

## 第4章 逻辑推理

逻辑推理是根据一阶谓词逻辑的逻辑规则进行的一种推理，主要推理方法有自然演绎推理、归结演绎推理及与/或形演绎推理等。由于这种推理是基于谓词逻辑的，其真值只有“真”和“假”两种，因此它是一种精确推理，或称为确定性推理。

### 4.1 推理的基本概念

从已知的事实出发，通过运用已掌握的知识，找出其中蕴含的事实或归纳出新的事实，这一过程称为推理。或者说，推理是按某种策略由已知判断推出另一判断的思维过程。实现推理的程序称为推理机。

#### 4.1.1 推理方式及其分类

人类的智能活动有多种思维方式，人工智能作为对人类智能的模拟，相应也有多种推理方式，下面分别从不同的角度对它们进行讨论。

##### 1. 演绎推理、归纳推理、默认推理

从推理的途径来分类，推理可分为演绎推理、归纳推理及默认推理。

演绎推理是从全称判断推导出特称判断的过程，即由一般性知识推理适合于某一具体情况的结论。这是一种从一般到个别的推理。演绎推理有多种形式，经常用的是三段论式，它包括：

- ①大前提：已知的一般性知识或假设。
- ②小前提：关于所研究的具体情况或个别事实的判断。
- ③结论：由大前提推出的适合于小前提的新判断。

例如：

- ①足球运动员的身体都是强壮的。
- ②李波是一名足球运动员。
- ③所以，李波的身体是强壮的。

这就是一个三段论推理。其中：①是大前提；②是小前提；③是经演绎推出的结论。结论“李波的身体是强壮的”是蕴含于“足球运动员的身体都是强壮的”这一大前提之中的，它没有超出大前提所断定的范围。在任何情况下，由演绎推理导出的结论都是蕴含在大前提的一般性知识之中的。只要大前提和小前提是正确的，由它们推出的结论也必然是正确的。演绎推理是人工智能中的一种重要推理方式。

归纳推理是从足够多的事例中归纳出一般性结论的推理过程，是一种从个别到一般的推理。从归纳时所选事例的广泛性来划分，归纳推理又可分为完全归纳推理与不完全归纳推理。完全归纳推理是指在进行归纳时考察了相应事物的全部对象，并根据这些对象是否都具有某种属性，从而推出这个事物是否具有这个属性。例如，某厂进行产品质量检查，如果对每一件产

品都进行了严格检查，并且都是合格的，则推导出结论“该厂生产的产品是合格的”，这就是一个完全归纳推理。不完全归纳推理是指只考察了相应事物的部分对象，就得出了结论。例如，检查产品质量时，只是随机地抽查了部分产品，只要它们都合格，就得出“该厂生产的产品是合格的”结论，这就是一个不完全归纳推理。不完全归纳推理推出的结论不具有必然性，属于非必然性推理，而完全归纳推理是必然性推理。但由于要考察事物的所有对象通常比较困难，因而大多数归纳推理都是不完全归纳推理。归纳推理是人类思维活动中最基本、最常用的一种推理形式，人们在由个别到一般的思维过程中经常要用到它。

默认推理又称为缺省推理，它是在知识不完全的情况下假设某些条件已经具备所进行的推理。例如，在条件 A 已成立的情况下，如果没有足够的证据能证明条件 B 不成立，则就默认 B 是成立的，并在此默认的前提下进行推理，推导出某个结论。由于这种推理允许默认某些条件是成立的，使得在知识不完全的情况下也能进行推理。在默认推理过程中，如果某一时刻发现原先所作的默认不正确，则要撤消所作的默认以及由此默认推出的结论，重新按新情况进行推理。

## 2. 确定性推理和不确定性推理

按推理时所用知识的确定性来分类，推理可分为确定性推理与不确定性推理。

确定性推理是指推理时所用的知识都是确定的，推出的结论也是确定的，其真值或者为真，或者为假，没有第三种情况出现。本章将要讨论的逻辑推理就属于这一类。

不确定性推理是指推理时所用的知识不都是确定的，推出的结论也不完全是肯定的，其真值位于真与假之间。现实世界中的事物和现象大都是不确定的，许多概念是模糊的，没有明确的类属界限。在此情况下，若仍用精确逻辑做精确处理，势必要人为地在本来没有明确界限的事物间划定界限，从而舍弃了事物固有的模糊性，失去了真实性。人工智能把不确定知识的表示与不确定性推理作为重要的研究课题。

## 3. 单调推理和非单调推理

按推理过程中推出的结论是否单调增加，或者说推出的结论是否越来越接近最终目标来分类，推理可分为单调推理与非单调推理。

单调推理是指在推理过程中随着推理过程向前推进及新知识的进入，推出的结论呈单调增加的趋势，并且越来越接近最终目标，在推理过程中不会出现反复的情况，即不会由于新知识的加入否定了前面推出的结论，从而使推理又退回到前面的某一步。本章将要讨论的基于谓词逻辑的演绎推理属于单调性推理。

非单调推理是指在推理过程中由于新知识的加入，不仅没有加强已推出的结论，反而要否定它，使得推理退回到前面的某一步，重新开始。非单调推理多是在知识不完全的情况下发生的。由于知识不完全，为使推理进行下去，就要先做某些假设，并在此假设的基础上进行推理，当以后由于新知识的加入发现原先的假设不正确时，就需要推翻该假设以及以此假设为基础推出的一切结论，再用新知识重新进行推理。显然，前面所说的默认推理是非单调推理。

## 4. 启发式推理和非启发式推理

按推理中是否运用与问题有关的启发性知识，推理可分为启发式推理与非启发式推理。启发性知识是指与问题有关且能加快推理进程、求得问题解的知识。

## 5. 基于知识的推理和直觉推理

从方法论的角度分类，推理可分为基于知识的推理与直觉推理。

基于知识的推理是根据已掌握的事实，通过运用知识进行推理。我们所讨论的推理都属于这一类。

直觉推理又称为常识性推理，是根据常识进行的推理。例如，当你从某建筑物下面走过时，突然发现有一物体从建筑物上掉落下来，你立即会意识到“有危险”，并立即躲开，这就是使用了直觉推理。目前，在计算机上实现直觉推理还是一件很困难的工作，有待进行深入的研究。

#### 4.1.2 推理的控制策略

推理过程是一个求解问题的过程。问题求解的质量与效率依赖于求解问题的策略，即推理的控制策略。推理的控制策略主要包括推理方向、搜索策略、冲突消解策略、求解策略及限制策略等。

##### 1. 推理方向

推理方向用于确定推理的驱动方式，分为正向推理、逆向推理、混合推理。无论按哪种方向进行推理，一般都要求系统具有一个存放知识的知识库、一个存放初始已知事实及问题状态的数据库和一个用于推理的推理机。

##### (1) 正向推理。

正向推理是以已知事实作为出发点的推理，又称为数据驱动推理、前向链推理、模式制导推理及前件推理等。

正向推理的基本思想是：从用户提供的初始已知事实出发，在知识库 KB 中找出当前可用知识，构成可用知识集 KS，然后按某种冲突消解策略从 KS 中选出一条知识进行推理，并将推出的新事实加入到数据库 DB 中作为下一步推理的已知事实，在此之后再在知识库中选取可用知识进行推理，如此重复进行这一过程，直到求得了所要求的解或者知识库中再无可用的知识为止。其推理过程可描述如下：

①将用户提供的初始已知事实送入数据库 DB。

②检查数据库 DB 中是否已经包含了问题的解，若有，则求解结束，并成功退出；否则执行下一步。

③根据数据库 DB 中的已知事实，扫描知识库 KB，检查 KB 中是否有可用（即可与 DB 中已知事实匹配）的知识，若有，则转④；否则转⑥。

④把 KB 中所有的可用知识都选出来，构成可用知识集 KS。

⑤若 KS 不为空，则按某种冲突消解策略从中选出一条知识进行推理，并将推出的新事实加入 DB 中，然后转②；若 KS 为空，则转⑥。

⑥询问用户是否可进一步补充新的事实，若可补充，则将补充的新事实加入 DB 中，然后转③；否则，失败退出。

上述推理过程在具体实现时还有许多工作要做。例如，在推理过程中要从知识库 KB 中选出可用知识，就要用知识库中的知识与数据库中的已知事实进行匹配，为此就需要确定匹配的方法。而且，匹配通常难以做到完全一致，这就需要解决怎样才算是匹配成功的问题。其次，为了进行匹配，就要查找知识，这就涉及按什么策略搜索知识库。再如，如果可用的知识只有一条，系统就可立即用它进行推理，并将推出的新事实送入数据库 DB 中；但是，如果当前可用的知识有多条，应该先用哪一条？这是推理的一个重要问题，称为冲突消解策略。

