∨ Pass(y)

⑦ ¬Lucky(z) ∨ Pass(z)

⑧ ¬Study(Yun)

⑨ Lucky(Yun)

⑩ ¬Lucky(k) ∨ Win(k)

结论的否定：¬Happy(Yun)

其中，⑤由①化简得出；⑥和⑦由②化简得出；⑧和⑨由③化简得出；⑩由④化简得出；再加上结论的否定，共同构成了子句集。该子句集归结过程如图 3.32 所示。

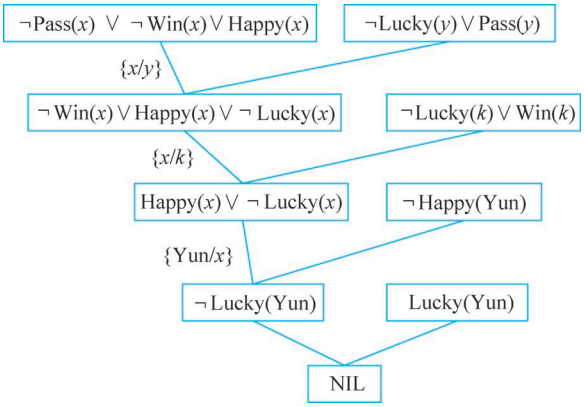

图 3.32　归结过程的树状表达

由于归结结果得到了空子句，于是原命题得证。

从例 3.41 可知，对于实际生活中的逻辑推理问题，可以先由谓词逻辑法进行知识表示，然后应用归结演绎推理即可证明逻辑结论。这样的问题求解思想是人工智能符号主义学派的主要思路，其基本假设是符号运算能够模拟人类智能。

归结演绎推理是不是只能在逻辑结论已知的情况下对其完成证明，而不能在逻辑结论未知的条件下推导出逻辑结论呢？其实，归结演绎推理不仅仅只能完成逻辑结论的证明。

例 3.42　上课教室问题。已知：信息工程专业的学生在 301 教室或 302 教室上课；张帆和李振都是信息工程专业的学生；张帆和李振在不同教室上课；张帆在 302 教室上课。试问李振在哪个教室上课？

解：用归结演绎推理对其进行求解。

首先，定义谓词如下。

Inf_Stu(x)：x 是信息工程专业的学生。

At(y,z)：y 在 z 上课，其中，y 的论域是学生集合，z 的论域是教室集合。

基于上述谓词，由题意可知，前提条件可表达为

① $(\forall x)$(Inf_Stu(x)→At(x,301) ∨ At(x,302))

② Inf_Stu(Zhang)

③ Inf_Stu(Li)

④ $At(Zhang,z)\rightarrow \neg At(Li,z)$

⑤ $At(Zhang,302)$

将前提条件的谓词公式化简为子句集，可得

⑥ $\neg Inf\_Stu(x) \vee At(x,301) \vee At(x,302)$

⑦ $Inf\_Stu(Zhang)$

⑧ $Inf\_Stu(Li)$

⑨ $\neg At(Zhang,z) \vee \neg At(Li,z)$

⑩ $At(Zhang,302)$

其中，⑥由①化简得到；⑦由②化简得到；⑧由③化简得到；⑨由④化简得到；⑩由⑤化简得到。

由于本题是求解而非证明，则把目标的否定化成子句式，并用重言式表达，得到

$$\neg At(Li,v) \vee At(Li,v)$$

重言式没有实际意义，表示李振要么在教室 v，要么不在。显然，这是一个永真的命题。把重言式加入子句集中，该子句集归结过程如图 3.33 所示。

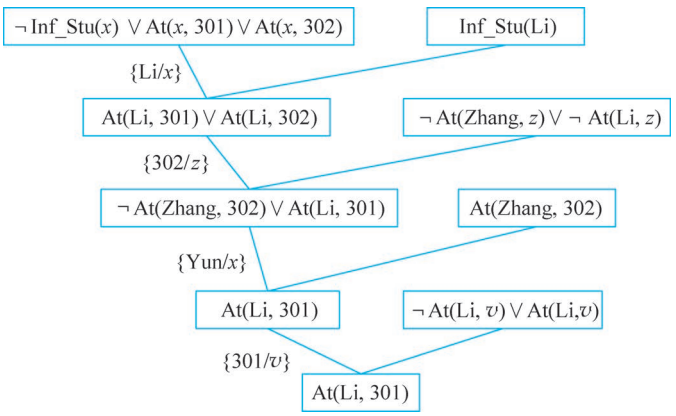

图 3.33　归结过程的树状表达

由于归结结果得到了 $At(Li,301)$，表明李振在 301 教室上课。

3.4 本章小结

本章介绍了确定性知识的表示和推理方法，要点回顾如下。

- 知识是由“信息”和“关联”两个基本要素构成的。实现信息之间关联的形式可以有很多种，其中最常用的一种形式是“如果……则……”。
- 从推理的结构角度出发，认为推理由两个以上判断组成，每个判断揭示的是概念之间的联系和关系；从过程的角度出发，可以认为推理是在给定信息和已有知识的基础上进行的一系列加工操作。
- 常用的确定性知识表示方法有谓词逻辑法、产生式表示法和语义网络法。
- 常用的确定性知识推理方法有产生式推理、自然演绎推理和归结演绎推理。

