

第3章

逻辑推理及方法



讲解视频



人物介绍

逻辑推理是对人类思维过程的一种模拟,也是人工智能中的一个重要领域。逻辑推理有多种分类方式,根据逻辑基础的不同可分为演绎推理和归纳推理;按演绎方法的差异又可分为归结演绎推理和非归结演绎推理。归结演绎推理和非归结演绎推理是本章重点讨论的逻辑推理方法。归结演绎推理的理论基础——鲁宾逊归结原理是机器定理证明的基石,它的基本思想是检查子句集 S 中是否包含矛盾:若子句集 S 中包含矛盾,或者能从 S 中推导出矛盾来,就说明子句集 S 是不可满足的。非归结演绎推理可运用的推理规则比较丰富,因篇幅所限,我们仅讨论其中的自然演绎推理和与或形演绎推理的部分方法。

3.1 逻辑推理概述

逻辑推理是证明一个公式可以由一些其他确定的公式按逻辑推理出来的过程。由于这些推理所用的知识与证据都是确定的,因此推出的结论也是确定的,故我们也可将逻辑推理称为确定性推理。本节首先阐述逻辑推理的定义和方法,然后再对推理的控制策略与方向进行介绍。

3.1.1 逻辑推理的定义

人类在对各种问题进行分析与决策时,往往是从已知的事实出发,再通过应用已掌握的知识来找出其中蕴含的事实或归纳出新的结论。逻辑推理所用的事实可分为两种:一种是与求解问题有关的初始证据;另一种是推理过程中所得到的中间结论。本章涉及的逻辑推理是关于从一个永真的前提“必然地”推出一些结论的科学。比如在医疗诊断专家系统中,所有与诊断有关的医疗常识和专家经验都被保存在知识库中。当系统开始工作时,首先需要把病人的症状和检查结果放到事实库中,然后从事实库中的这些初始证据出发,按照某种策略在知识库中寻找可以匹配的知识。如果得到的是一些中间结论,还需要把它们作为新

的事实放入事实库中,并继续寻找可以匹配的知识。如此反复进行,直到找出最终结论为止。至此,我们可以总结出逻辑推理的定义:逻辑推理是以原始证据为出发点,依照某种策略不断地应用知识库中的已有知识,通过逐步推导得出陈述或结论的过程。

在人工智能领域,逻辑推理是非常重要的一个分支,下面我们将从逻辑推理的分类与逻辑推理的控制策略这两个方面来进行阐述。

3.1.2 逻辑推理的分类

逻辑推理方法主要解决在逻辑推理过程中存在的前提与结论之间的逻辑关系。逻辑推理可以有多种不同的分类方法。例如,可以按照推理的逻辑基础、所用知识的确定性、推理过程的单调性,以及推理的方向来分类。

1. 按推理的逻辑基础分类

按照推理的逻辑基础,常用的逻辑推理方法可分为演绎推理和归纳推理。

1) 演绎推理

演绎推理是从已知的一般性知识出发,去推出蕴含在这些已知知识中的适合于某种个别情况下的结论的一种逻辑推理方法。演绎推理按照演绎方法的差异又可分为归结演绎推理和非归结演绎推理,它们是本章重点讨论的逻辑推理方法。演绎推理的核心是三段论,常用的三段论由一个大前提、一个小前提和一个结论组成。其中,大前提是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,并且满足小前提的判断。

例如,有如下判断:

- (1) 软件工程专业的大学生都会编程。
- (2) 小航是软件工程学院的一名学生。
- (3) 小航会编程。

这是一个三段论推理。其中,(1)是大前提,(2)是小前提,(3)是经演绎推出来的结论。从这个例子可以看出,“小航会编程”这一结论是蕴含在“软件工程专业的大学生都会编程”这个大前提中的。因此,演绎推理就是从已知的大前提中推导出满足小前提的判断的结论,即从已知的一般性知识中抽取其所包含的特殊性知识。由此可见,只要大前提和小前提是正确的,则由它们推出的结论也必然是正确的。

2) 归纳推理

归纳推理是从一类事物的大量特殊事例出发,去推出该类事物的一般性结论。它是一种由个别到一般的逻辑推理方法。归纳推理的基本思想是:先从已知事实归纳出一个结论,然后对这个结论的正确性加以证明确认。数学归纳法就是归纳推理的一个典型例子。对于归纳推理,如果按照所选事例的广泛性分类,可分为完全归纳推理和不完全归纳推理;如果按照推理所使用的方法分类,可分为枚举归纳推理和类比归纳推理等。

完全归纳推理是指在进行归纳时需要考察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,来推出该类事物是否具有此属性。例如,某工厂生产一批洗衣机,如果对每台洗衣机都进行了质量检验,并且都合格,则可得出结论:这批洗衣机的质量是合格的。

不完全归纳推理是指在进行归纳时只考察了相应事物的部分对象,就得出了关于该事

物的结论。例如,某工厂生产一批电视机,如果只随机地抽查了其中部分电视机的质量,可根据这些被抽查电视机的质量来推出整批电视机的质量。

枚举归纳推理是指在进行归纳时,如果已知某类事物的有限可数个具体事物都具有某种属性,则可推出该类事物都具有此种属性。设 $a_1, a_2, a_3, \dots, a_i$ 是某类事物 A 中的具体事物,若已知 $a_1, a_2, a_3, \dots, a_i$ 都具有属性 B , 并没有发现反例,那么当 i 足够大时,就可得出“ A 中的所有事物都具有属性 B ”这一结论。

例如,设有如下事例:

小航是软件学院的学生,他会编程;小北是软件学院的学生,她也会编程;小软也是软件学院的学生,他同样会编程……

当这些具体事例足够多时,就可归纳出一个一般性的知识:

凡是软件学院的学生,就一定会编程。

类比归纳推理是指在两个或两类事物有许多属性都相同或相似的基础上,推出它们在其他属性上也相同或相似的一种归纳推理。

设 A 、 B 分别是两类事物的集合

$$A = (a_1, a_2, a_3, \dots, a_i)$$

$$B = (b_1, b_2, b_3, \dots, b_i)$$

并设 a_i 与 b_i 总是成对出现,且当 a_i 有属性 P 时, b_i 就有属性 Q 与之对应,即

$$P(a_i) \rightarrow Q(b_i), \quad i = 1, 2, 3, \dots$$

则当 A 与 B 中有新的元素对出现时,若已知 a' 有属性 P , b' 有属性 Q , 即

$$P(a') \rightarrow Q(b')$$

类比归纳推理的基础是相似原理,其可靠程度取决于两个或两类事物的相似程度,以及这两个或两类事物的相同属性与推出的那个属性之间的相关程度。

3) 演绎推理与归纳推理的区别

演绎推理与归纳推理是两种完全不同的推理方法。演绎推理是在已知领域内的一般性知识的前提下,通过演绎求解一个具体问题或者证明一个给定的结论。这个结论实际上早已蕴含在一般性知识的前提中,演绎推理只不过是将其揭示出来,因此它不能增加新知识。

在归纳推理中,所推出的结论是没有包含在前提内容中的。这种由个别事物或现象推出一般性知识的过程,是增加新知识的过程。例如,一名计算机维修工人,当他刚开始从事这项工作时是一名只有书本知识而无实际经验的维修新手。但当他经过一段时间的工作实践后,就会通过大量实例积累起来一些经验而成为维修技术高超的熟练工人,这些使他成为熟练工人的一般性知识就是采用归纳推理的方式从一个个维修实例中归纳总结出来的。而当他有了这些一般性知识后,就可以运用这些已具备的知识去完成计算机的维修工作,此时这种针对某一台具体的计算机运用一般性知识进行维修的过程就是演绎推理,因为熟练工人仅仅是用经验去解决老问题,并未增加新知识。

2. 按所用知识的确定性分类

按所用知识的确定性,逻辑推理可分为确定性推理和非确定性推理。所谓确定性推理,是指推理所使用的知识和推出的结论都是可以精确表示的,其真值要么为真,要么为假,不会有第三种情况出现。

所谓非确定性推理,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确定的,其真值会位于真与假之间。由于现实世界中的大多数事物都具有一定程度的不确定性,并且这些事物是很难用精确的数学模型来进行表示与处理的,因此非确定性推理也就成了人工智能的一个重要研究课题。非确定性推理问题将在第4章讨论。

3. 按推理过程的单调性分类

按照推理过程的单调性,或者说按照推理过程所得到的结论是否越来越接近目标,逻辑推理可分为单调推理与非单调推理。所谓单调推理是指在推理过程中,每当使用新的知识后,所得到的结论会越接近目标,而不会出现反复情况,即不会由于新知识的加入而否定了前面推出的结论,从而使推理过程又退回先前的某一步。

所谓非单调推理是指在推理过程中,当某些新知识加入后,会否定原来推出的结论,使推理过程退回先前的某一步。非单调推理往往是在知识不完全的情况下发生的。在这种情况下,为使推理能够进行下去,就需要先进行某些假设,并在这些假设的基础上进行推理。但是,当后来由于新的知识加入,发现原来的假设不正确时,就需要撤销原来的假设及由这些假设为基础推出的一切结论,再运用新知识重新进行推理。在我们的日常生活中,经常会遇到非单调推理的情形。例如,当看到名称牌子上露出的一个字“猫”时,我们一般会在脑海中浮现出柔软可爱的小动物的形象,它可以被人当宠物饲养;但之后随着遮挡物的移除,我们看到了牌子的完整内容是“天猫购物”——众所周知,天猫是阿里巴巴集团旗下的购物平台,并不是一只猫,于是我们就取消了之前加入的“猫很可爱,可以当宠物养”的结论,而重新加入“天猫是一个网上购物平台”这一个新的结论。

4. 按推理的方向分类

1) 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法,也称为数据驱动推理或前向链推理。其基本思想是:用户需要事先提供一组初始事实,并将其放入综合数据库 DB 中;推理开始后,推理机根据综合数据库 DB 中的已有事实,到知识库 KB 中寻找当前可用知识,形成一个当前可用知识库 KS;然后按照冲突消解策略,从该知识库 KS 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 DB,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者知识库 KB 中再无可用知识为止。

正向推理过程可用如下算法描述,如图 3-1 所示。

- (1) 把用户提供的初始事实放入综合数据库 DB 中。
- (2) 检查综合数据库 DB 中是否包含了问题的解,若已包含,则求解结束,并成功退出;否则,执行下一步。
- (3) 检查知识库 KB 中是否有可用知识,若有,执行步骤(4);否则,执行步骤(6)。
- (4) 将知识库 KB 中的所有适用知识都选择出来构成当前可用知识库 KS。
- (5) 如果可用知识库 KS 为空,则执行步骤(6);如果可用知识库 KS 不是空集合,就按照某种冲突消解策略选取可用知识库 KS 中的一条知识进行推理,判断推出的事实是否为新事实,若是新事实,则将其加入综合数据库 DB 中并执行步骤(2);否则,将再次执行步骤(5)。
- (6) 询问用户是否可以进一步补充新的事实,若可补充,则将补充的新事实加入综合数