### （2）逆向推理。

逆向推理是以某个假设目标作为出发点的推理，又称为目标驱动推理、逆向链推理、目标制导推理及后件推理等。

逆向推理的基本思想是：首先选定一个假设目标，然后寻找支持该假设的证据，若所需的证据都能找到，则说明原假设是成立的；若无论如何都找不到所需要的证据，则说明原假设不成立，此时需要另外选定新的假设。其推理过程可描述如下：

①提出要求证的目标（假设）。

②检查该目标是否已在数据库 DB 中，若在，则该目标成立，成功退出推理或者对下一个假设目标进行验证；否则，转③。

③判断该目标是否是证据，即它是否为应由用户证实的原始事实，若是，则询问用户；否则转④。

④在知识库 KB 中找出所有能导出该目标的知识，形成可用知识集 KS，然后转⑤。

⑤从 KS 中选出一条知识，并将该知识的运用条件作为新的假设目标，然后转②。

与正向推理相比，逆向推理更复杂一些，上述过程只是描述了它的大致过程，许多细节也没有反映出来。

逆向推理的主要优点是不必使用与目标无关的知识，目的性强。其主要缺点是初始目标的选择有盲目性，若不符合实际，就要多次提出假设，也会影响到系统的效率。

### （3）混合推理。

正向推理可能会推出许多与问题求解无关的子目标；逆向推理中，若提出的假设目标不符合实际，也会降低系统的效率。可把正向推理与逆向推理结合起来，使其各自发挥自己的优势，取长补短，既有正向又有逆向的推理称为混合推理。在下述几种情况下，通常需要进行混合推理：

①已知的事实不充分。当数据库中的已知事实不够充分时，若用这些事实与知识的运用条件进行匹配进行正向推理，可能一条适用知识都选不出来，这就使推理无法进行下去。此时，可通过正向推理先把其运用条件不能完全匹配的知识都找出来，并把这些知识可导出的结论作为假设，然后分别对这些假设进行逆向推理。由于在逆向推理中可以向用户询问有关证据，这就可能使推理进行下去。

②由正向推理推出的结论可信度不高。用正向推理进行推理时，虽然推出了结论，但可信度可能不高，达不到预定的要求。此时为了得到一个可信度符合要求的结论，可用这些结论作为假设，然后进行逆向推理，通过向用户询问进一步的信息，有可能会得到可信度较高的结论。

③希望得到更多的结论。在逆向推理过程中，由于可与用户进行对话，有针对性地向用户提出询问，这就可能获得一些原来不掌握的有用信息，这些信息不仅可用于证实要证明的假设，同时还可能有助于推出一些其他结论。因此，在用逆向推理证实了某个假设之后，可以再用正向推理推出另外一些结论。

由以上讨论可以看出，混合推理分为两种情况：一种是先进行正向推理，帮助选择某个目标，即从已知事实演绎出部分结果，然后再用逆向推理证实该目标或提高其可信度；另一种情况是先假设一个目标进行逆向推理，然后再利用逆向推理中得到的信息进行正向推理，以推出更多的结论。

## 2. 求解策略

所谓推理的求解策略是指，推理是只求一个解，还是求所有解以及最优解等。例如前述的正向推理只用于求一个解，只要略加修改就可用来求所有解。

## 3. 限制策略

为了防止无穷的推理过程，以及由于推理过程太长而增加时间及空间的复杂性，可在控制策略中指定推理的限制条件，以对推理的深度、宽度、时间、空间等进行限制。

## 4. 冲突消解策略

在推理过程中，系统要不断地用当前已知事实与知识库中的知识进行匹配，此时可能发生如下三种情况：

- ① 已知事实不能与知识库中的任何知识匹配成功。
- ② 已知事实恰好与知识库中的一个知识匹配成功。
- ③ 已知事实可与知识库中的多个知识匹配成功。

当第一种情况发生时，由于找不到与当前已知事实匹配成功的知识，就使得推理无法继续进行下去，这或者是由于知识库中缺少某些必要的知识，或者是由于欲求解的问题超出了系统的功能范围等，此时可根据当时的实际情况做相应的处理。对于第二种情况，由于匹配成功的知识只有一个，所以它就是可用知识，可直接把它用于当前的推理。第三种情况不仅有知识匹配成功，而且有多个知识匹配成功，称这种情况为发生了冲突。此时需要按一定策略解决冲突，以便从中挑选一个知识用于当前的推理，称这一解决冲突的过程为冲突消解，解决冲突所用的方法称为冲突消解策略。

例如，在产生式系统中，若出现如下情况就认为发生了冲突：

① 对正向推理而言，如果有多条产生式规则的前件都和已知事实匹配成功，或者有多组不同的已知事实与多条产生式规则的前件匹配成功。

② 对逆向推理而言，如果有多条产生式规则的后件都和同一个假设匹配成功；或者有多条产生式规则的后件可与多个假设匹配成功。

冲突消解的任务是解决冲突。对正向推理来说，它将决定选择哪一组已知事实来激活哪一条产生式规则，产生其后件指出的结论或执行后件指定的操作。对逆向推理来说，它将决定用哪一个假设与哪一个产生式规则的后件进行匹配，推出相应的前件作为新的假设。

目前已有多种消解冲突的策略，其基本思想都是对知识进行排序，常用的有以下几种。

(1) 按针对性排序。

设有如下两条产生式规则：

$$R_1: \text{if } A_1 \text{ and } A_2 \text{ and } \cdots \text{ and } A_n \text{ then } H_1$$

$$R_2: \text{if } B_1 \text{ and } B_2 \text{ and } \cdots \text{ and } B_m \text{ then } H_2$$

如果  $R_2$  中除了包含  $R_1$  的全部条件  $A_1, A_2, \dots, A_n$  外，还包含其他条件，则称  $R_2$  比  $R_1$  有更大的针对性， $R_1$  比  $R_2$  有更大的通用性。

本策略优先选用针对性较大的产生式规则。因为它要求的条件较多，所以其结论一般更接近于目标，一旦得到满足，就可缩短推理过程。

(2) 按匹配度排序。

在不确定性匹配中，为了确定两个知识模式是否可以匹配，需要计算这两个模式的相似程度，当其相似度达到某个预先规定的值时，就认为它们是可匹配的。相似度又称为匹配度，

它除了可用来确定两个知识模式是否可匹配外，还可用于冲突消解。若产生式规则  $R_1$  与  $R_2$  都可匹配成功，则可根据它们的匹配度来决定哪一个产生式规则可优先被使用。

(3) 根据领域问题的特点排序。

如果事先可知道领域问题的某些特点，则可根据这些特点事先确定知识库中知识的使用顺序。例如：

①当领域问题有固定的求解次序时，可按该次序对知识库中的知识排序，排在前面的知识优先被使用。

②当已知某些产生式规则被使用后会明显地有利于问题的求解时，可对这些产生式规则指定较高的优先级，使这些产生式规则优先被使用。

#### 4.1.3 模式匹配及其变量代换

模式匹配是指两个知识模式（如两个谓词公式、两个框架片断等）的比较，检查这两个知识模式是否完全一致或近似一致。如果两者完全一致，或者虽不完全一致但其相似度落在指定的限度内，就称它们是可匹配的，否则为不可匹配。

模式匹配是推理中必须进行的一项重要工作，因为只有经过模式匹配才能从知识库中选出当前适用的知识，才能进行推理。例如在产生式系统中，为了由已知的初始事实推出相应的结论，首先必须从知识库中选出可与已知事实匹配的的产生式规则，然后才能应用这些产生式规则进行推理，逐步推出结论。框架推理与此类似，也需要先通过匹配选出相应的框架片断，然后再进行推理。

按匹配时两个知识模式的相似程度划分，模式匹配可分为确定性匹配与不确定性匹配。

确定性匹配是指两个知识模式完全一致，或者经过变量代换后变得完全一致。例如，设有如下两个知识模式：

$P_1$ : Father (李四, 李小四) and Man (李小四)

$P_2$ : Father( $x$ , $y$ ) and Man ( $y$ )

若用常量“李四”代换变量  $x$ ，用常量“李小四”代换变量  $y$ ，则  $P_1$  与  $P_2$  就变得完全一致。若用这两个知识模式进行匹配，则它们是确定性匹配。确定性匹配又称为完全匹配或精确匹配。

不确定性匹配是指两个知识模式不完全一致，但从总体上看，它们的相似程度又落在规定的限度内。

无论是确定性匹配还是不确定性匹配，在进行匹配时一般都需要进行变量的代换。

**定义 4.1** 代换是形如

$$\{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$$

的有限集合。其中， $t_1, t_2, \dots, t_n$  是项； $x_1, x_2, \dots, x_n$  是互不相同的变元； $t_i/x_i$  表示用  $t_i$  代换  $x_i$ ，不允许  $t_i$  与  $x_i$  相同，也不允许变元  $x_i$  循环地出现在另一个  $t_j$  中。