习题

1. 按知识的适用范围、知识的作用效果、知识的确定性，知识可以被分为哪些类别？分别列出类别，并举例说明。

2. 用谓词逻辑表示如下知识。

(1) 李白不仅擅长写诗，而且爱喝酒。

(2) 每个学生都有自己的指导老师。

(3) 凡是喜欢编程序的人都喜欢计算机。

(4) 在 302 教室中的学生要么是信息工程专业的，要么是通信工程专业的。

(5) 除非邱云是广东人，否则他一定不会说粤语。

(6) 不是所有生活在大海中的生物都是鱼。

3. 指出下列谓词公式的变元、量词辖域、约束变元与自由变元。

(1) $(\forall x)(P(x)\rightarrow Q(x,y))$

(2) $(\forall x)P(x,y)\rightarrow(\forall z)(\exists y)Q(z,y)$

(3) $(\forall x)(\exists y)(P(x,y)\vee Q(y,z))\wedge(\exists x)R(x,y,z)$

4. 什么是产生式？它的基本形式是什么？代表什么含义？

5. 简述产生式系统的基本组成，并逆向列举三个生活中使用产生式系统(专家系统)的案例。

6. 简述产生式系统中正向推理和逆向推理的过程及适用范围。

7. 对下列命题分别画出其语义网络。

(1) 杜老师从 4 月到 8 月给信息工程系的学生讲授“模式识别”课程。

(2) 考研培训班的学生有男有女，有应届生有往届生。

(3) 信息工程系足球队和通信工程系足球队进行了一场友谊赛，最终 2∶2 握手言和。

(4) 王朋将一本《三国演义》作为生日礼物送给了金峰。

(5) 天河科技公司位于科海路 123 号，其人力资源部主管是陈巧，女，28 岁，硕士学位，陈巧拥有一辆白色丰田牌轿车。

8. 将下列命题用一个语义网络表示出来。

(1) 树和草都是植物。

(2) 树和草都有根。

(3) 水草是草，其生活在水中。

(4) 果树是树，能结果。

(5) 桃树是果树，能结桃子。

9. 什么是置换？什么是合一？

10. 判断下列公式能否合一，如果可以合一，求出其相应的置换。

(1) $P(x,y)$和 $P(a,b)$

(2) $P(x,f(y),y)$ 和 $P(f(a),f(b),x)$

(3) $P(f(x),b)$ 和 $P(y,z)$

(4) $P(x,y)$ 和 $P(y,x)$

(5) $P(f(x),y)$ 和 $P(y,f(b))$

11. 假设 F 为已知前提，G 为欲证明的结论，简述证明 G 为真的归结演绎推理过程。

12. 将下列谓词公式化简为子句集形式。

(1) $(\forall x)(P(x)\rightarrow Q(x,y))$

(2) $\neg(\forall x)P(x,y)\rightarrow(\forall z)(\exists y)Q(z,y)$

(3) $(\forall x)(\exists y)(P(x,y)\vee Q(y,z))\rightarrow(\forall y)(\exists x)R(x,y,z)$

(4) $(\forall x)(\forall y)(P(x,y)\rightarrow\neg(\exists z)(Q(x,y)\vee\neg R(x,z))$

13. 对下列各题分别证明 G 是否为 F 的逻辑结论。

(1) F：$(\exists x)(\exists y)(P(f(x))\wedge Q(f(y)))$

G：$P(f(a))\wedge P(y)\wedge Q(y)$

(2) F：$(\forall x)(P(x)\rightarrow(\forall y)(Q(y)\rightarrow\neg R(x,y)))$

$(\exists x)(P(x)\wedge(\forall y)(K(y)\rightarrow R(x,y)))$

G：$(\forall x)(K(x)\rightarrow\neg Q(x))$

(3) F：$(\forall x)(P(x)\rightarrow(Q(x)\wedge R(x)))$

$(\exists x)(P(x)\wedge S(x))$

G：$(\exists x)(S(x)\wedge R(x))$

14. 已知所有不贫穷并且聪明的人都是快乐的，那些看书的人是聪明的。粘健能看书且不贫穷，快乐的人过着激动人心的生活。用归结演绎推理求证粘健过着激动人心的生活。

15. 已知能够阅读的个体都是有文化的；海豚是没有文化的；某些海豚是有智能的。用归结原理证明某些有智能的个体并不能阅读。

16. 假设张家被盗，公安局派出 5 个人去调查。分析案情时，侦察员 A 说“赵与钱中至少有一个人作案”，侦察员 B 说“钱与孙中至少有一个人作案”，侦察员 C 说“孙与李中至少有一个人作案”，侦察员 D 说“赵与孙中至少有一个人与此案无关”，侦察员 E 说“钱与李中至少有一个人与此案无关”。如果这 5 个侦察员的话都是可信的，试用归结演绎推理求出谁是盗窃犯。

17. 画出思维导图，串联本章所讲的知识点。