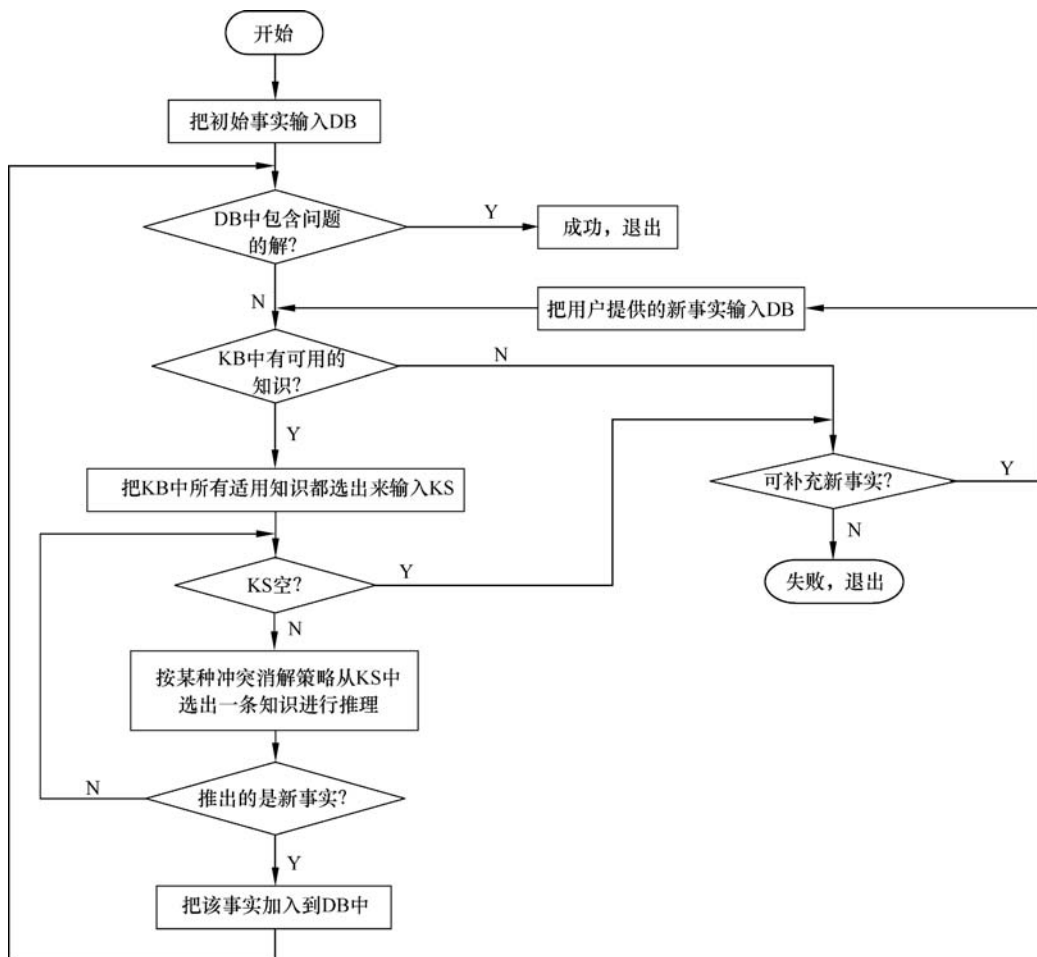


图 3-1 正向推理的算法示意

数据库 DB, 然后执行步骤(3); 否则表示无解, 失败退出。

仅从正向推理的算法来看好像正向推理比较简单, 但实际上, 推理的每步都还有许多工作要做。例如, 如何根据综合数据库 DB 中的事实在知识库 KB 中选取可用知识; 当知识库 KB 中有多条知识可用时, 应该先使用哪条知识等。这些问题涉及知识的匹配方法和冲突消解策略, 如何解决, 后文将进行讨论。正向推理的优点是比较直观, 它允许用户主动提供有用的事实信息, 适用于诊断、设计、预测、监控等领域的问题求解。其主要缺点是推理无明确的目标, 求解问题时可能会执行许多与解无关的操作, 导致推理效率较低。

2) 反向推理

反向推理也称逆向推理, 它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。其基本思想是: 首先根据问题求解的要求提出假设, 将要求证的目标(称为假设)构成一个假设集; 然后从假设集中取出一个假设对其进行验证, 检查该假设是否在综合数据库 DB 中, 是否为用户认可的事实; 当该假设在数据库中时, 该假设成立, 此时若假设集为空, 则成功退出; 若假设不在综合数据库 DB 中, 但可被用户证实为初始事实时, 将该假设放入综合数据库 DB 中, 此时若假设集为空, 则成功退出; 若假设可由

知识库 KB 中的一条或多条知识导出,则将知识库 KB 中所有可以导出该假设的知识构成一个可用知识库 KS,并根据冲突消解策略,从可用知识库 KS 中取出一条知识,将其前提中的所有子条件都作为新的假设放入假设集。重复上述过程,直到假设集为空时成功退出,或假设集非空但可用知识集为空时失败退出为止。

反向推理过程可用如下算法描述,如图 3-2 所示。

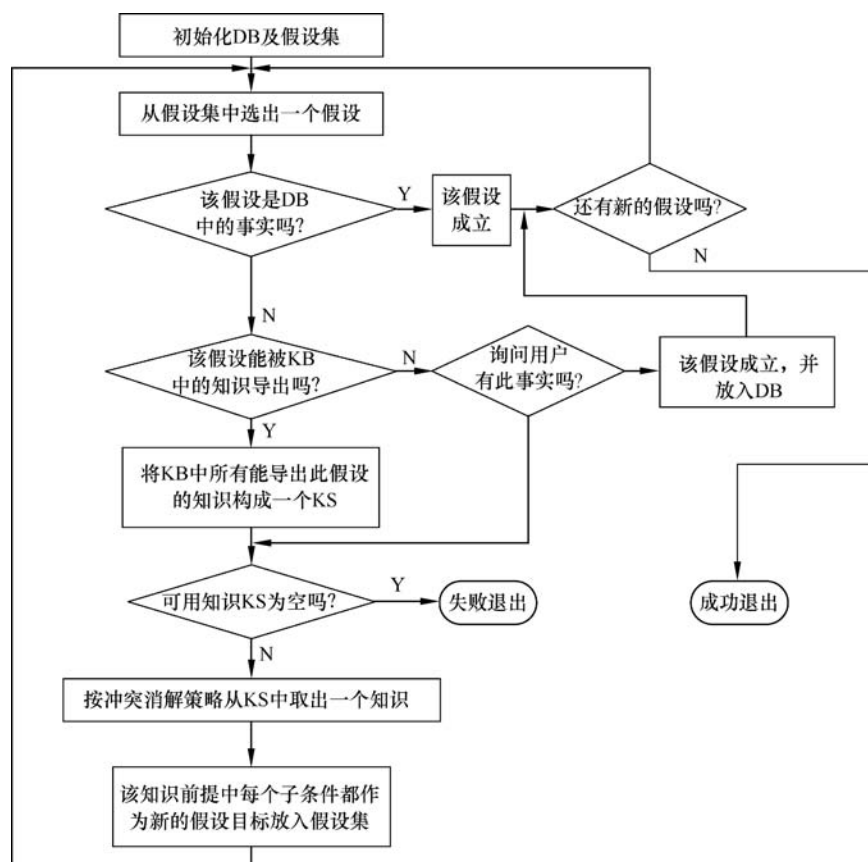


图 3-2 反向推理的算法示意

(1) 将问题的原始证据和欲求证的目标分别放入综合数据库 DB 和假设集。

(2) 从假设集中选出一个假设,检查该假设是否在综合数据库 DB 中。若在,则该假设成立。

此时,若假设集为空,则成功退出;否则,仍执行步骤(2)。若该假设不在数据库 DB 中,则执行下一步。

(3) 检查该假设是否可由知识库 KB 中的某条知识导出。若不能由某条知识导出,则询问用户该假设是否可为用户证实的初始事实。若是,该假设成立,并将其放入综合数据库 DB 中,再重新寻找新的假设;若不是,则执行步骤(5)。若能由某条知识导出,则执行下一步。

(4) 将知识库 KB 中可以导出该假设的所有知识构成一个可用知识库 KS。

(5) 检查可用知识库 KS 是否为空,若空,失败退出;否则,执行下一步。

(6) 按冲突消解策略从可用知识库 KS 中取出一条知识,继续执行下一步。

(7) 将该知识的前提中的每个子条件都作为新的假设目标放入假设集,执行步骤(2)。

反向推理的主要优点是不必寻找和使用那些与假设目标无关的信息和知识,推理过程的目标明确,也有利于向用户提供解释,在诊断性专家系统中较为有效。其主要缺点是当用户对解的情况认识不清时,由系统自主选择假设目标的盲目性比较大,若选择不好,可能需要多次提出假设,会影响系统效率。

3) 双向推理

在定理证明中,经常采用双向推理。所谓双向推理是指正向推理与反向推理同时进行,且在推理过程中的某一步骤上“碰头”的一种推理。其基本思想是:一方面根据已知事实进行正向推理,但并不推导至最终的目标;另一方面从某假设目标出发进行反向推理,也无须推导至初始事实,而是让它们在中间相遇。即由正向推理所得到的中间结论恰好是反向推理此时所要求的证据,这时推理就可以结束,反向推理时所做的假设就是推理的最终的目标。

双向推理的困难在于“碰头”的判断。另外,如何权衡正向推理与反向推理的比重,即如何确定“碰头”的时机也是一个困难的问题。

4) 混合推理

由以上讨论可知,正向推理和反向推理都有各自的优缺点。当问题较复杂时,单独使用其中任何一种,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,避免各自的短处,取长补短,可以将它们结合起来使用。这种把正向推理和反向推理结合起来进行的推理称为混合推理。混合推理可以有多种具体的实现方法。例如,可以采用先正向推理,后反向推理的方法;也可以采用先反向推理,后正向推理的方法;还可以采用随机选择正向推理和反向推理的方法。由于这些方法仅是正向推理和反向推理的某种结合,因此对这 3 种情况不再进行讨论。

3.1.3 逻辑推理的控制策略

逻辑推理不仅依赖于所用的推理方法,也依赖于推理的控制策略。推理的控制策略是指如何使用领域知识使推理过程尽快达到目标的策略。由于智能系统的推理过程一般表现为一种搜索过程,因此,推理的控制策略又可分为推理策略和搜索策略。其中,推理策略主要解决推理方向、冲突消解等问题,如推理方向控制策略、求解策略、限制策略、冲突消解策略等;搜索策略主要解决推理路线、推理效果、推理效率等问题。

控制策略用来确定推理的控制方式,即推理过程是从初始证据开始到目标,还是从目标开始到初始证据。按照对推理方向的控制,推理可分为正向推理、反向推理和混合推理等。无论哪一种推理方式,系统都需要有一个存放知识的知识库、一个存放初始事实及中间结果的综合数据库和一个用于推理的推理机。求解策略是指只求一个解,还是求所有解或最优解等。限制策略是指对推理的深度、宽度、时间、空间等进行的限制。冲突消解策略是指当推理过程中有多条知识可用时,如何从这多条可用知识中选出一条最佳知识用于推理的策略。常用的冲突消解策略有领域知识优先和新鲜知识优先等。所谓领域知识优先,是指把领域问题的特点作为选择知识的依据;而新鲜知识优先,是指把知识前提条件中事实的新

鲜性作为选择知识的依据。例如,综合数据库中后生成的事实比先生成的事实具有更大的新鲜性。

对于推理的控制策略中所包含的推理策略和搜索策略,本章主要讨论推理策略,搜索策略将在第5章讨论。

3.2 逻辑推理的基础

在前文的论述中,已经大概了解了逻辑推理的分类以及知识表示,在2.3节中对谓词逻辑也进行了相关的阐述,谓词逻辑是一种形式语言,是人工智能中一种常用的知识表示方法。接下来我们将主要讨论逻辑推理所需要的谓词逻辑基础。

3.2.1 谓词公式

如果 P 是一个不能再分解的 n 元谓词变元, $x_1, x_2, x_3, \dots, x_n$ 是个体变元,那么我们称 $P(x_1, x_2, x_3, \dots, x_n)$ 是原子公式或谓词公式。当 n 为 0 时, P 表示命题变元或原子命题公式,故可以说谓词逻辑是更为广泛的一个定义,我们根据下述规则得到谓词公式:

- (1) 单个谓词是谓词公式,我们通常称其为原子谓词公式。
- (2) 如果 A 是谓词公式,那么 $\neg A$ 也应该是谓词公式。
- (3) 如果 A, B 是谓词公式,那么其连接词组合 $A \vee B, A \wedge B, A \rightarrow B, A \leftrightarrow B$ 均为谓词公式。
- (4) 如果 A 是谓词公式,那么其量词组合如 $\forall x A, (\exists x) A$ 同样也为谓词公式。
- (5) 当且仅当使用有限次规则(1)~(4)时,得到的公式才仍然是谓词公式。