例如

$$\{g(y)/x, f(x)/y\}$$

不是一个代换，因为代换的目的是使某些变元被另外的变元、常量或函数取代，使之不再在公式中出现，而  $\{g(y)/x, f(x)/y\}$  在  $x$  与  $y$  之间出现了循环代换的情况，它既没有消去  $x$ ，也没有消去  $y$ 。如果将它改为

$$\{g(A)/x, f(x)/y\}$$

就可以，它将把公式中的  $x$  用  $g(A)$  代换， $y$  用  $f(g(A))$  代换，从而消去了变元  $x$  和  $y$ 。

定义 4.2 设

$$\theta = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$$

$$\lambda = \{u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$$

是两个代换，则此两个代换的复合也是一个代换，它是从

$$\{t_1\lambda/x_1, t_2\lambda/x_2, \dots, t_n\lambda/x_n, u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$$

中删去如下两种元素：

$$t_i\lambda/x_i \quad \text{当 } t_i\lambda = x_i$$

$$u_i/y_i \quad \text{当 } y_i \in \{x_1, x_2, \dots, x_n\}$$

后剩下的元素所构成的集合，记为  $\theta \circ \lambda$ 。

例如，设有代换

$$\theta = \{f(y)/x, z/y\}$$

$$\lambda = \{A/x, B/y, y/z\}$$

则

$$\theta \circ \lambda = \{f(B)/x, y/z\}$$

定义 4.3 设有公式集  $F = \{F_1, F_2, \dots, F_n\}$ ，若存在一个代换  $\lambda$  使得

$$F_1\lambda = F_2\lambda = \dots = F_n\lambda$$

则称  $\lambda$  为公式集  $F$  的一个合一，且称  $F_1, F_2, \dots, F_n$  是可合一的。

例如，设有公式集

$$F = \{P(x, y, f(y)), P(A, g(x), z)\}$$

则下式是它的一个合一：

$$\lambda = \{A/x, g(A)/y, f(g(A))/z\}$$

一个公式集的合一通常是不唯一的。

定义 4.4 设  $\sigma$  是公式集  $F$  的一个合一，如果对任一个合一  $\theta$  都存在一个代换  $\lambda$ ，使得

$$\theta = \sigma \circ \lambda$$

则称  $\sigma$  是公式集  $F$  的最一般合一 (mgu)。

最一般合一是唯一的。若用最一般合一去代换那些可合一的谓词公式，可使它们变成完全一致的谓词公式。由此可知，为了使两个知识模式匹配，可用其最一般合一对它们进行代换。

在给出求取最一般合一的算法之前，先引入差异集的概念。设有如下两个谓词公式：

$$F_1: P(x, y, z)$$

$$F_2: P(x, f(A), h(B))$$

分别从  $F_1$  与  $F_2$  的第一个符号开始，逐个向右比较，此时发现  $F_1$  中  $y$  与  $F_2$  中的  $f(A)$  不同，它们构成了一个差异集：

$$D_1 = \{y, f(A)\}$$

当继续向右边比较时，又发现  $F_1$  中的  $z$  与  $F_2$  中的  $h(B)$  不同，则又得到一个差异集：

$$D_2 = \{z, h(B)\}$$

下面，对任意公式集  $F$ ，给出求最一般合一的算法。

### 求最一般合一算法

- (1) 初始化, 令  $k=0$ ,  $F_k=F$ ,  $\sigma_k = \Phi$ 。其中,  $\Phi$  是代换空集。
- (2) 若  $F_k$  只含一个表达式, 则算法停止,  $\sigma_k$  就是最一般合一; 否则转步骤 (3)。
- (3) 找出  $F_k$  的差异集  $D_k$ 。
- (4) 若  $D_k$  中存在变元  $x_k$  和项  $t_k$ , 且  $x_k$  不在  $t_k$  中出现, 则:

$$\begin{aligned}\sigma_{k+1} &= \sigma_k \circ \{t_k / x_k\} \\ F_{k+1} &= F_k \{t_k / x_k\} \\ k &= k + 1\end{aligned}$$

转步骤 (2); 否则, 算法终止,  $F$  的最一般合一不存在。

#### 例 4.1 设有公式集

$$F = \{P(A, x, f(g(y))), P(z, f(z), f(u))\}$$

求其最一般合一。

解: 初始化, 令  $k=0$ ,  $\sigma_0 = \Phi$ ,  $F_0 = F = \{P(A, x, f(g(y))), P(z, f(z), f(u))\}$

Loop 1:  $F_0$  含有 2 个表达式, 故  $\sigma_0$  不是最一般合一,

$F_0$  的差异集  $D_0 = \{A, z\}$ , 可有代换  $A/z$ ,

$$\sigma_1 = \sigma_0 \circ \{A/z\} = \{A/z\}$$

$$F_1 = F_0 \{A/z\} = \{P(A, x, f(g(y))), P(A, f(A), f(u))\}$$

Loop 2:  $F_1$  含有 2 个表达式, 故  $\sigma_1$  不是最一般合一,

$F_1$  的差异集  $D_1 = \{x, f(A)\}$ , 可有代换  $\{f(A)/x\}$ ,

$$\sigma_2 = \sigma_1 \circ \{f(A)/x\} = \{A/z\} \circ \{f(A)/x\} = \{A/z, f(A)/x\}$$

$$F_2 = F_1 \{f(A)/x\} = \{P(A, f(A), f(g(y))), P(A, f(A), f(u))\}$$

Loop 3:  $F_2$  含有 2 个表达式, 故  $\sigma_2$  不是最一般合一,

$F_2$  的差异集  $D_2 = \{g(y), u\}$ , 可有代换  $\{g(y)/u\}$ ,

$$\sigma_3 = \sigma_2 \circ \{g(y)/u\} = \{A/z, f(A)/x\} \circ \{g(y)/u\} = \{A/z, f(A)/x, g(y)/u\}$$

$$F_3 = F_2 \{g(y)/u\} = \{P(A, f(A), f(g(y))), P(A, f(A), f(g(y)))\} = \{P(A, f(A), f(g(y)))\}$$

Loop 4:  $F_3$  中只含有一个表达式, 故算法成功终止。

$\sigma_3 = \{A/z, f(A)/x, g(y)/u\}$ , 即为公式集  $F$  的最一般合一。

## 4.2 归结演绎推理

自动定理证明是人工智能的一个重要研究领域, 定理证明是对已知前提  $P$  和待证结论  $Q$  证明  $P \rightarrow Q$  的永真性。但是, 正如在第 2 章所讨论的那样, 要证明一个谓词公式的永真性是相当困难的, 甚至在某些情况下是不可能的。通过研究发现, 应用反证法的思想可把永真性的证明转化为不可满足性的证明, 即欲证明  $P \rightarrow Q$  永真, 只要证明  $P \wedge \neg Q$  是不可满足的就可以了。关于不可满足性的证明, 海伯伦 (Herbrand) 及鲁宾逊 (Robinson) 先后进行了卓有成效的研究, 提出了相应的理论和方法。海伯伦的理论和鲁宾逊的归结原理都是以子句集为背景开展研究的。

### 4.2.1 谓词公式化为子句集的方法

**定义 4.5** 在谓词逻辑中,把原子谓词公式及其否定统称为文字。任何文字的析取式称为子句。

例如,  $P(x) \vee Q(x)$ ,  $\neg P(x, f(x)) \vee Q(x, g(x))$  都是子句。

**定义 4.6** 不包含任何文字的子句称为空子句。

由于空子句不含有文字,它不能被任何解释满足,所以空子句是永假的,不可满足的。

由子句构成的集合称为子句集。在谓词逻辑中,任何一个谓词公式都可通过应用等价关系化成相应的子句集。

把谓词公式化成子句集的步骤如下。

(1) 消去蕴含连词。

利用下述等价关系消去谓词公式中的蕴含连词“ $\rightarrow$ ”。

$$P \rightarrow Q \Leftrightarrow \neg P \vee Q$$

(2) 减小否定连词的辖域。

利用下述等价关系把“ $\neg$ ”移到紧靠谓词的位置上:

$$\begin{aligned} \neg(\neg P) &\Leftrightarrow P \\ \neg(P \wedge Q) &\Leftrightarrow \neg P \vee \neg Q \\ \neg(P \vee Q) &\Leftrightarrow \neg P \wedge \neg Q \\ \neg(\forall x)P &\Leftrightarrow (\exists x)\neg P \\ \neg(\exists x)P &\Leftrightarrow (\forall x)\neg P \end{aligned}$$

(3) 约束变元标准化。

重新命名约束变元名,使不同量词的约束变元有不同的名字。

(4) 消去存在量词。

若存在量词不在全称量词的辖域内,则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存在量词(因为若原公式为真,则总能找到一个个体常量,替换后仍使公式为真);若存在量词位于一个或多个全称量词的辖域内,例如

$$(\forall x_1)(\forall x_2)\cdots(\forall x_n)(\exists y)P(x_1, x_2, \cdots, x_n, y)$$

则需要用 Skolem 函数  $f(x_1, x_2, \cdots, x_n)$  替换受该存在量词约束的变元  $y$ , 然后才能消去存在量词。

(5) 组成全称量词前缀。

把全称量词全部移到公式的左边,组成全称量词前缀。

(6) 把母式化为 Skolem 标准形。

利用下述等价关系

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

把公式化为 Skolem 标准形。

Skolem 标准形的一般形式是

$$(\forall x_1)(\forall x_2)\cdots(\forall x_n)M$$

其中,  $M$  是子句的合取式,称为 Skolem 标准形的母式。

(7) 消去全称量词。

(8) 对变元更名,使不同子句中的变元不同名。

(9) 消去合取连词, 得到子句集。

例 4.2 请将下述谓词公式化为子句集:

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

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

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

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

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

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

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

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

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

消去合取词后, 上式就变为下述子句集:

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

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

如果谓词公式是不可满足的, 则其子句集也一定是不可满足的, 反之亦然。因此, 在不可满足的意义上两者是等价的, 由此给出下述定理。

**定理 4.1** 设有谓词公式 F, 其标准形的子句集为 S, 则 F 不可满足的充要条件是 S 不可满足。

由此定理可知, 为要证明一个谓词公式是不可满足的, 只要证明相应的子句集是不可满足的就可以了。

#### 4.2.2 归结原理

由谓词公式转化子句集的过程可以看出, 子句集中的子句之间是合取关系, 其中只要有一个子句不可满足, 则子句集就不可满足。另外, 已经指出空子句是不可满足的。因此, 若一个子句集中包含空子句, 则这个子句集一定是不可满足的。鲁宾逊归结原理就是基于这一认识提出来的。其基本思想是: 检查子句集 S 中是否包含空子句, 若包含, 则 S 不可满足; 若不包含, 就在子句集中选择合适的子句进行归结, 一旦通过归结能推出空子句, 就说明子句集 S 是不可满足的。

##### 1. 命题逻辑中的归结原理

**定义 4.7** 若 P 是原子谓词公式, 则称 P 与  $\neg P$  为互补文字。

**定义 4.8** 设  $C_1$  与  $C_2$  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, 那么从  $C_1$  和  $C_2$  中分别消去  $L_1$  和  $L_2$ , 并将二个子句中余下的部分析取, 构成一个新子句  $C_{12}$ , 称这一过程为归结, 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。

**定理 4.2** 归结式  $C_{12}$  是其亲本子句  $C_1$  与  $C_2$  的逻辑结论。

**证明:** 设

$$C_1 = L \vee C'_1, C_2 = \neg L \vee C'_2$$

通过归结可以得到

$$C_{12} = C'_1 \vee C'_2$$

$C_1$  和  $C_2$  是  $C_{12}$  的亲本子句。

$$\begin{aligned} & \because C_1 \vee L \Leftrightarrow \neg C_1 \rightarrow L \\ & \quad \neg L \vee C_2 \Leftrightarrow L \rightarrow C_2 \\ & \therefore C_1 \wedge C_2 = (\neg C_1 \rightarrow L) \wedge (L \rightarrow C_2) \end{aligned}$$

根据假言三段论得到:

$$\begin{aligned} & (\neg C_1 \rightarrow L) \wedge (L \rightarrow C_2) \Rightarrow \neg C_1 \rightarrow C_2 \\ & \therefore \neg C_1 \rightarrow C_2 \Leftrightarrow C_1 \vee C_2 = C_{12} \\ & \therefore C_1 \wedge C_2 \Rightarrow C_{12} \end{aligned}$$

由第2章关于逻辑结论的定义2.6, 可知  $C_{12}$  是其亲本子句  $C_1$  和  $C_2$  的逻辑结论。

这个定理是归结原理中的一个很重要的定理, 由它可得到如下两个推论。

**推论 4.1** 设  $C_1$  与  $C_2$  是子句集  $S$  中的两个子句,  $C_{12}$  是它们的归结式, 若用  $C_{12}$  代替  $C_1$  和  $C_2$  后得到新子句集  $S_1$ , 则由  $S_1$  的不可满足性可推出原子句集  $S$  的不可满足性, 即

$$S_1 \text{ 的不可满足性} \Rightarrow S \text{ 的不可满足性}$$

**推论 4.2** 设  $C_1$  与  $C_2$  是子句集  $S$  中的两个子句,  $C_{12}$  是它们的归结式, 若把  $C_{12}$  加入  $S$  中, 得到新子句集  $S_2$ , 则  $S$  与  $S_2$  在不可满足的意义上是等价的, 即

$$S_2 \text{ 的不可满足性} \Leftrightarrow S \text{ 的不可满足性}$$

这两个推论给出了用归结原理证明子句集不可满足性的基本思想: 为了要证明子句集  $S$  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  $S$ , 或者用归结式替换它的亲本子句, 然后对新子句集 ( $S_1$  或  $S_2$ ) 证明不可满足性就可以了。如果经过归结能得到空子句, 根据空子句的不可满足性, 立即可得出原子句集  $S$  是不可满足的。

在命题逻辑中, 对不可满足的子句集  $S$ , 归结原理是完备的, 即: 若子句集不可满足, 则必然存在一个从  $S$  到空子句的归结演绎; 若存在一个从  $S$  到空子句的归结演绎, 则  $S$  一定是不可满足的。但是对于可满足的子句集  $S$ , 用归结原理得不到任何结果。

## 2. 谓词逻辑中的归结原理

在谓词逻辑中, 由于子句中含有变元, 所以不可直接消去互补文字, 而需要先用最一般合一对变元进行代换, 然后才能进行归结。例如, 设有如下两个子句:

$$\begin{aligned} C_1 &= P(x) \vee Q(x) \\ C_2 &= \neg P(A) \vee R(y) \end{aligned}$$

由于  $P(x)$  与  $P(A)$  不同, 所以  $C_1$  与  $C_2$  不能直接进行归结, 但若用最一般合一

$$\sigma = \{A/x\}$$

对两个子句分别进行代换:

$$C_1\sigma = P(A) \vee Q(A) \quad C_2\sigma = \neg P(A) \vee R(y)$$

就可对它们进行归结, 消去  $P(A)$  与  $\neg P(A)$ , 得到如下归结式:

$$Q(A) \vee R(y)$$

下面给出谓词逻辑中关于归结的定义。

**定义 4.9** 设  $C_1$  与  $C_2$  是两个没有相同变元的子句,  $L_1$  和  $L_2$  分别是  $C_1$  和  $C_2$  中的文字, 若  $\sigma$  是  $L_1$  和  $\neg L_2$  的最一般合一, 则称

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

为  $C_1$  和  $C_2$  的二元归结式,  $L_1$  和  $L_2$  称为归结式的文字。

## 例 4.3 设

$$C_1 = P(A) \vee \neg Q(x) \vee R(x)$$

$$C_2 = \neg P(y) \vee Q(B)$$

给出  $C_1$  和  $C_2$  的归结式。

解：若选  $L_1 = P(A)$ ,  $L_2 = \neg P(y)$ , 则  $\sigma = \{A/y\}$  是  $L_1$  与  $\neg L_2$  的最一般合一, 根据定义 4.9, 可得:

$$\begin{aligned} C_{12} &= (C_1\sigma - \{L_1\sigma\}) \cup (C_2\sigma - \{L_2\sigma\}) \\ &= (\{P(A), \neg Q(x), R(x)\} - \{P(A)\}) \cup (\{\neg P(A), Q(B)\} - \{\neg P(A)\}) \\ &= (\{\neg Q(x), R(x)\}) \cup (\{Q(B)\}) \\ &= \{\neg Q(x), R(x), Q(B)\} \\ &= \neg Q(x) \vee R(x) \vee Q(B) \end{aligned}$$

上述归结过程可以用归结树表示, 如图 4.1 所示。

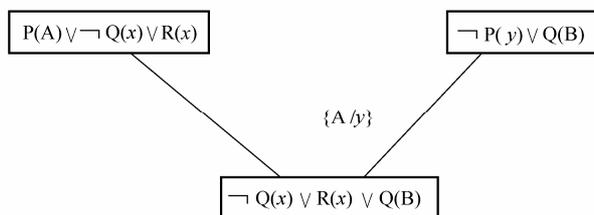


图 4.1 例 4.3 的一种归结树

若选  $L_1 = \neg Q(x)$ ,  $L_2 = Q(B)$ ,  $\sigma = \{B/x\}$ , 则可得:

$$\begin{aligned} C_{12} &= (\{P(A), \neg Q(B), R(B)\} - \{\neg Q(B)\}) \cup (\{\neg P(y), Q(B)\} - \{Q(B)\}) \\ &= (\{P(A), R(B)\}) \cup (\{\neg P(y)\}) \\ &= \{P(A), R(B), \neg P(y)\} \\ &= P(A) \vee R(B) \vee \neg P(y) \end{aligned}$$