谈及谓词公式,我们不可避免地要了解量词的辖域与变元的约束。在一个公式中,如果有量词出现,那么位于量词后面的单个谓词或者用括号括起的谓词公式就称为该量词的辖域。之所以要定义辖域的概念,是因为在量词的辖域内所有与量词同名的变元都有一个统一的名称——约束变元。同理,在量词的辖域内,与量词无关的变元就称为非约束变元或者自由变元,例如

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

其中, $P(x) \rightarrow (\exists y) Q(x, y, z)$ 是 $\forall x$ 的辖域,辖域内的变元 x 是受 $\forall x$ 约束的变元。同样地, $Q(x, y, z)$ 是 $\exists y$ 的辖域,则变元 y 是受 $\exists y$ 约束的变元,而公式中的 z 则不受任何约束,即非约束变元。这里要注意,在谓词逻辑中,变元的具体名称是不受太多束缚的,我们可以任意地对谓词公式中的变元名进行自由替换,但在更名的过程中要注意:必须把量词辖域内所有同名的约束变元都改成相同的名称,同时更改的名称不能与辖域内的非约束变元的名称重复;当对辖域内的非约束变元改名时,也要注意不能将其名称与约束变元的名称重复。像 $(\exists y) Q(x, y, z)$ 改名为 $(\exists t) Q(m, t, n)$,其中将约束变元 y 更名为 t ,将非约束变元 x 与 z 更名为 m 和 n ,这是完全符合规则的。

在命题逻辑中,命题公式的一个解释就是指对该命题公式中各个命题变元的一次真值指派。有了命题公式的解释,只要命题确定了,就可以根据这个解释及其连接词的定义求出该命题公式的真值。但谓词逻辑就不一样了,由于谓词公式中可能包含个体常量、个体变元或函数,因此不能像命题公式那样直接通过真值指派给出解释,必须先考虑个体常量和函数

在个体域上的取值,然后才能根据常量与函数的具体取值为谓词分别指派真值。综上所述,存在许多种组合情况,所以一个谓词公式极有可能有许多种不同的解释,且根据每一个解释,谓词公式均可以求出一个真值 T 或者 F。下面介绍谓词公式的永真性、可满足性和不可满足性。

定义 3.1 如果谓词公式 P 对非空个体域 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 在任何非空个体域上均是永真的,则称 P 永真。

定义 3.2 如果谓词公式 P 对非空个体域 D 上的任一解释都取真值 F,则称 P 在 D 上是永假的;如果 P 在任何非空个体域上均是永假的,则称 P 永假。谓词公式的永假性又称不可满足性或不相容性。

由以上定义可以看出,要判定一个谓词公式为永真或者永假,必须对每个非空个体域上的每个解释逐一进行判断。当解释的个数有限时,尽管工作量大,公式的永真性和永假性费些力还是可以判断的;但当解释个数无限时,其永真性和永假性就很难判断了。

定义 3.3 对于谓词公式 P,如果至少存在 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,反之称公式 P 是不可满足的。谓词公式的可满足性也称为相容性。

下面介绍谓词公式的等价性,谓词公式的等价性可以用相应的等价式来表示,由于这些等价式是演绎推理的主要依据,所以也称它们为推理规则。

谓词公式的等价式的定义如下。

定义 3.4 设 P 与 Q 是 D 上的两个谓词公式,若对 D 上的任意解释,P 与 Q 都有相同的真值,则称 P 与 Q 在 D 上是等价的。如果 D 是任意非空个体域,则称 P 与 Q 是等价的,记为 $P \Leftrightarrow Q$ 。

本书中常用的一些等价式如下。

(1) 对合律(双重否定律):

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

(2) 交换律:

$$\begin{aligned} P \vee Q &\Leftrightarrow Q \vee P \\ P \wedge Q &\Leftrightarrow Q \wedge P \end{aligned}$$

(3) 结合律:

$$\begin{aligned} (P \vee Q) \vee R &\Leftrightarrow P \vee (Q \vee R) \\ (P \wedge Q) \wedge R &\Leftrightarrow P \wedge (Q \wedge R) \end{aligned}$$

(4) 分配律:

$$\begin{aligned} P \vee (Q \wedge R) &\Leftrightarrow (P \vee Q) \wedge (P \vee R) \\ P \wedge (Q \vee R) &\Leftrightarrow (P \wedge Q) \vee (P \wedge R) \end{aligned}$$

(5) 德·摩根律:

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

(6) 吸收律:

$$\begin{aligned} P \vee (P \wedge Q) &\Leftrightarrow P \\ P \wedge (P \vee Q) &\Leftrightarrow P \end{aligned}$$

(7) 否定律:

$$\begin{aligned} P \vee \neg P &\Leftrightarrow T \\ P \wedge \neg P &\Leftrightarrow F \end{aligned}$$

(8) 逆否律:

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

(9) 连接词化规律:

$$\begin{aligned} P \rightarrow Q &\Leftrightarrow \neg P \vee Q \\ P \leftrightarrow Q &\Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P) \\ P \leftrightarrow Q &\Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q) \end{aligned}$$

(10) 量词转化规律:

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

(11) 量词分配规律:

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

下面介绍谓词公式的永真蕴含性。谓词公式的永真蕴含性可以用相应的永真蕴含式来表示,由于这些永真蕴含式是演绎推理的主要依据,所以它们也是推理规则的一种。

谓词公式的永真蕴含式可定义如下。

定义 3.5 对谓词公式 P 和 Q ,如果 $P \rightarrow Q$ 永真,则称 P 永真蕴含 Q ,且称 Q 为 P 的逻辑结论, P 为 Q 的前提,记作 $P \Rightarrow Q$ 。

本书中常用的永真蕴含式如下。

(1) 附加律:

$$\begin{aligned} P &\Rightarrow P \vee Q \\ Q &\Rightarrow P \vee Q \\ Q &\Rightarrow P \rightarrow Q \end{aligned}$$

(2) 化简律:

$$\begin{aligned} P \wedge Q &\Rightarrow P \\ P \wedge Q &\Rightarrow Q \end{aligned}$$

(3) 假言推理:

$$P, P \rightarrow Q \Rightarrow Q$$

即由 P 为真和 $P \rightarrow Q$ 为真,可以推出 Q 为真。

(4) 拒取式推理:

$$P \rightarrow Q, \neg Q \Rightarrow \neg P$$

即由 $P \rightarrow Q$ 为真和 Q 为假,可以推出 P 为假。

(5) 假言三段论:

$$P \rightarrow Q, Q \rightarrow R \Rightarrow P \rightarrow R$$

即由 $P \rightarrow Q$ 为真和 $Q \rightarrow R$ 为真,可以推出 $P \rightarrow R$ 为真。

(6) 析取三段论:

$$\neg P, P \vee Q \Rightarrow Q$$

(7) 二难推理:

$$P \vee Q, P \rightarrow R, Q \rightarrow R \Rightarrow R$$

(8) 存在固化:

$$(\exists x)P(x) \Rightarrow P(y)$$

注意: y 是个体域中的某一个可以使得 $P(y)$ 为真的个体, 利用此永真蕴含式可消去谓词公式中的存在量词。

(9) 全称固化:

$$(\forall x)P(x) \Rightarrow P(y)$$

注意: y 是个体域中的任意个体, 利用此永真蕴含式可消去谓词公式中的全称量词。

上面给出的等价式和永真蕴含式是进行演绎推理的重要依据, 因此这些公式也称为推理规则。除了这些公式以外, 我们在 3.3 节的归结演绎推理中, 还需要将反证法推广到谓词公式集。接下来再介绍 3 条在谓词逻辑中非常重要的推理规则。

P 规则: 只要是在推理的过程中, 那么在任意步骤前后均可以引入前提假设。

T 规则: 在推理的过程中, 若前面步骤中有一个或几个公式是永真蕴含式 M 的, 那么我们就可以将公式 M 引入推理过程。

CP 规则: 若能从 N (N 为任意引入的命题) 和前提集合中推出结论 M 来, 那么就能够从前提集合中推出 $N \rightarrow M$ 。

3.2.2 谓词公式的范式

在实际操作过程中, 谓词公式的形式千变万化, 这就给谓词的演算带来了很大的困难。为了简化谓词演算, 我们将谓词公式在不失其原始语义的情况下进行标准变形, 使其成为某种标准形, 而这种标准形就是范式。即范式是谓词公式的标准形式, 谓词公式往往需要变换为同它等价的范式, 以便对它们进行一般性处理, 从而简化对谓词公式的研究。在谓词逻辑中, 根据量词在谓词公式中出现的位置不同, 在谓词演算中, 我们可将谓词公式的范式分为两种。

1. 前束范式

定义 3.6 设 F 为一个谓词公式, 如果其中的所有量词均非否定地出现在公式的最前面, 而它们的辖域为整个公式, 则称 F 为前束范式。一般地, 前束范式可写成

$$(Q_1 x_1)(Q_2 x_2) \cdots (Q_n x_n) M(x_1, x_2, x_3, \cdots, x_n)$$

式中, $Q_i (i=1, 2, 3, \cdots, n)$ 为前缀, 它是一个由全称量词或存在量词组成的量词串; $M(x_1, x_2, x_3, \cdots, x_n)$ 为母式, 它是一个不含任何量词的谓词公式。

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

任意含有量词的谓词公式均可化为与其对应的前束范式, 其化简方法将在后文子句集的化简中讨论。

2. 斯克林范式

定义 3.7 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斯克林(Skolem)范式。

例如, $(\exists x)(\exists z)(\forall y)(P(x) \vee Q(y, z) \wedge R(x, z))$ 是斯克林范式。

任意含有量词的谓词公式均可化为与其对应的斯克林范式, 其化简方法也将后文子句集的化简中讨论。

3.2.3 置换与合一

在不同的谓词公式中, 往往会出现谓词名相同但其个体不同的情况, 此时逻辑推理过程是不能直接进行匹配的, 需要先进行置换。例如, 可根据全称固化推理和假言推理由谓词公式

$$P(y) \text{ 和 } (\forall x)(P(x) \rightarrow Q(x))$$

推出 $Q(y)$, 这对谓词 $P(y)$ 而言可以看作是全称固化推理, 即由 $(\forall x)P(x) \Rightarrow P(y)$ 推出来的, 其中 y 是任意个体常量。想要使用假言推理, 需要先找到项 y 对变元 x 的置换, 使 $P(y)$ 与 $P(x)$ 一致。类似这种寻找项对变元的置换从而将谓词统一化的过程, 就称为合一。下面详细讨论置换与合一的有关概念及方法。

1. 置换

置换(substitution)可以简单地理解为是在一个谓词公式中用置换项去替换变元。其形式定义如下。

定义 3.8 置换是形如

$$\left\{ \frac{m_1}{x_1}, \frac{m_2}{x_2}, \frac{m_3}{x_3}, \dots, \frac{m_n}{x_n} \right\}$$

的有限集合。其中, $m_1, m_2, m_3, \dots, m_n$ 是项, $x_1, x_2, x_3, \dots, x_n$ 是互不相同的变元。 $\frac{m_i}{x_i}$ 表示用 m_i 置换 x_i , 并且要求 m_i 与 x_i 不相同。

例如

$$\left\{ \frac{a}{x}, \frac{b}{y}, \frac{f(c)}{z} \right\}$$

是一个置换。但是

$$\left\{ \frac{g(y)}{x}, \frac{f(x)}{y} \right\}$$

则不是一个置换, 原因是在 x 与 y 之间出现了循环置换现象。置换的目的是将某些变元用另外的变元、常量或函数取代, 使其不在公式中出现。但在 $\left\{ \frac{g(y)}{x}, \frac{f(x)}{y} \right\}$ 中, 用 $g(y)$ 置换 x , 又用 $f(g(y))$ 置换 y , 既没有消去 x , 也没有消去 y 。若改为

$$\left\{ \frac{g(a)}{x}, \frac{f(x)}{y} \right\}$$

就可以了, 因为此时用 $f(g(a))$ 来置换变元 y , 从而消去了 x 和 y 。

通常, 置换是用希腊字母 $\theta, \delta, \tau, \alpha$ 来表示的。

定义 3.9 设 $\theta = \left\{ \frac{m_1}{x_1}, \frac{m_2}{x_2}, \frac{m_3}{x_3}, \dots, \frac{m_n}{x_n} \right\}$ 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 $m_i (i=1, 2, 3, \dots, n)$, 得到一个新的公式 G , 称 G 为 F 在置换 θ 下的例示, 记作 $G=F\theta$ 。