上述归结过程的归结树如图 4.2 所示。

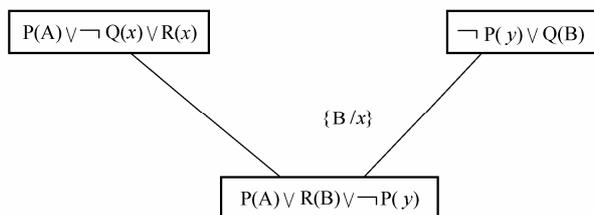


图 4.2 例 4.3 的另一种归结树

由此可见, 谓词逻辑中含有变元, 对 2 个子句中的变元选择不同的代换, 得出的归结式是不同的。

## 例 4.4 设

$$C_1 = P(x) \vee Q(A), \quad C_2 = \neg P(B) \vee R(x)$$

给出  $C_1$  和  $C_2$  的归结式。

**解：**由于  $C_1$  与  $C_2$  有相同的变元，不符合定义 4.9 的要求。为了进行归结，需修改  $C_2$  中变元的名字，令  $C_2 = \neg P(B) \vee R(y)$ 。此时，对  $C_1$  和  $C_2$  有

$$L_1 = P(x), L_2 = \neg P(B)$$

$L_1$  与  $\neg L_2$  的最一般合一  $\sigma = \{B/x\}$ ，则

$$\begin{aligned} C_{12} &= (\{P(B), Q(A)\} - \{P(B)\}) \cup (\{\neg P(B), R(y)\} - \{\neg P(B)\}) \\ &= \{Q(A), R(y)\} \\ &= Q(A) \vee R(y) \end{aligned}$$

可以用归结树直接表示  $C_1$  和  $C_2$  的归结，如图 4.3 所示。

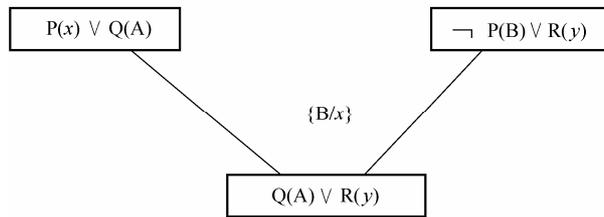


图 4.3 例 4.4 的归结树

如果在参加归结的子句内部含有可合一的文字，则在进行归结之前应对这些文字先进行合一。

**例 4.5** 设有如下两个子句：

$$C_1 = P(x) \vee P(f(A)) \vee Q(x)$$

$$C_2 = \neg P(y) \vee R(B)$$

给出  $C_1$  和  $C_2$  的归结式。

**解：**在  $C_1$  中有可合一的文字  $P(x)$  与  $P(f(A))$ ，若用它们的最一般合一  $\theta = \{f(A)/x\}$  进行代换，得到

$$C_1 \theta = P(f(A)) \vee Q(f(A))$$

此时可对  $C_1 \theta$  和  $C_2$  进行归结，从而得到  $C_1$  与  $C_2$  的二元归结式。

对  $C_1 \theta$  和  $C_2$  分别选  $L_1 = P(f(A))$ ， $L_2 = \neg P(y)$ 。 $L_1$  和  $\neg L_2$  的最一般合一是  $\sigma = \{f(A)/y\}$ ，则

$$C_{12} = R(B) \vee Q(f(A))$$

在上例中，把  $C_1 \theta$  称为  $C_1$  的因子。一般来说，若子句  $C$  中有两个或两个以上的文字具有最一般合一  $\sigma$ ，则称为  $C\sigma$  为子句  $C$  的因子。如果  $C\sigma$  是一个单文字，则称它为  $C$  的单元因子。

应用因子的概念，可对谓词逻辑中的归结原理给出如下定义。

**定义 4.10** 子句  $C_1$  和  $C_2$  的归结式是下列二元归结式之一：

- ①  $C_1$  与  $C_2$  的二元归结式；
- ②  $C_1$  与  $C_2$  的因子  $C_2\sigma_2$  的二元归结式；
- ③  $C_1$  的因子  $C_1\sigma_1$  与  $C_2$  的二元归结式；
- ④  $C_1$  的因子  $C_1\sigma_1$  与  $C_2$  的因子  $C_2\sigma_2$  的二元归结式。

对于谓词逻辑，定理 4.2 仍然适用，即归结式是它的亲本子句的逻辑结论。用归结式取代它在子句集  $S$  中的亲本子句所得到的新子句仍然保持着原子句集  $S$  的不可满足性。

一阶谓词逻辑从不可满足的意义上说，归结原理也是完备的。即：若子句集是不可满足的，则必存在一个从该子句集到空子句的归结演绎；若从子句集存在一个到空子句的演绎，则该子句集是不可满足的。

### 4.2.3 归结反演

为了证明  $Q$  为  $P_1, P_2, \dots, P_n$  的逻辑结论，根据定义 2.6 只需证明谓词公式

$$(P_1 \wedge P_2 \wedge \dots \wedge P_n) \wedge \neg Q$$

是不可满足的；根据定理 4.1，在不可满足的意义上，该谓词公式与其子句集是等价的。可用归结原理来进行子句集的不可满足性的证明，从而证明  $Q$  是  $P_1, P_2, \dots, P_n$  的逻辑结论。

应用归结原理证明结论为真的过程称为归结反演。

设  $F$  为已知前提的公式集， $Q$  为目标公式（结论），用归结反演证明  $Q$  为真的步骤是：

- ① 否定  $Q$ ，得到  $\neg Q$ ；
- ② 把  $\neg Q$  加入到公式集  $F$  中，得到  $\{F, \neg Q\}$ ；
- ③ 把公式集  $\{F, \neg Q\}$  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，并把每次归结得到的归结式都加入  $S$  中。

如此反复进行，若出现了空子句，则停止归结，此时就证明了  $Q$  为真。

**例 4.6** 已知

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

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

求证： $G$  是  $F$  的逻辑结论。

**证：** 首先把  $F$  和  $\neg G$  化为子句集：

$$\left. \begin{array}{l} (1) \neg A(x, y) \vee \neg B(y) \vee C(f(x)) \\ (2) \neg A(x, y) \vee \neg B(y) \vee D(x, f(x)) \end{array} \right\} F$$

$$\left. \begin{array}{l} (3) \neg C(z) \\ (4) A(A, B) \\ (5) B(B) \end{array} \right\} \neg G$$

归结反演过程用归结树表示如图 4.4 所示。

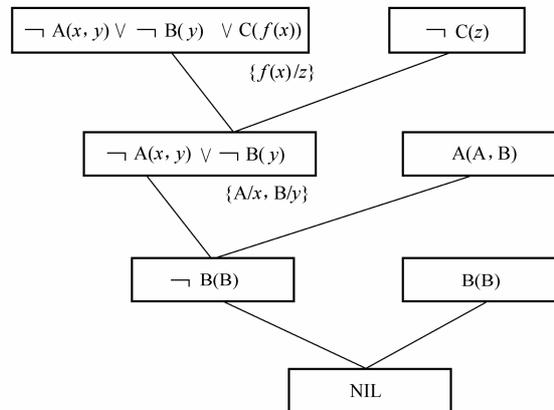


图 4.4 例 4.6 的归结树

例 4.7 在第 2 章的例 2.4 中曾经得到如下公式：

$F_1: (\forall x)(N(x) \rightarrow GZ(x) \wedge I(x))$  自然数都是大于零的整数。

$F_2: (\forall x)(I(x) \rightarrow E(x) \vee O(x))$  所有整数不是偶数就是奇数。

$F_3: (\forall x)(E(x) \rightarrow I(s(x)))$  偶数除以 2 是整数。

求证：所有自然数不是奇数就是其一半为整数的数。

证明：首先把求证的问题用谓词公式表示出来：

$$G: (\forall x)(N(x) \rightarrow (O(x) \vee I(s(x))))$$

把  $F_1, F_2, F_3$  及  $\neg G$  化成子句集：

- (1)  $\neg N(x) \vee GZ(x)$
- (2)  $\neg N(u) \vee I(u)$
- (3)  $\neg I(y) \vee E(y) \vee O(y)$
- (4)  $\neg E(z) \vee I(s(z))$
- (5)  $N(A)$
- (6)  $\neg O(A)$
- (7)  $\neg I(s(A))$

对上述子句进行的归结反演过程用归结树表示如图 4.5 所示。

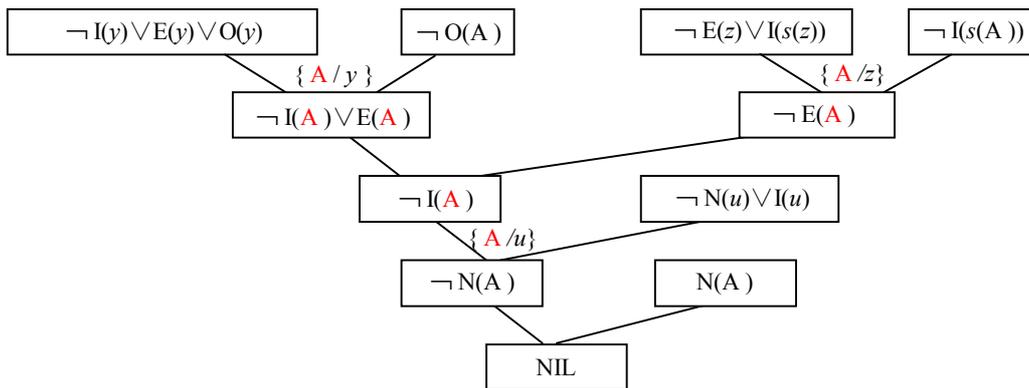


图 4.5 例 4.7 的归结树

### 4.3 基于归结反演的问题求解

归结反演除了可用于定理证明外，还可用来求取问题的答案，问题求解的方法与定理证明类似。问题求解的步骤如下：

①把已知前提用谓词公式表示出来，并且化为相应的子句集  $S$ 。

②把待求解的问题也用谓词公式表示出来，然后把它的否定式与谓词 ANSWER 构成一个析取式，ANSWER 是一个为了求解问题而专设的谓词，其变元数量和变元名必须与问题公式的变元完全一致。