显然, 一个谓词公式中的任何例示都是该公式的逻辑结论。

下面介绍求两个置换合成的方法。

定义 3.10 设

$$\delta = \left\{ \frac{p_1}{x_1}, \frac{p_2}{x_2}, \frac{p_3}{x_3}, \dots, \frac{p_n}{x_n} \right\}$$

$$\theta = \left\{ \frac{q_1}{y_1}, \frac{q_2}{y_2}, \frac{q_3}{y_3}, \dots, \frac{q_m}{y_m} \right\}$$

是两个置换, 且 δ 与 θ 的合成也是一个置换, 记作 $\theta \circ \delta$, 它是从集合

$$\left\{ \frac{p_1\theta_1}{x_1}, \frac{p_2\theta_2}{x_2}, \frac{p_3\theta_3}{x_3}, \dots, \frac{p_n\theta_n}{x_n}, \frac{q_1}{y_1}, \frac{q_2}{y_2}, \frac{q_3}{y_3}, \dots, \frac{q_m}{y_m} \right\}$$

中删去以下两类元素:

(1) 当 $p_i\theta_i = x_i$ 时, 删去 $\frac{p_i\theta_i}{x_i} (i=1, 2, 3, \dots, n)$;

(2) 当 $y_j \in \{x_1, x_2, x_3, \dots, x_n\}$ 时, 删去 $\frac{q_j}{y_j} (j=1, 2, 3, \dots, m)$ 。

后剩下元素所构成的集合, 其中 $p_i\theta$ 表示对 p_i 运用 θ 置换, 实际上 $\theta \circ \delta$ 就是对一个公式先运用 θ 置换, 再运用 δ 置换。接下来看一个求置换合成的例子。

例 3.1 设 $\delta = \left\{ \frac{f(y)}{x}, \frac{z}{y} \right\}, \theta = \left\{ \frac{a}{x}, \frac{b}{y}, \frac{y}{z} \right\}$, 求 δ 与 θ 的合成。

解: 先求出集合

$$\left\{ \frac{f\left(\frac{b}{y}\right)}{\frac{x}{y}}, \frac{\left(\frac{y}{z}\right)}{\frac{y}{y}}, \frac{a}{x}, \frac{b}{y}, \frac{y}{z} \right\} = \left\{ \frac{f(b)}{x}, \frac{y}{y}, \frac{a}{x}, \frac{b}{y}, \frac{y}{z} \right\}$$

式中, $\frac{f(b)}{x}$ 中的 $f(b)$ 是置换 θ 作用于 $f(y)$ 的结果; $\frac{y}{y}$ 中的 y 是置换 θ 作用于 z 的结果。

在该集合中, $\frac{y}{y}$ 满足定义 3.10 中的条件(1), 需要删除; $\frac{a}{x}$ 和 $\frac{b}{y}$ 满足定义中的条件(2), 也需要删除。删除整理后得

$$\delta \circ \theta = \left\{ \frac{f(b)}{x}, \frac{y}{z} \right\}$$

即为所求。

2. 合一

什么是合一呢? 其实很简单, 合一(unifier)可以理解为寻找项对变量的置换, 使两个谓词公式一致。通常来讲, 一个公式集的合一往往不是唯一的。合一的形式定义如下。

定义 3.11 设有公式集 $F = \{F_1, F_2, F_3, \dots, F_n\}$, 若存在一个置换 θ , 可使 $F_1\theta = F_2\theta = F_3\theta = \dots = F_n\theta$, 则称 θ 是 F 的一个合一, 此时称 $F_1, F_2, F_3, \dots, F_n$ 是可合一的。

例如, 设有公式集 $F = \{P(x, y, f(y)), P(a, g(x), z)\}$, 那么

$$\theta = \left\{ \frac{a}{x}, \frac{g(a)}{y}, \frac{f(g(a))}{z} \right\}$$

满足上述定义,故 θ 是公式集 F 的一个合一。

定义 3.12 设 θ 是公式集 F 的一个合一。如果对 F 的任意一个合一 δ ,都存在一个置换 λ ,使得

$$\delta = \theta \circ \lambda$$

则称 θ 是 F 的最一般合一。

一个公式集的最一般合一是唯一的。若用最一般合一置换那些可合一的谓词公式,则可使它们变成完全一致的谓词公式,即一模一样的字符串。那么如何求取最一般合一呢?我们需要先引入差异集的概念。差异集是指两个公式中相同位置处有不同符号的集合。

例如,有两个谓词公式

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

$$F_2: P(x, f(a), h(b))$$

分别从 F_1 和 F_2 的第一个符号开始,逐项向右比较,此时可发现 F_1 中的 y 与 F_2 中的 $f(a)$ 不同。再继续比较,又可知 F_1 中的 z 与 F_2 中的 $h(b)$ 不同。于是可得到两个差异集

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

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

求公式集 F 的最一般合一的算法如下:

- (1) 令 $k=0$ 、 $F_k=F$ 、 $\sigma_k=\epsilon$, ϵ 代表空置换。 F 为欲求最一般合一的公式集。
- (2) 若 F_k 只含一个表达式,则算法停止。 σ_k 就是最一般合一;否则,执行步骤(3)。
- (3) 找出 F_k 的差异集 D_k 。
- (4) 若 D_k 中存在元素 x_k 和 t_k ,其中 x_k 是变元, t_k 是项,且 x_k 不在 t_k 中出现,则进行

$$\sigma_{k+1} = \sigma_k \circ \left\{ \frac{t_k}{x_k} \right\}$$

$$F_{k+1} = F_k \left\{ \frac{t_k}{x_k} \right\}$$

$$k = k + 1$$

然后执行步骤(2)。若不存在这样的 x_k 和 t_k ,则执行步骤(5)。

- (5) 算法终止。 F 的最一般合一不存在。下面来看一个求最一般合一的例子。

例 3.2 求出下面公式集的最一般合一:

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

解:

- (1) 令 $F_0=F$, $\sigma_0=\epsilon$ 。 F_0 中有两个表达式,所以 σ_0 不是最一般合一。
- (2) 得到差异集 $D_0 = \{a, z\}$ 。
- (3)

$$\sigma_1 = \sigma_0 \circ \left\{ \frac{a}{z} \right\} = \left\{ \frac{a}{z} \right\}$$

$$F_1 = \{P(a, x, f(g(y))), P(a, f(a), f(u))\}$$

- (4) 得到差异集 $D_1 = \{x, f(a)\}$ 。

(5)

$$\sigma_2 = \sigma_1 \circ \left\{ \frac{f(a)}{x} \right\} = \left\{ \frac{a}{z}, \frac{f(a)}{x} \right\}$$

$$F_2 = F_1 \left\{ \frac{f(a)}{x} \right\} = \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}$$

(6) 得到差异集 $D_2 = \{g(y), u\}$ 。

(7)

$$\sigma_3 = \sigma_2 \circ \left\{ \frac{g(y)}{u} \right\} = \left\{ \frac{a}{z}, \frac{f(a)}{x}, \frac{g(y)}{u} \right\}$$

$$F_3 = F_2 \left\{ \frac{g(y)}{u} \right\} = \{P(a, f(a), f(g(y)))\}$$

(8) 因为 F_3 中只有一个表达式, 所以 σ_3 就是最一般合一。

(9) 所求最一般合一为

$$\left\{ \frac{a}{z}, \frac{f(a)}{x}, \frac{g(y)}{u} \right\}$$

3.3 归结演绎推理

归结演绎推理是一种基于鲁宾逊归结原理的机器推理方法, 它在人工智能、逻辑编程、定理证明和数据库理论等诸多领域都有广泛的应用。归结原理也称为消解原理, 是鲁宾逊于 1965 年在海伯伦 (Herbrand) 原理的基础上提出的一种基于逻辑“反证法”的机械化定理证明方法。在人工智能中, 几乎所有的问题都可以转化为一个定理证明问题。而定理证明的实质, 就是要对前提 P 和结论 Q , 证明 $P \rightarrow Q$ 永真。由 3.2 节可知, 要证明 $P \rightarrow Q$ 永真, 就是要证明 $P \rightarrow Q$ 在任何一个非空的个体域上都是永真的。这将是非常困难的, 甚至是不可实现的。为此, 人们进行了大量的探索, 后来发现可以采用反证法的思想, 把关于永真性的证明转化为关于不可满足性的证明。即要证明 $P \rightarrow Q$ 永真, 只要能够证明 $P \wedge \neg Q$ 为不可满足即可, 这正是归结演绎推理的基本出发点。

3.3.1 子句集

由于鲁宾逊归结原理是在子句集的基础上进行定理证明的, 因此, 在讨论这些方法之前, 需要先介绍子句集的有关概念。

1. 子句和子句集

定义 3.13 原子谓词公式及其否定统称为文字。

例如, $P(x)$ 、 $Q(x)$ 、 $\neg P(x)$ 、 $\neg Q(x)$ 等都是文字。

定义 3.14 任何文字的析取式称为子句。

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

定义 3.15 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。空子句一般记为 NIL。

定义 3.16 由子句或空子句构成的集合称为子句集。

2. 子句集的化简

在谓词逻辑中,任何一个谓词公式都可以通过应用等价关系及推理规则化成相应的子句集。其化简步骤如下。

(1) 消去连接词“ $\rightarrow$ ”和“ $\leftrightarrow$ ”。

反复使用如下等价公式:

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

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

就能消去谓词公式中的连接词“ $\rightarrow$ ”和“ $\leftrightarrow$ ”。

如公式

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

经等价变换后变为

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

(2) 减少否定符号的辖域。

反复使用双重否定律

$$\neg(\neg P) \Leftrightarrow P$$

德·摩根律

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

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

量词转化规律

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

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

将每个否定符号“ $\neg$ ”移动到仅靠谓词之后的位置,使得每个否定符号最多仅仅作用在一个谓词上。

例如,(1)中所得公式经过本变换后为

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

(3) 对变元标准化。

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没有出现过的变元代替,使不同量词约束的变元有不同的名字。

例如,上步所得公式经本变换后为

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

(4) 化为前束范式。

化为前束范式的方法是把所有量词都移到公式的左边,并且在移动时不能改变其相对顺序。由于在(3)中已对变元进行了标准化,每个量词都有自己的变元,这就消除了任何由变元引起冲突的可能,因此这种移动是可行的。

例如,上步所得公式化为前束范式后为

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

(5) 消去存在量词。

消去存在量词时,需要区分以下两种情况。

若存在量词不出现全称量词的辖域内(即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。

若该存在量词位于一个或多个全称量词的辖域内,例如

$$(\forall x_1)(\forall x_2)\cdots(\forall x_n)(\exists y)P(x_1, x_2, x_3, \cdots, x_n, y)$$

则需要用斯克林函数 $f(x_1, x_2, x_3, \cdots, x_n)$ 替换受该存在量词约束的变元,然后再消去该存在量词。

例如,在上步所得公式中存在量词 $\exists y$ 和 $\exists z$ 都位于 $\forall x$ 的辖域内,因此都需要用斯克林函数来替换。设替换 y 和 z 的斯克林函数分别是 $f(x)$ 和 $g(x)$,则替换后的公式为

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

(6) 化为斯克林标准形。

斯克林标准形的一般形式为

$$(\forall x_1)(\forall x_2)(\forall x_3)\cdots(\forall x_n)M(x_1, x_2, x_3, \cdots, x_n)$$

式中, $M(x_1, x_2, x_3, \cdots, x_n)$ 是斯克林标准形的母式,它由子句的合取所构成。

把谓词公式化为斯克林标准形需要使用以下等价关系:

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

例如,上步所得的公式化为斯克林标准形后为

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

(7) 消去全称量词。

由于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,因此可以消去全称量词。但剩下的母式,仍假设其变元是被全称量词量化的。

例如,上步所得公式消去全称量词后为

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

(8) 消去合取词。

在母式中消去所有合取词,把母式用子句集的形式表示出来。其中,子句集中的每一个元素都是一个子句。

例如,上步所得公式的子句集中包含以下两个子句:

$$\begin{aligned} & \neg P(x, f(x)) \vee Q(x, g(x)) \\ & \neg P(x, f(x)) \vee \neg R(x, g(x)) \end{aligned}$$

(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。由于每一个子句都对应母式中的一个合取元,并且所有变元都是由全称量词量化的,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(8)中所得公式,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:

$$\begin{aligned} & \neg P(x, f(x)) \vee Q(x, g(x)) \\ & \neg P(y, f(y)) \vee \neg R(y, g(y)) \end{aligned}$$

3. 子句集的应用

通过上述化简步骤,可以将谓词公式化简为一个标准子句集。由于在消去存在量词时所用的斯克林函数可以不同,因此化简后的标准子句集是不唯一的。这样,当原谓词公式为非永假时,它与其标准子句集并不等价。但是,当原谓词公式为永假(即不可满足)时,其标准子句集则一定是永假的,即斯克林标准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。

在证明此定理之前,先进行如下说明。

为方便讨论问题,设给定的谓词公式 F 已为前束型

$$(Q_1 x_1) \cdots (Q_r x_r) \cdots (Q_n x_n) M(x_1, x_2, x_3, \cdots, x_n)$$

式中, $M(x_1, x_2, x_3, \cdots, x_n)$ 已化为合取范式。由于将 F 化为这种前束型是一种很容易实现的等值运算,因此这种假设是可以的。

又设 $Q_r x_r$ 是第一个出现的存在量词 $\exists x_r$, 即 F 为

$$F = (\forall x_1)(\forall x_2) \cdots (\forall x_{r-1})(\exists x_r) \cdots (Q_{r+1} x_{r+1}) \cdots (Q_n x_n) \\ M(x_1, \cdots, x_{r-1}, x_r, x_{r+1}, \cdots, x_n)$$

为把 F 化为斯克林标准形,需要先消去这个 $\exists x_r$, 并引入斯克林函数,得到

$$F_1 = (\forall x_1)(\forall x_2) \cdots (\forall x_{r-1}) \cdots (Q_{r+1} x_{r+1}) \cdots (Q_n x_n) \\ M(x_1, \cdots, x_{r-1}, f(x_1, \cdots, x_{r-1}), x_{r+1}, \cdots, x_n)$$

若能证明

$$F \text{ 不可满足} \Leftrightarrow F_1 \text{ 不可满足}$$

则同理可证

$$F_1 \text{ 不可满足} \Leftrightarrow F_2 \text{ 不可满足}$$

重复这一过程,直到证明了

$$F_1 \text{ 不可满足} \Leftrightarrow F_m \text{ 不可满足}$$

为止。此时, F_m 已为 F 的斯克林标准形,而 S 只不过是 F_m 的一种集合表示形式。因此有

$$F \text{ 不可满足} \Leftrightarrow S \text{ 不可满足}$$

下面开始用反证法证明

$$F \text{ 不可满足} \Leftrightarrow F_1 \text{ 不可满足}$$

(1) 先证明 $\Rightarrow$ 。

证明: 已知 F 不可满足,假设 F_1 是可满足的,则存在一个解释 I ,使 F_1 在解释 I 下为真。即对任意 $x_1, \cdots, x_{r-1}$ 在 I 的设定下有

$$(Q_{r+1} x_{r+1}) \cdots (Q_n x_n) M(x_1, \cdots, x_{r-1}, f(x_1, \cdots, x_{r-1}), x_{r+1}, \cdots, x_n)$$

为真。亦即对任意的 $x_1, \cdots, x_{r-1}$ 都有一个 $f(x_1, \cdots, x_{r-1})$,使

$$(Q_{r+1} x_{r+1}) \cdots (Q_n x_n) M(x_1, \cdots, x_{r-1}, f(x_1, \cdots, x_{r-1}), x_{r+1}, \cdots, x_n)$$

为真。即在 I 下有

$$(\forall x_1)(\forall x_2) \cdots (\forall x_{r-1})(\exists x_r) \cdots (Q_{r+1} x_{r+1}) \cdots (Q_n x_n) M(x_1, \cdots, x_{r-1}, x_r, x_{r+1}, \cdots, x_n)$$

为真。即 F 在 I 下为真。

但这与前提 F 是不可满足的相矛盾,即假设 F_1 为可满足是错误的。从而可以得出“若 F 不可满足,则必有 F_1 不可满足”。

(2) 再证明 $\Leftarrow$ 。

证明: 已知 F_1 不可满足,假设 F 是可满足的。于是便有某个解释 I 使 F 在 I 下为真。即对任意的 $x_1, \dots, x_{r-1}$, 在 I 的设定下都可找到一个 x_r , 使

$$(Q_{r+1}x_{r+1}) \cdots (Q_n x_n) M(x_1, \dots, x_{r-1}, x_r, x_{r+1}, \dots, x_n)$$

为真。若扩充 I , 使它包含一个函数 $f(x_1, \dots, x_{r-1})$, 且有

$$x_r = f(x_1, \dots, x_{r-1})$$

这样, 就可以把所有的 $(x_1, \dots, x_{r-1})$ 映射到 x_r , 从而得到一个新的解释 I' , 并且在此解释下对任意的 $x_1, \dots, x_{r-1}$ 都有

$$(Q_{r+1}x_{r+1}) \cdots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1}, \dots, x_n)$$

为真。即在 I' 下有

$$(\forall x_1)(\forall x_2) \cdots (\forall x_{r-1})(Q_{r+1}x_{r+1}) \cdots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1}, \dots, x_n)$$

为真。它说明 F 在解释 I' 下为真。但这与前提 F_1 是不可满足的相矛盾,即假设 F 为可满足是错误的。从而可以得出“若 F_1 不可满足,则必有 F 上不可满足”。

于是,定理得证。

由此定理可知。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即可。而有关如何证明一个子句集的不可满足性的问题就无须我们关心了,它可由鲁滨逊归结原理来解决。

3.3.2 鲁滨逊归结原理

鲁滨逊归结原理是在对子句集中的子句依次进行归结的基础上证明子句集的不可满足性的一种基础定理。由谓词公式转化为子句集的方法可以知道,在子句集中,子句之间是合取关系。其中,只要有一个子句是不可满足的,则整个子句集就是不可满足的。另外,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,则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来的,它的基本思想是:首先把欲证明问题的结论否定,并将其加入子句集,得到一个扩充的子句集 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;若不含有空子句,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。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。

1. 命题逻辑归结原理

归结原理的核心是求两个子句的归结式,因此需要先讨论归结式的定义和性质,然后再讨论命题逻辑的归结过程。下面先讨论归结式的定义与性质。

定义 3.17 若 P 是原子谓词公式,则称 P 与 $\neg P$ 为互补文字。

定义 3.18 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} 。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。

例 3.3 设 $C_1 = P \vee Q \vee R, C_2 = \neg P \vee S$,求 C_1 和 C_2 的归结式 C_{12} 。

解: 这里 $L_1 = P, L_2 = \neg P$,通过归结可以得到

$$C_{12} = Q \vee R \vee S$$

例 3.4 设 $C_1 = \neg Q, C_2 = Q$,求 C_1 和 C_2 的归结式 C_{12} 。

解: 这里 $L_1 = \neg Q, L_2 = Q$,通过归结可以得到

$$C_{12} = \text{NIL}$$

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: 设 $C_1 = L \vee C'_1, C_2 = \neg L \vee C'_2$ 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。对于解释 I 而言, L 和 $\neg L$ 中必有一个为假。

若 L 为假,则必有 C'_1 为真,否则就会使 C_1 为假,这将与前提假设 C_1 为真矛盾,因此只能有 C'_1 为真。

同理,若 $\neg L$ 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 也为真。即 C_{12} 是 C_1 和 C_2 的逻辑结论。

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 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 与 S_1 的不可满足性可以推出原子句集 S 的不可满足性。即

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

推论 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 与 S_2 的不可满足性是等价的。即

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

推论 1 和推论 2 的证明可利用不可满足性的定义和解释 I 的定义来完成,本书从略。这两个推论说明,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 ,或者用归结式代替它的亲本子句,然后对新的子句集证明其不可满足性就可以了。如果经归结能得到空子句,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,对不可满足的子句集 S ,其归结原理是完备的。这种不可满足性可用如下定理描述。

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。

要证明此定理,需要用到海伯伦原理,正是从这种意义上说,鲁滨逊归结原理是建立在海伯伦原理的基础上的。

这里需要指出,鲁滨逊归结原理对可满足的子句集 S 是得不出任何结果的。

2. 谓词逻辑归结原理

在谓词逻辑中,由于子句中含有变元,所以不能像命题逻辑的归结原理那样直接消去互补文字,在归结之前必须要用最一般合一对变元进行置换,比如有如下两个子句

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

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

在这种情况下,由于 $P(x)$ 与 $P(a)$ 的不一致,因此亲本子句 C_1 和 C_2 就无法进行归结,此时

就要用 C_1 与 C_2 的最一般合一 $\delta = \left\{ \frac{a}{x} \right\}$ 来对它们进行置换

$$C_1\delta = P(a) \vee Q(a)$$

$$C_2\delta = \neg P(a) \vee R(y)$$

置换后消去 $P(a)$ 和 $\neg P(a)$ 即可对以上二式进行归结并得出最终结果

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

在一般情形下,我们往往会遇到比较复杂的问题,所以要用一套合适的规则来描述,谓词逻辑中的归结可用如下定义来描述。

定义 3.19 设 C_1 和 C_2 是两个没有公共变元的子句, L_1 和 L_2 分别是 C_1 和 C_2 中的文字。

如果 L_1 和 $\neg L_2$ 存在最一般合一 δ , 则称

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

为 C_1 和 C_2 的二元归结式, 而 L_1 和 L_2 为归结式中的文字。

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

此外, 定义中还要求 C_1 和 C_2 无公共变元, 这也是合理的。例如 $C_1 = P(x)$, $C_2 = \neg P(f(x))$, 而 $S = \{C_1, C_2\}$ 是不可满足的。但由于 C_1 和 C_2 的变元相同, 就无法合一了。没有归结式, 就不能用归结法证明 S 的不可满足性, 这就限制了归结法的使用范围。但是, 如果对 C_1 或 C_2 的变元进行换名, 便可通过合一对 C_1 和 C_2 进行归结。如上例, 若先对 C_2 进行换名, 即 $C_2 = \neg P(f(y))$, 则可对 C_1 和 C_2 进行归结, 可得到一个空子句 NIL, 至此即可证明 S 是不可满足的。

事实上, 在由公式集化为子句集的过程中, 其最后一步就是进行换名处理。因此, 定义中假设 C_1 和 C_2 没有相同变元是可以的。下面看一些谓词逻辑归结的例子。

例 3.5 设 $C_1 = P(a) \vee R(x)$, $C_2 = \neg P(y) \vee Q(b)$, 求 C_{12} 。

解: 取 $L_1 = P(a)$, $L_2 = \neg P(y)$, 则 L_1 和 L_2 的最一般合一是 $\delta = \left\{ \frac{a}{y} \right\}$, 根据定义 3.19 可得

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

例 3.6 设 $C_1 = P(x) \vee Q(a)$, $C_2 = \neg P(x) \vee R(x)$, 求 C_{12} 。

解: 由于 C_1 和 C_2 有相同的变元 x , 不符合定义 3.19 的要求。所以为了进行归结, 需要修改 C_2 中变元的名字, 令 $C_2 = \neg P(b) \vee R(y)$ 。此时 $L_1 = P(x)$, $L_2 = \neg P(b)$, L_1 和 $\neg L_2$