③把此析取式化为子句集，并且把该子句集加入到子句集  $S$  中，得到子句集  $S'$ 。

④对  $S'$  应用归结原理进行归结。

⑤若在归结树的根节点中仅得到归结式 ANSWER，则答案就在 ANSWER 中。

**例 4.8** 已知

F<sub>1</sub>: 王 (Wang) 先生是小李 (Li) 的老师。

F<sub>2</sub>: 小李与小张 (Zhang) 是同班同学。

F<sub>3</sub>: 如果  $x$  与  $y$  是同班同学，则  $x$  的老师也是  $y$  的老师。

求: 小张的老师是谁?

**解:** 首先定义谓词:

$T(x, y)$   $x$  是  $y$  的老师。

$C(x, y)$   $x$  与  $y$  是同班同学。

把已知前提及待求解的问题表示成谓词公式:

F<sub>1</sub>:  $T(\text{Wang}, \text{Li})$

F<sub>2</sub>:  $C(\text{Li}, \text{Zhang})$

F<sub>3</sub>:  $(\forall x)(\forall y)(\forall z)(C(x, y) \wedge T(z, x) \rightarrow T(z, y))$

G:  $(\exists x)T(x, \text{Zhang})$

目标公式 G 的否定式与 ANSWER 的析取式为:

$$\neg(\exists x)T(x, \text{Zhang}) \vee \text{ANSWER}(x)$$

把上述公式化为子句集:

(1)  $T(\text{Wang}, \text{Li})$

(2)  $C(\text{Li}, \text{Zhang})$

(3)  $\neg C(x, y) \vee \neg T(z, x) \vee T(z, y)$

(4)  $\neg T(u, \text{Zhang}) \vee \text{ANSWER}(u)$

应用归结原理进行归结，归结过程可用图 4.6 所示的归结树表示。

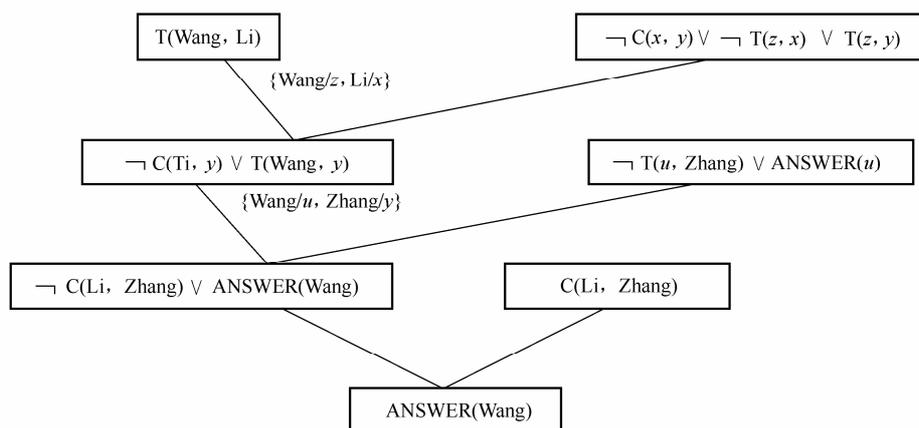


图 4.6 例 4.8 的归结树

由 ANSWER(Wang) 得出小张的老师是王先生。

**例 4.9** 设 A, B, C 三人中有人从不说真话，也有人从不说假话，某人向这三个人分别提出同一个问题：谁是说谎者？A 答：“B 和 C 都是说谎者。” B 答：“A 和 C 都是说谎者。” C

答：“A 和 B 中至少有一个是说谎者。”求谁是老实人，谁是说谎者？

解：首先，定义谓词  $T(x)$  表示  $x$  说真话。

如果 A 说的是真话，则有

$$T(A) \rightarrow \neg T(B) \wedge \neg T(C)$$

如果 A 说的是假话，则有

$$\neg T(A) \rightarrow T(B) \vee T(C)$$

对 B 和 C 说的话作相同的处理，可得

$$T(B) \rightarrow \neg T(A) \wedge \neg T(C)$$

$$\neg T(B) \rightarrow T(A) \vee T(C)$$

$$T(C) \rightarrow \neg T(A) \vee \neg T(B)$$

$$\neg T(C) \rightarrow T(A) \wedge T(B)$$

把上面这些公式化成子句集 S:

- (1)  $\neg T(A) \vee \neg T(B)$
- (2)  $\neg T(A) \vee \neg T(C)$
- (3)  $T(A) \vee T(B) \vee T(C)$
- (4)  $\neg T(B) \vee \neg T(C)$
- (5)  $\neg T(C) \vee \neg T(A) \vee \neg T(B)$
- (6)  $T(C) \vee T(A)$
- (7)  $T(C) \vee T(B)$

下面首先求谁是老实人。把目标公式  $(\exists x)T(x)$  的否定式  $\neg(\exists x)T(x)$  与  $ANSWER(x)$  组成的析取式化为子句加入 S 得到  $S_1$ ，即  $S_1$  比 S 多如下一个子句：

- (8)  $\neg T(x) \vee ANSWER(x)$

应用归结原理进行归结，归结过程如图 4.7 所示。由  $ANSWER(C)$  可得 C 是老实人，即 C 从不说假话。

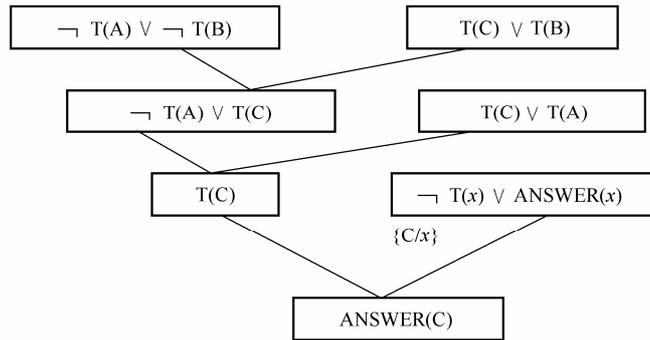


图 4.7 求谁是老实人的归结树

除此之外，无论如何对  $S_1$  进行归结，都推不出  $ANSWER(B)$  和  $ANSWER(A)$ 。

下面来证明 A 和 B 不是老实人。

设 A 不是老实人，则有  $\neg T(A)$ ，把它的否定式  $T(A)$  并入到 S 中，得到子句集  $S_2$ ，即  $S_2$  比 S 多一个子句  $T(A)$ 。

应用归结原理对  $S_2$  进行归结，归结过程如图 4.8 所示。从而证明了 A 不是老实人。同理，可证明 B 也不是老实人。

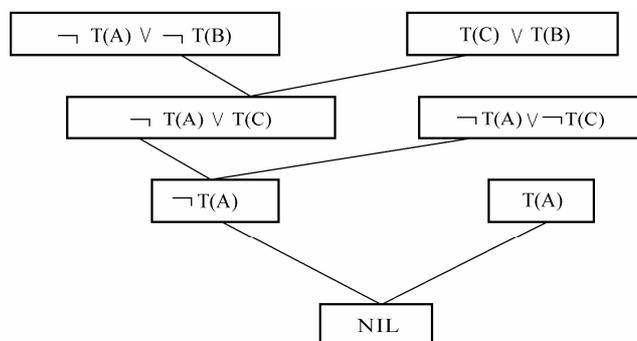


图 4.8 例 4.9 中证明 A 不是老实人的归结树

由上面的例子可以看出，在归结时并不要求把子句集中的全部子句都用到，只要在定理证明时能归结出空子句，在求取问题答案时能归结出 ANSWER 就可以了。另外，在归结过程中，一个子句还可以多次被用来进行归结。

## 4.4 归结反演的改进策略

对子句集进行归结反演时，由于事先不知道哪两个子句可以进行归结，更不知道通过对哪些子句对的归结可以尽快地得到空子句，因而必须对子句集中的所有子句逐对地进行比较，对任何一个可归结的子句对都进行归结。也就是说，在子句集中采用了类似宽度优先搜索的方法来搜索亲本子句，因此归结的效率很低。为此，人们研究了多种归结反演的改进策略。这些归结反演策略大致可分为两类：一类是删除策略，另一类是限制策略。前一类通过删除某些对归结反演过程无用的子句来缩小归结的范围，后一类通过对参加归结的子句进行种种限制，尽可能地减小归结的盲目性，尽快地归结出空子句。

### 4.4.1 删除策略

归结过程是一个不断寻找可归结子句的过程，子句越多，时空花费就越大。如果在归结时先把子句集中的无用子句删除，就会提高归结的效率。删除策略主要有以下几种。

#### 1. 纯文字删除

如果某文字  $L$  在子句集中不存在可与之互补的文字  $\neg L$ ，则称该文字为纯文字。显然，在归结时纯文字不可能被消去，因而用包含它的子句进行归结时不可能得到空子句，即这样的子句对归结是无意义的，所以可以把它所在的子句从子句集中删去而不会影响对子句集的不可满足性的证明。例如，设有子句集：

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

其中， $P$  是纯文字，因此可将子句  $P \vee Q \vee R$  从  $S$  中删去。

#### 2. 重言式删除

如果一个子句中同时包含互补文字时，则称该子句为重言式。例如  $P(x) \vee \neg P(x)$ ， $P(x) \vee$

$Q(x) \vee \neg P(x)$ 都是重言式。重言式是真值为真的子句。不管  $P(x)$ 为真还是为假,  $P(x) \vee \neg P(x)$ 以及  $P(x) \vee Q(x) \vee \neg P(x)$ 都均为真。对于一个子句集来说, 增加或者删去一个真值为真的子句都不会影响它的不可满足性, 因而可从子句集中删去重言式。

### 3. 包孕删除

设有子句  $C_1$  和  $C_2$ , 如果存在一个代换  $\sigma$ , 使得  $C_1\sigma \subseteq C_2$ , 则称  $C_1$  包孕于  $C_2$ 。例如:

$$\begin{aligned} P(x) \text{ 包孕于 } P(y) \vee Q(z) & \quad \sigma = \{y/x\} \\ P(x) \vee Q(A) \text{ 包孕于 } P(f(A)) \vee Q(A) \vee R(y) & \quad \sigma = \{f(A)/x\} \\ P(x) \vee Q(y) \text{ 包孕于 } P(A) \vee Q(u) \vee R(w) & \quad \sigma = \{A/x, u/y\} \end{aligned}$$

把子句集中被包孕的子句删去后, 不会影响子句集的不可满足性, 因而可从子句集中删去被其他子句包孕的子句。

## 4.4.2 限制策略

首先讨论类似宽度优先搜索的归结的一般过程, 然后讨论几种限制策略。

### 1. 归结的一般过程

设有初始子句集

$$S = \{C_1, C_2, C_3, C_4\}$$

其中,  $C_1, C_2, C_3, C_4$ 是  $S$  中的子句。对此子句集进行归结的一般过程是:

①从子句  $C_1$  开始, 逐个与  $C_2, C_3, C_4$  进行比较, 看哪两个子句可进行归结。若能找到, 就求出归结式。然后用  $C_2$  与  $C_3, C_4$  进行比较, 凡可归结的都进行归结, 最后用  $C_3$  与  $C_4$  比较, 若能归结也对它们进行归结。经过这一轮的比较及归结后, 就会得到一组归结式, 称为第一级归结式。

②再从  $C_1$  开始, 用  $S$  中的所有子句分别与第一级归结式中的子句逐个地进行比较、归结, 以及第一级归结式中的子句之间的比较、归结, 这样又会得到一组归结式, 称为第二级归结式。

③仍然从  $C_1$  开始, 用  $S$  中的所有子句及第一级归结式中的所有子句逐个地与第二级归结式中的子句进行比较、归结, 以及第二级归结式中的子句之间的比较、归结, 得到第三级归结式。

如此继续, 只要子句集是不可满足的, 上述归结过程终将归结出空子句而终止。

### 2. 支持集策略

支持集策略对参加归结的子句提出了如下限制: 每一次归结时, 亲本子句中至少应有一个是由目标公式的否定所得到的子句, 或者是它们的后裔。可以证明, 支持集策略是完备的, 即若子句集是不可满足的, 则由支持集策略一定可以归结出空子句。

**例 4.10** 设有初始子句集:

$$S = \{\neg I(x) \vee R(x), I(A), \neg R(y) \vee \neg L(y), L(A)\}$$

其中  $\neg I(x) \vee R(x)$  是目标公式否定得到的子句。

**解:** 用支持集策略进行归结的归结树如图 4.9 所示。

### 3. 线性输入策略

线性输入策略对参加归结的子句提出了如下限制: 参加归结的两个子句中必须至少有一个是初始子句集中的子句。初始子句集是由已知前提与结论的否定组成的子句集。

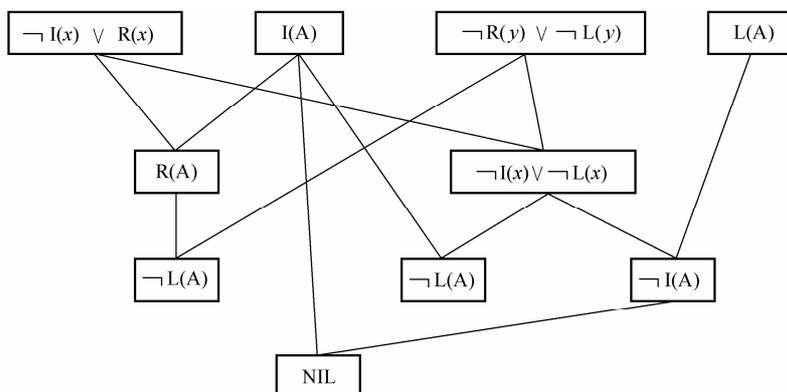


图 4.9 支持集策略归结树

例 4.11 应用线性输入策略对例 4.10 的初始子句集进行归结。

解：用线性输入策略进行归结的归结树如图 4.10 所示。

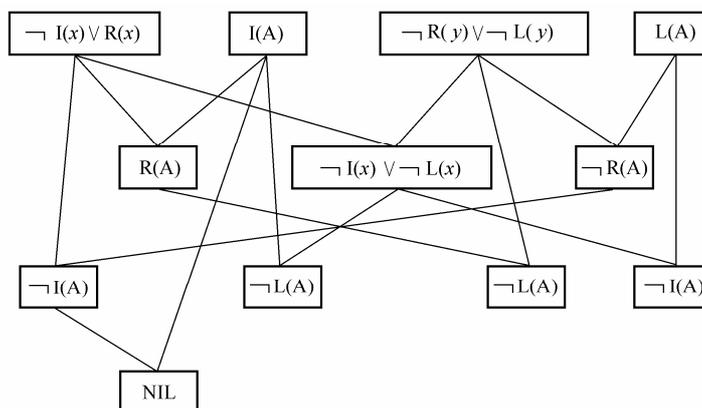


图 4.10 线性输入策略归结树

线性输入策略可限制生成归结式的数量，具有简单、高效的优点，但它是不完备的。也就是说，即使子句集是不可满足的，用线性输入策略进行归结也不一定能归结出空子句。例如，对于子句集

$$S = \{P(x) \vee Q(x), \neg P(y) \vee Q(y), P(u) \vee \neg Q(u), \neg P(t) \vee \neg Q(t)\}$$

可以证明它是不可满足的，但用线性输入策略归结却得不出空子句。

#### 4. 单文字子句策略

如果一个子句只包含一个文字，则称它为单文字子句。单文字子句策略要求参加归结的两个子句中必须有一个是单文字子句。

例 4.12 对例 4.10 给出的初始子句集按单文字子句策略进行归结。

解：用单文字子句策略进行归结的归结树如图 4.11 所示。

用单文字子句策略归结时，归结式将比亲本子句含有较少的文字，这有利于朝着空子句的方向前进，因此它有较强的归结效率。但是，这种归结策略是不完备的。另外，当初始子句中不包含单文字子句时，归结就无法进行。

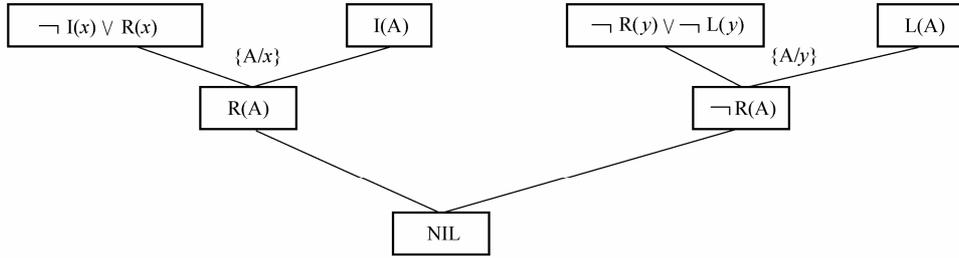


图 4.11 单文字子句策略的归结树

5. 祖先过滤形策略

祖先过滤形策略与线性输入策略比较相似，但放宽了限制。当对两个子句  $C_1$  和  $C_2$  进行归结时，要求它们满足下述两个条件中的任意一个条件：

- ①  $C_1$  与  $C_2$  中至少有一个是初始子句集中的子句。
- ② 如果两个子句都不是初始子句集中的子句，则一个应是另一个的祖先。所谓一个子句（例如  $C_1$ ）是另一个子句（例如  $C_2$ ）的祖先是指  $C_2$  是由  $C_1$  与别的子句归结后得到的后裔归结式。