的最一般合一是 $\delta = \left\{ \frac{b}{x} \right\}$, 则有

$$\begin{aligned} C_{12} &= (\{C_1\delta\} - \{L_1\delta\}) \cup (\{C_2\delta\} - \{L_2\delta\}) \\ &= (\{P(b), Q(a)\} - \{P(b)\}) \cup (\{\neg P(b), R(y)\} - \{\neg P(b)\}) \\ &= \{Q(a)\} \cup \{R(y)\} \\ &= \{Q(a), R(y)\} \\ &= \{Q(a), \vee R(y)\} \end{aligned}$$

例 3.7 设 $C_1 = P(x) \vee \neg Q(b)$, $C_2 = \neg P(a) \vee Q(y) \vee R(z)$, 求 C_{12} 。

解: 对 C_1 和 C_2 利用最一般合一, 可以得到两个互补对。但是需要注意, 求归结式不能同时消去两个互补对, 同时消去两个互补对的结果不是二元归结式, 如在

$$\delta = \left\{ \frac{a}{x}, \frac{b}{y} \right\}$$

中, 若同时消去两个互补对, 所得的 $R(z)$ 不是 C_1 和 C_2 的二元归结式。

例 3.8 设 $C_1 = P(x) \vee P(f(a)) \vee Q(x)$, $C_2 = \neg P(y) \vee R(b)$, 求 C_{12} 。

解: 对参与归结的某个子句, 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。本例的 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 $\delta = \left\{ \frac{f(a)}{x} \right\}$ 进行置换, 可得到

$$C_1\delta = \{P(f(a)) \vee Q(f(a))\}$$

此时可对 $C_1\delta$ 与 C_2 进行归结。选 $L_1 = P(f(a))$, $L_2 = \neg P(y)$, L_1 和 $\neg L_2$ 的最一般合一是 $\delta = \left\{ \frac{f(a)}{y} \right\}$, 则可得到 C_1 和 C_2 的二元归结式为

$$C_{12} = R(b) \vee Q(f(a))$$

在这个例子中, 把 $C_1\delta$ 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 δ , 则称 $C\delta$ 为子句 C 的因子。如果 $C\delta$ 是一个单文字, 则称它为 C 的单元因子。应用因子概念, 可对谓词逻辑中的归结原理给出如下定义。

定义 3.20 若 C_1 和 C_2 是无公共变元的子句, 则存在以下二元归结式:

- (1) C_1 和 C_2 的二元归结式。
- (2) C_1 和 C_2 的因子 $C_2\delta_2$ 的二元归结式。
- (3) C_1 的因子 $C_1\delta_1$ 和 C_2 的二元归结式。
- (4) C_1 的因子 $C_1\delta_1$ 和 C_2 的因子 $C_2\delta_2$ 的二元归结式。

这 4 种二元归结式都是子句 C_1 和 C_2 的二元归结式, 记为 C_{12} 。

对于谓词逻辑的归结原理来说, 归结式仍然为其亲本子句的逻辑结论。因此, 用归结式 C_{12} 来取代它在子句集 S 中的亲本子句, 所得到的新子句集仍然保持原子句集 S 的不可满足性。下面来看一个求二元归结式的例子。

例 3.9 设 $C_1 = P(y) \vee P(f(x)) \vee Q(g(x))$, $C_2 = \neg P(f(g(a))) \vee Q(b)$, 求 C_{12} 。

解: 对 C_1 来说, 取最一般合一 $\delta = \left\{ \frac{f(x)}{y} \right\}$, 得 C_1 的因子

$$C_1\delta_1 = P(f(x)) \vee Q(g(x))$$

将 C_1 的因子和 C_2 归结, 可得到 C_1 和 C_2 的二元归结式

$$C_{12} = Q(g(g(a))) \vee Q(b)$$

3.3.3 归结反演

有了鲁宾逊归结原理,我们就可以据此进行定理的证明。应用归结原理证明定理的过程称为归结反演。归结原理给出了证明子句集不可满足性的方法,即归结演绎的推理方法,其包含命题逻辑的归结反演和谓词逻辑的归结反演两种。下面分别进行介绍。

1. 命题逻辑的归结反演

归结原理给出了证明子句集不可满足性的方法。若假设 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。根据前面提到的反证法: 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。再根据之前的定理,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,就可用归结原理来进行定理的自动证明。

在命题逻辑中,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 ,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 。如此反复进行,若出现空子句,则停止归结,此时就证明了 G 为真。

例 3.10 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,即 $\neg R$ 为真,将 $\neg R$ 加入公式集,并将其化为子句集,步骤如下:

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

由于文字 R 与 $\neg R$ 、 P 与 $\neg P$ 、 Q 与 $\neg Q$ 、 T 与 $\neg T$ 均为互补文字,根据归结原理,我们对子句集中的这些互补对分别进行归结处理,并且每一个互补对归结后均把归结式并入子句中,迭代这一过程直到出现空子句 NIL 为止。此时,根据归结原理的完备性,可得子句集 S 是不可满足的,即开始时假设的 $\neg R$ 为真是错误的假设,故 R 为真已被证明。

2. 谓词逻辑的归结反演

谓词逻辑的归结反演与命题逻辑的归结反演的最大区别是两种方法中每个步骤的处理对象是不同的,但两种归结反演的主要思想都是统一的。谓词逻辑的归结反演是仅有一条推理规则的问题求解方法,在使用归结反演来证明 $P \rightarrow Q$ 成立时,我们实际上是证明其反面不成立,即 $\neg(P \rightarrow Q)$ 不可满足。在谓词逻辑中,由于子句集中的谓词一般都含有变元,因此不能像命题逻辑那样直接消去互补文字,而需要先用一个最一般合一对变元进行置换,然后才能进行归结。可见,谓词逻辑的归结反演要比命题逻辑的归结反演稍复杂些。下面来看一个谓词逻辑的归结反演的示例。

例 3.11 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。

证明: 先把 G 否定, 并放入 F 中, 得到的新子句集 $\{F, \neg G\}$ 为

$$\{(\forall x)((\exists y)A(x, y) \wedge B(y) \rightarrow (\exists y)(C(y) \wedge D(x, y))) \\ \neg(\neg(\exists x)C(x) \rightarrow (\forall x)(\forall y)(A(x, y) \rightarrow \neg B(y)))\}$$

再把 $\{F, \neg G\}$ 化成子句集, 得到

- (1) $\neg A(x, y) \vee \neg B(y) \vee C(f(x))$
- (2) $\neg A(u, v) \vee \neg B(v) \vee D(u, f(u))$
- (3) $\neg C(z)$
- (4) $A(m, n)$
- (5) $B(k)$

其中, (1)、(2) 是由 F 化出的两个子句, (3)、(4)、(5) 是由 $\neg G$ 化出的 3 个子句。

最后应用谓词逻辑的归结原理, 对上述子句集进行归结, 其过程如下:

$$(6) \neg A(x, y) \vee \neg B(y), \text{由(1)和(3)归结, 取 } \delta = \left\{ \frac{f(x)}{z} \right\}.$$

$$(7) \neg B(n), \text{由(4)和(6)归结, 取 } \delta = \left\{ \frac{m}{x}, \frac{n}{y} \right\}.$$

$$(8) \text{NIL}, \text{由(5)和(7)归结, 取 } \delta = \left\{ \frac{k}{x} \right\}.$$

至此, 出现空子句 NIL, 故 G 是 F 的逻辑结论得证。

3.3.4 归结策略

归结演绎推理实际上就是从子句集中不断寻找可进行归结的子句对, 并通过对这些子句对的归结, 最终得出一个空子句。由于我们实际并不知道哪些子句对可进行归结, 更不知道通过对哪些子句对的归结能尽快得到空子句, 因此就需要对子句集中的所有子句逐对进行比较, 直到得出空子句为止。假设有子句集 $S = \{C_1, C_2, C_3, C_4\}$, 则计算机中对此子句集的归结过程一般如下:

(1) 把 S 内任意子句两两进行归结, 得到一组归结式, 称为第一级归结式, 记为 S_1 。

(2) 把 S 与 S_1 内的任意子句两两进行归结, 得到一组归结式, 称为第二级归结式, 记为 S_2 。

(3) 把 S 和 S_1 内的子句与 S_2 内的任意子句两两进行归结, 得到一组归结式, 称为第三级归结式, 记为 S_3 。

(4) 如此反复, 直到出现空子句或者不能再继续归结为止。只要子句集是不可满足的, 则上述归结过程中一定会归结出空子句。

这种盲目地全面进行归结的方法, 不仅会产生许多无用的归结式, 更严重的是会产生组合爆炸问题。因此, 需要研究有效的归结策略来解决这些问题。

目前, 常用的归结策略可分为两大类: 一类是删除策略; 另一类是限制策略。删除策略是通过删除某些无用的子句来缩小归结范围; 限制策略是通过参与归结的子句进行某些限制, 来减少归结的盲目性, 以尽快得到空子句。下面介绍几种使用频率较高的归结策略。

1. 删除策略

删除策略有以下几种删除法。

1) 纯文字删除法

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。因而用包含纯文字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。所以可以把纯文字所在的子句从子句集中删去, 这样并不影响子句集的不可满足性。例如, 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) 重言式删除法

如果一个子句中同时包含互补文字对, 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。对于一个子句集来说, 不管是增加还是删去一个真值为真的子句都不会影响它的不可满足性。所以可从子句集中删去重言式。

3) 包孕删除法

设有子句 C_1 和 C_2 , 如果存在一个置换 δ 使得 $C_1\delta \in C_2$, 则称 C_1 包孕于 C_2 。

例如:

$$P(x) \text{ 包孕于 } P(y) \vee Q(z), \delta = \left\{ \frac{y}{x} \right\};$$

$$P(x) \text{ 包孕于 } P(a) \vee Q(z), \delta = \left\{ \frac{a}{x} \right\};$$

$$P(x) \vee Q(a) \text{ 包孕于 } P(f(a)) \vee Q(a) \vee R(y), \delta = \left\{ \frac{f(a)}{x} \right\}。$$

删去子句集中包孕的子句(即较长的子句), 不会影响子句集的不可满足性。所以可从子句集中删去包孕子句。

2. 限制策略

1) 支持集策略

支持集策略是一种限制策略。其限制的方法是: 每次归结时, 参与归结的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。支持集策略是完备的, 即假如对一个不可满足的子句集运用支持集策略进行归结, 那么最终会导出空子句。

例 3.12 用支持集策略归结子句集 $S = \{\neg I(x) \vee R(x), I(a), \neg R(y) \vee \neg L(y), L(a)\}$, 其中 $\neg I(x) \vee R(x)$ 是目标公式否定后得到的子句。

解: 用支持集策略进行归结的过程如下。

S : ① $\neg I(x) \vee R(x)$

② $I(a)$

③ $\neg R(y) \vee \neg L(y)$

④ $L(a)$

S_1 : ①与②归结得⑤ $R(a)$, 其中运用了最一般合一 $\left\{ \frac{a}{x} \right\}$

①与③归结得⑥ $\neg I(x) \vee \neg L(x)$, 其中运用了最一般合一 $\left\{ \frac{x}{y} \right\}$

①与④无法归结

S_2 : ①与⑤无法归结

①与⑥无法归结

②与⑤无法归结

②与⑥归结得⑦ $\neg L(a)$, 其中运用了最一般合一 $\left\{ \frac{a}{x} \right\}$

③与⑤归结得⑧ $\neg L(a)$, 其中运用了最一般合一 $\left\{ \frac{a}{y} \right\}$

③与⑥无法归结

④与⑤无法归结

④与⑥归结得⑨ $\neg I(a)$, 其中运用了最一般合一 $\left\{ \frac{a}{x} \right\}$

S_3 : ①与⑦无法归结

①与⑧无法归结

①与⑨无法归结

②与⑦无法归结

②与⑧无法归结

②与⑨归结得 NIL(结束)