可以证明祖先过滤形策略是完备的。

**例 4.13** 有子句集  $S = \{\neg P(x) \vee Q(x), \neg P(y) \vee Q(y), P(u) \vee Q(u), P(t) \vee \neg Q(t)\}$ ，用祖先过滤形策略进行归结。

**解：**用祖先过滤形策略进行归结时，归结树如图 4.12 所示。

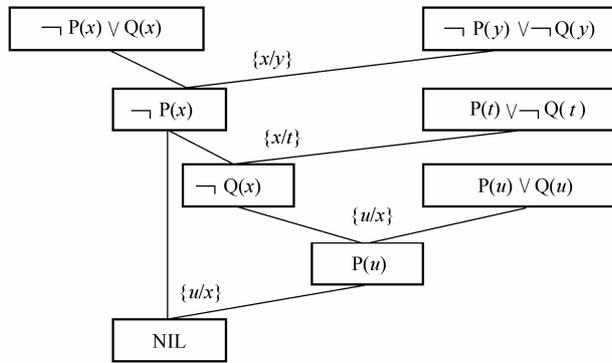


图 4.12 祖先过滤形策略归结树

在此例中，归结出空子句的两个子句是： $C_1 = \neg P(x)$ ， $C_2 = P(u)$ 。其中， $C_1$  是  $C_2$  的祖先。

以上讨论的几种基本的归结策略，在具体应用时可把几种策略组合在一起使用。

### 习题四

4.1 名词解释。

- |       |        |
|-------|--------|
| 演绎推理  | 归纳推理   |
| 确定性推理 | 不确定性推理 |
| 单调推理  | 非单调推理  |

## 基于知识的推理

## 常识性推理

- 4.2 何谓正向推理? 简要说明正向推理的基本过程。
- 4.3 何谓逆向推理? 简要说明逆向推理的基本过程。
- 4.4 何谓模式匹配? 什么是确定性匹配? 什么是不确定性匹配?
- 4.5 何谓冲突消解? 在产生式系统中, 什么情况下会发生冲突? 常用的冲突消解策略有哪些? 请分别简要说明。
- 4.6 什么是公式集的合一? 何谓最一般合一?
- 4.7 何谓归结式? 为什么说归结式是其亲本子句的逻辑结论?
- 4.8 应用归结原理来证明子句集的不可满足性的依据是什么? 如何进行证明?
- 4.9 哪些归结反演的限制策略是完备的?
- 4.10 下述公式集  $F$  是否可合一? 若可合一, 则求出  $F$  的最一般合一。
- (1)  $F = \{P(A, B), P(x, y)\}$
  - (2)  $F = \{P(f(x), B), P(y, z)\}$
  - (3)  $F = \{P(f(x), y), P(y, f(B))\}$
  - (4)  $F = \{P(f(y), y, x), P(x, f(A), f(B))\}$
- 4.11 把下列谓词公式分别化为相应的子句集:
- (1)  $(\forall x)(\forall y)(P(x, y) \wedge Q(x, y))$
  - (2)  $(\forall x)(\forall y)(P(x, y) \rightarrow Q(x, y))$
  - (3)  $(\forall x)(\exists y)(P(x, y) \vee (Q(x, y) \rightarrow R(x, y)))$
  - (4)  $(\forall x)(\forall y)(\exists z)(P(x, y) \rightarrow Q(x, y) \vee R(x, z))$
  - (5)  $(\exists x)(\exists y)(\forall z)(\exists u)(\forall v)(\exists w)(P(x, y, z, u, v, w) \wedge (Q(x, y, z, u, v, w) \vee \neg R(x, z, w)))$
- 4.12 判断下列子句集中哪些是不可满足的:
- (1)  $S = \{\neg P \vee Q, \neg Q, P, \neg P\}$
  - (2)  $S = \{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$
  - (3)  $S = \{P(y) \vee Q(y), \neg P(f(x)) \vee R(A)\}$
  - (4)  $S = \{\neg P(x) \vee Q(x), \neg P(y) \vee R(y), P(A), S(A), \neg S(z) \vee R(z)\}$
  - (5)  $S = \{\neg P(x) \vee \neg Q(y) \vee \neg L(x, y), P(A), \neg R(z) \vee L(A, z), R(B), Q(B)\}$
  - (6)  $S = \{\neg P(x) \vee Q(f(x), A), \neg P(h(y)) \vee Q(f(h(y)), A) \vee \neg P(z)\}$
  - (7)  $S = \{P(x) \vee Q(x) \vee R(x), \neg P(y) \vee R(y), \neg Q(A), \neg R(A)\}$
  - (8)  $S = \{P(x) \vee Q(x), \neg Q(y) \vee R(y), \neg P(z) \vee Q(z), \neg R(u)\}$
- 4.13 对下列各题分别验证  $G$  是否为前提公式集  $F$  的逻辑结论。
- (1)  $F_1: (\exists x)(\exists y) P(x, y)$   
 $G: (\forall y)(\exists x) P(x, y)$
  - (2)  $F_1: (\forall x)(P(x) \wedge (Q(A) \vee Q(B)))$   
 $G: (\exists x)(P(x) \wedge Q(x))$
  - (3)  $F_1: (\exists x)(\exists y)(P(f(x)) \wedge Q(f(y)))$   
 $G: (\exists y)(P(y) \wedge Q(y))$
  - (4)  $F_1: (\forall x)(P(x) \rightarrow (\forall y)(Q(y) \rightarrow \neg L(x, y)))$   
 $F_2: (\exists x)(P(x) \wedge (\forall y)(R(y) \rightarrow L(x, y)))$

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

(5)  $F_1: (\forall x)(P(x) \rightarrow (Q(x) \wedge R(x)))$

$F_2: (\exists x)(P(x) \wedge S(x))$

$G: (\exists x)(S(x) \wedge R(x))$

4.14 某公司招聘工作人员, A、B、C 三人应试。经面试后, 公司表示如下意见:

- ①三人中至少录用一人。
- ②如果录用 A 而不录用 B, 则一定录用 C。
- ③如果录用 B, 则一定录用 C。

求证结论: 公司一定录用 C。

4.15 已知前提: 每个储蓄的人都获得利息。求证结论: 如果没有利息, 那么就沒有人去储蓄。

4.16 已知前提:

- ①某些病人喜欢所有的医生。
- ②没有一个病人喜欢任何一个骗子。

求证结论: 任何一个医生都不是骗子。

4.17 应用归结反演方法证明理发师悖论: 若每个理发师都为不能给自己理发的人理发, 且每个理发师都不为能给自己理发的人理发, 那么不存在任何理发师。

4.18 已知:

- ①如果  $x$  是  $y$  的父亲,  $y$  是  $z$  的父亲, 则  $x$  是  $z$  的祖父。
- ②每个人都有一个父亲。

试用归结演绎推理证明: 对于某人  $u$ , 一定存在一个人  $v$ , 且  $v$  是  $u$  的祖父。

4.19 张某被盗, 公安局派出五个侦察员去调查。研究案情时, 侦察员 A 说“赵与钱中至少有一人作案”; 侦察员 B 说“钱与孙中至少有一人作案”; 侦察员 C 说“孙与李中至少有一人作案”; 侦察员 D 说“赵与孙中至少有一人与此案无关”; 侦察员 E 说“钱与李中至少有一人与此案无关”。如果这五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。

4.20 已知下述事实:

- ①小李只喜欢较容易的课程。
- ②工程类课程是较难的。
- ③PR 系的所有课程都是较容易的。
- ④PR150 是 PR 系的一门课程。

应用归结演绎推理回答问题: 小李喜欢什么课程?

4.21 已知:

- ①能阅读的都是识字的。
- ②海豚不识字。
- ③有些海豚是很聪明的。

分别用线性输入策略、祖先过滤形策略证明: 有些很聪明但不识字。

4.22 已知下述事实:

- ①小杨、小刘和小林是高山俱乐部成员。
- ②高山俱乐部的每个成员是滑雪者或登山者, 或者既滑雪又登山。

- ③没有一个登山者喜欢下雨。
- ④所有滑雪者都喜欢下雪。
- ⑤凡是小杨喜欢的，小刘就不喜欢。
- ⑥凡是小杨不喜欢的，小刘就喜欢。
- ⑦小杨喜欢下雨和下雪。

应用归结演绎推理回答问题：俱乐部是登山者而不是滑雪者的成员是谁？