至此, 归结结束。

上述支持集策略的归结过程可用归结树来表示, 如图 3-3 所示。

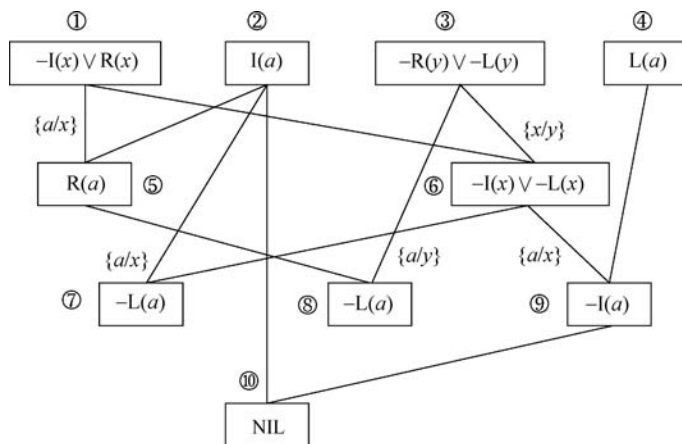


图 3-3 支持集策略的归结树

2) 线性输入策略

线性输入策略的限制方法是: 参与归结的两个子句中至少有一个是原始子句集中的子句(包括那些待证明公式的否定)。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。但是线性输入策略是不完备的。例如, 用线性输入策略对子句集 $S = \{P \vee Q, P \vee \neg Q, \neg P \vee Q, \neg P \vee \neg Q\}$ 进行归结, 就得不到空子句, 但是该子句集是不可满足的。

例 3.13 用线性输入策略对例 3.12 中的子句集进行归结。

解: 用线性输入策略进行归结的过程如下。

S: ① $\neg I(x) \vee R(x)$

② $I(a)$

③ $\neg R(y) \vee \neg L(y)$

④ $L(a)$

S_1 : ①与②归结得⑤ $R(a)$, 其中运用了最一般合一 $\left\{\frac{a}{x}\right\}$

①与③归结得⑥ $\neg I(x) \vee \neg L(x)$, 其中运用了最一般合一 $\left\{\frac{x}{y}\right\}$

①与④无法归结

②与③无法归结

②与④无法归结

③与④归结得⑦ $\neg R(a)$, 其中运用了最一般合一 $\left\{\frac{a}{y}\right\}$

S_2 : ①与⑤无法归结

①与⑥无法归结

①与⑦归结得⑧ $\neg I(a)$, 其中运用了最一般合一 $\left\{\frac{a}{x}\right\}$

②与⑤无法归结

②与⑥归结得⑨ $\neg L(a)$, 其中运用了最一般合一 $\left\{\frac{a}{x}\right\}$

②与⑦无法归结

③与⑤归结得 $\neg L(a)$, 其中运用了最一般合一 $\left\{\frac{a}{y}\right\}$

③与⑥无法归结

③与⑦无法归结

④与⑤无法归结

④与⑥归结得 $\neg I(a)$, 其中运用了最一般合一 $\left\{\frac{a}{x}\right\}$

④与⑦无法归结

S_3 : ①与⑧无法归结

①与⑨无法归结

①与⑩无法归结

①与⑪无法归结

②与⑧归结得 NIL(结束)

至此, 归结结束。

上述线性输入策略的归结过程可用归结树来表示, 如图 3-4 所示。

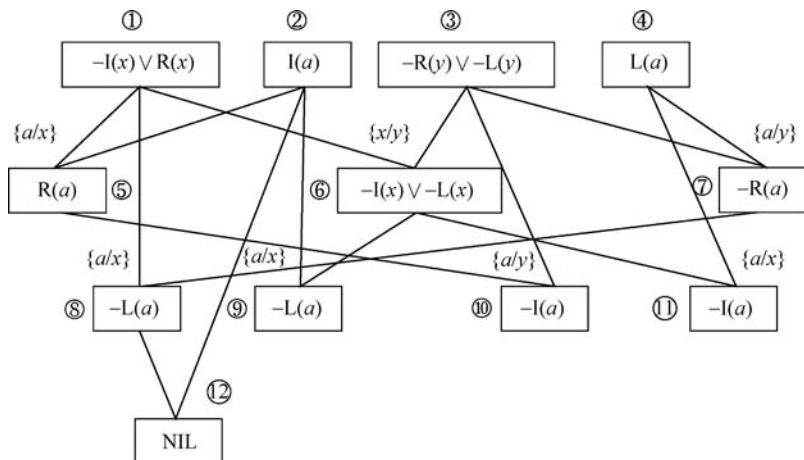


图 3-4 线性输入策略的归结树

3.4 非归结演绎推理

3.4.1 自然演绎推理

接下来我们讨论非归结演绎推理中的自然演绎推理。首先看它的定义：从一组已知为真的事实出发，直接运用经典逻辑的推理规则推出结论的过程称为自然演绎推理。自然演绎推理常被用在数理逻辑的证明中，其最主要的推理规则是由一大一小两个前提与一个结论组成的三段论法。其中，基本的推理是上一节提出的 P 规则、T 规则、假言推理和拒取式推理等。

假言推理的一般形式是

$$P, P \rightarrow Q \Rightarrow Q$$

表示：由 $P \rightarrow Q$ 为真及 P 为真，可推出 Q 为真。

例如，由“如果 x 是可燃物，那么 x 可以燃烧”和“塑料袋是可燃物”可以推出“塑料袋可以燃烧”的结论。

拒取式推理的一般形式是

$$P \rightarrow Q, \neg Q \Rightarrow \neg P$$

表示：由 $P \rightarrow Q$ 为真并且 Q 为假，则可推出 P 为假。

例如，由“如果 x 是可燃物，那么 x 可以燃烧”和“石头不可以燃烧”可以推出“石头不是可燃物”的结论。

说到这里，读者要特别注意避免否定前件 P 及肯定后件 Q 这两种类型的错误，例如如下推理：

- ① 如果打开 MP3，则能听到优美的音乐。
- ② 没有打开 MP3。
- ③ 所以，不能听到优美的音乐。

这就是应用了否定前件 P 的推理,显然是不正确的,因为其违背了确定性推理的逻辑规则。我们知道,通过手机、计算机、iPad 或前往音乐厅等诸多方式均能听到优美的音乐。故如此多的方法,凭什么得出“不打开 MP3 就听不到优美的音乐了”这种荒谬的结论呢?因此,当 $P \rightarrow Q$ 为真时,希望通过否定前件 P 为假来推出后件 Q 为假是不正确的,这是第一类错误。

又如如下推理:

- ① 如果打开 MP3,则能听到优美的音乐。
- ② 听到了优美的音乐。
- ③ 所以,打开了 MP3。

这就是应用了肯定后件 Q 的推理,同样是不正确的,其同样违背了确定性逻辑的逻辑规则。我们知道,通过手机、计算机、iPad 或前往音乐厅等诸多方式均能听到优美的音乐,那么凭什么断言我们听到的音乐就是由 MP3 发出的呢?因此,当 $P \rightarrow Q$ 为真时,希望通过肯定后件 Q 为真来推出前件 P 为真也是不正确的,这是第二类错误。

下面举例说明自然演绎推理方法。

例 3.14 假设已知如下事实:

- ① 只要是好玩的游戏小航都爱玩。
- ② “企鹅公司”出品的游戏都很有意思。
- ③ 《王者荣耀》是“企鹅公司”的一款游戏。

求证:小航爱玩游戏《王者荣耀》。

证明:

- ① 将谓词定义出来。

$F(x)$: x 是一款好玩的游戏。

$L(x, x)$: x 喜欢玩 y 游戏。

$T(x)$: x 是 T 公司的一款游戏。

- ② 将上述已知的事实与要证明的问题用谓词公式表示出来。

$\forall x (F(x) \rightarrow L(\text{Hang}, x))$: 只要是好玩的游戏,小航都喜欢玩。

$\forall x (T(x) \rightarrow F(x))$: “企鹅公司”出品的游戏都很有意思。

$T(w)$: 《王者荣耀》是“企鹅公司”出品的一款游戏。

$L(\text{Hang}, w)$: 求证小航爱玩游戏《王者荣耀》。

- ③ 根据上述自然演绎推理的相应规则进行推理。

因为

$$(\forall x)(F(x) \rightarrow L(\text{Hang}, x))$$

经全称固化,由一般到特殊有

$$F(m) \rightarrow L(\text{Hang}, m)$$

又因为

$$(\forall x)(T(x) \rightarrow F(x))$$

经全称固化,由一般到特殊有

$$(T(n) \rightarrow F(n))$$

依据 P 规则引入前提,再由假言推理得

$$T(w), T(n) \rightarrow F(n) \Rightarrow F(w)$$

由 T 规则将永真蕴含公式 $(\forall x)(T(x) \rightarrow F(x))$ 引入推理过程得

$$F(w), F(m) \rightarrow L(m) \Rightarrow F(\text{Hang}, w)$$

再次应用 P 规则及假言推理得

$$L(\text{Hang}, w)$$

综上所述,小航喜欢玩《王者荣耀》得证。

通常意义上讲,由已知事实推出的结论可能有两个甚至更多个,但只要欲解决的问题包含在这些结论之中,那么我们就可以认定推理成功,即欲证问题得以解决。

自然演绎推理是一种证明过程自然流畅、表达方法简明易懂的推理方法,它拥有非常清晰准确的推理规则,推理过程严密而不失灵活性,可以在推理规则中嵌入某领域的特定知识;但自然演绎推理也有其局限性,因其在推理过程中得到的中间结论过于繁多且增长速率高(呈指数级增长),所以容易产生组合爆炸,故不适用于复杂的问题。接下来我们讨论适用于复杂问题的推理方法。

3.4.2 与或型演绎推理

本节将讨论第二种经典的非归结演绎推理方法——与或型演绎推理。它与归结演绎推理最大的区别是:归结演绎推理要求把有关问题的知识及目标的否定都化成子句形式,然后通过归结进行演绎推理;而与或型演绎推理所遵循的推理规则只有一条,即归结规则。对于许多公式来说,子句集是一种不够高效的表达式。为提高公式推理和证明的效率,与或型演绎推理不再把有关知识转化为子句集,而是把领域知识和已知事实分别用蕴含式及与或型表示出来,然后通过蕴含式进行演绎推理,从而证明某个目标公式。

与或型演绎推理分为正向演绎、逆向演绎和双向演绎 3 种推理形式,下面对其分别进行讨论。

1. 与或型正向演绎推理

在与或型正向演绎推理中,以正向方式使用的规则称为 F 规则,其具有如下形式

$$L \rightarrow W$$

式中, L 为单文字, W 为与或型。

与或型正向演绎的推理方法是从已知事实出发,正向使用蕴含式(F 规则)进行演绎推理,直至得到某个目标公式的一个终止条件为止。在这种推理中,对已知事实、F 规则及目标公式的表示形式均有一定要求。如果不是所要求的形式,则需要进行变换。

1) 事实表达式的与或型变换及其树形表示

与或型正向演绎推理要求已知事实用不含蕴含符号“ $\rightarrow$ ”的与或型表示。把一个公式化为与或型的步骤与化为子句集的步骤类似,只是不必把公式化为子句的合取形式,也不能消去公式中的合取词。其具体过程如下:

- ① 利用 $P \rightarrow Q \Leftrightarrow \neg P \vee Q$ 消去公式中的蕴含连接词“ $\rightarrow$ ”。
- ② 利用德·摩根律及量词转化规律把否定词“ $\neg$ ”移到紧靠谓词的位置上。
- ③ 重新为变元命名,使不同量词约束的变元有不同的名字。
- ④ 引入斯克林函数消去存在量词。

⑤ 消去全称量词,且使各主要合取式中的变元不同名。

例如,对如下谓词公式

$$(\exists x) \forall y \{Q(y, x) \wedge \neg [R(y) \vee P(y) \wedge S(x, y)]\}$$

按上述步骤转化后得到

$$Q(z, a) \wedge \{[\neg R(y) \wedge \neg P(y)] \vee \neg S(a, y)\}$$

这个不包含蕴含连接词“ $\rightarrow$ ”的表达形式,称为与或型。

事实表达式的与或型可用一棵与或树表示,称为事实与或树。和其他树形结构相同,与或树最顶端的节点称作根节点,无分支的节点称为叶子节点。例如上式可用图 3-5 所示的与或树表示。

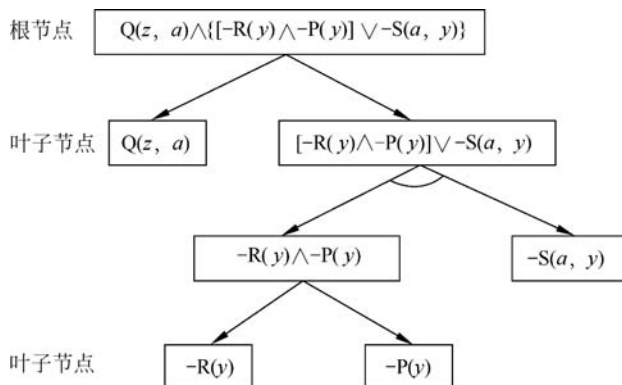


图 3-5 事实与或树

在图 3-5 中,根节点代表整个表达式,叶子节点表示不可再分解的原子公式,其他节点表示还可分解的子表达式。对于用析取符号“ $\vee$ ”连接而成的表达式,用一个 n 连接符,即图中的半圆弧把它们连接起来。对于用合取符号“ $\wedge$ ”连接而成的表达式,无须使用连接符。由与或树也可以很方便地获得原表达式的子句集。

2) F 规则的表示形式

我们知道,在与或型正向演绎推理中 F 规则的形式为 $L \rightarrow W$ 。式中我们之所以限制 F 规则的左部为单文字,是因为在进行演绎推理时,要用 F 规则作用于事实与或树。而该与或树的叶子节点都是单文字。这样就可用 F 规则的左部与叶子节点进行简单匹配(合一)了。

如果知识领域的表示形式不是所要求的形式,则需要通过变换将它变成规定的形式。变换步骤如下。

① 暂时消去蕴含词“ $\rightarrow$ ”。例如,对公式

$$(\forall x) \{[(\exists y)(\forall z)P(x, y, z)] \rightarrow (\forall u)Q(x, u)\}$$

运用等价关系可化为

$$(\forall x) \{ \neg [(\exists y)(\forall z)P(x, y, z)] \vee (\forall u)Q(x, u) \}$$

② 把否定词“ $\neg$ ”移到紧靠谓词的位置上。运用德·摩根律及量词转化规律将否定词“ $\neg$ ”移到括号中。于是上式可化为

$$(\forall x) \{ (\forall y)(\exists z)[\neg P(x, y, z)] \vee (\forall u)Q(x, u) \}$$

③ 引入斯克林函数消去存在量词。消去存在量词之后上式可化为

$$(\forall x)\{(\forall y)[\neg P(x, y, f(x, y))] \vee (\forall u)Q(x, u)\}$$

④ 消去全称量词。将上式消去全称量词后则化为

$$\neg P(x, y, f(x, y)) \vee Q(x, u)$$

此时公式中的变元都被视为受全称量词约束的变元。

⑤ 恢复为蕴含式。例如,用等价关系将上式变为

$$P(x, y, f(x, y)) \rightarrow Q(x, u)$$

3) 目标公式的表示形式

在与或型正向演绎推理中,要求目标公式用子句表示。如果目标公式不是子句形式,就需要化成子句形式,转化方法如 3.4.1 节所述。

2. 与或型逆向演绎推理

与或型逆向演绎推理是从待证明的问题(目标)出发,通过逆向使用蕴含式(B 规则)进行演绎推理,直到得到包含已知事实的终止条件为止。

与或型逆向演绎推理对目标公式、B 规则及已知事实的表示形式也有一定的要求。若不符合要求,则需进行转换。

1) 目标公式的与或型变换及与或树表示

在与或型逆向演绎推理中,要求目标公式用与或型表示。其变换过程和与或型逆向演绎推理中对已知事实的变换基本相似。但是要用存在量词约束的变元的斯克林函数替换由全称量词约束的相应变元,并且先消去全称量词,再消去存在量词。这是与或型逆向演绎推理与正向演绎推理进行变换的不同之处。例如,对如下目标公式

$$(\exists y)(\forall x)\{P(x) \rightarrow [Q(x, y) \wedge \neg R(x) \wedge S(y)]\}$$

经过与或型逆向演绎方法转化后可得到

$$\neg P(f(z)) \vee \{Q(f(y), y) \wedge [\neg R(f(y)) \vee \neg S(y)]\}$$

在变换时应使各个主要的析取式具有不同的变元名。

目标公式的与或型也可用与或树表示,但其表示方式和与或型正向演绎中的事实与或树的表示略有不同。目标公式与或树中的 $\wedge$ 连接符用来把具有合取关系的子表达式连接起来,而在与或型正向演绎中的连接符则是把已知事实中具有析取关系的子表达式连接起来。上述目标公式的与或树如图 3-6 所示。

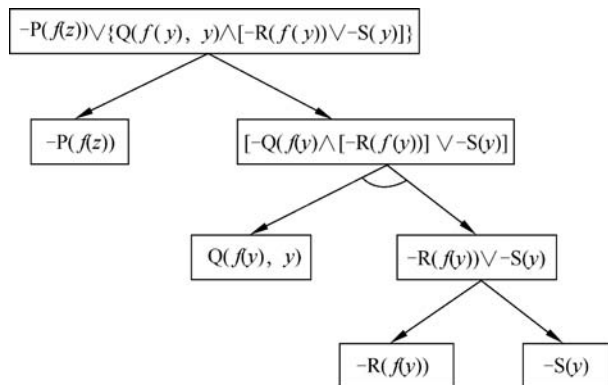


图 3-6 目标公式的与或树

2) B 规则的表示形式

B 规则的表示形式为

$$W \rightarrow L$$

其中, W 为与或型, L 为单文字。

之所以限制规则的右部为文字, 是因为推理时要用它与目标与或树中的叶子节点进行匹配, 而目标与或树中的叶子节点是文字。如果已知的 B 规则不是所要求的形式, 则可用与转换 F 规则类似的方法将其转化成规定的形式。特别是对于

$$W \rightarrow (L_1 \wedge L_2)$$

这样的蕴含式均可化为两个 B 规则

$$W \rightarrow L_1, W \rightarrow L_2$$

3) 已知事实的表示形式

在与或型逆向演绎推理中, 要求已知事实是文字的合取式, 即形如

$$F_1 \wedge F_2 \wedge \cdots \wedge F_n$$

在问题求解中, 由于每个 $F_i (i=1, 2, \cdots, n)$ 都可单独起作用, 因此可把上式表示为事实的集合

$$\{F_1, F_2, \cdots, F_n\}$$

4) 推理过程

应用 B 规则进行与或型逆向演绎推理的目的在于求解问题。当从目标公式的与或树出发, 通过运用 B 规则最终得到了某个终止在事实节点上的一致解图时, 推理就成功结束。一致解图是指在推理过程中所用到的置换应该是一致的。与或型逆向演绎推理的过程如下:

(1) 用与或树将目标公式表示出来。

(2) 将 B 规则的右部和与或树的叶子节点进行匹配, 并将匹配成功的 B 规则加入与或树中。

(3) 重复步骤(2), 直到产生某个终止在事实节点上的一致解图为止。

上述推理过程如图 3-7 所示。

3. 与或型双向演绎推理

与或型正向演绎推理要求目标公式是文字的析取式, 与或型逆向演绎推理要求事实公式为文字的合取式。这两点使得与或型正向演绎和逆向演绎推理都有一定的局限性。为了克服这种局限性, 充分发挥各自优势, 可使用双向演绎推理。

正向和逆向组合系统是建立在两个系统相结合的基础上的。此组合系统的总数据库由表示目标和表示事实的两个与或图组成。这些与或图最初用来表示给出的事实和目标的某些表达式集合, 现在这些表达式的形式都不受约束。这些与或图分别用正向系统的 F 规则和逆向系统的 B 规则来修正。设计者必须决定哪些规则用来处理事实图以及哪些规则用来处理目标图。尽管新系统在修正由两部分构成的数据库时只沿一个方向进行, 但我们仍然把这些规则分别称为 F 规则和 B 规则, 继续限制 F 规则为单文字前项和 B 规则为单文字后项。

组合演绎系统的主要复杂之处在于其终止条件。终止涉及两个图之间的适当交接。

在完成两个图之间的所有可能匹配之后, 目标图中根节点上的表达式是否已经根据事实图中根节点上的表达式和规则得到证明仍然需要判定。只有得到这样的证明时, 推理才

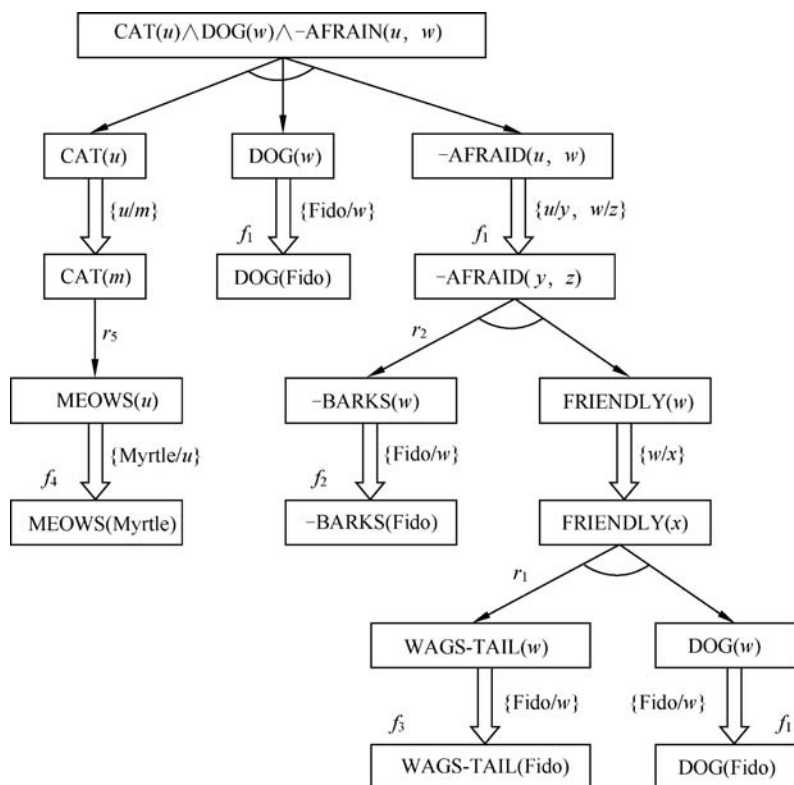


图 3-7 与或型逆向演绎的推理过程

算成功终止。当然,若能断定在给定方法限度内找不到证明时,推理则以失败告终。也就是说,分别从正、反两个方向进行推理,其与或树分别向着对方扩展。只有当它们对应的叶子节点都可以合一时,推理才能结束。在推理过程中用到的所有置换必须是一致的。

定义 3.21 设置换集合

$$\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$$

中的第 i 个置换 θ_i ($i=1, 2, \dots, n$) 为

$$\theta_i = \left\{ \frac{t_{i1}}{x_{i1}}, \frac{t_{i2}}{x_{i2}}, \dots, \frac{t_{im(i)}}{x_{im(i)}} \right\}$$

其中, t_{ij} ($j=1, 2, \dots, m(i)$) 为项, x_{ij} ($j=1, 2, \dots, m(i)$) 为变元, 则置换集合是一致的充要条件是如下两个元组可合一:

$$T = \{(t_{11}, t_{12}, \dots, t_{1m(1)}), (t_{21}, t_{22}, t_{2m(2)}), \dots, (t_{nm(n)})\}$$

$$X = \{(x_{11}, x_{12}, \dots, x_{1m(1)}), (x_{21}, x_{22}, x_{2m(2)}), \dots, (x_{nm(n)})\}$$

例如:

- (1) 设 $\theta_1 = \left\{ \frac{x}{y} \right\}$, $\theta_2 = \left\{ \frac{y}{z} \right\}$, 则 $\theta = \{\theta_1, \theta_2\}$ 是一致的。
- (2) 设 $\theta_1 = \left\{ \frac{f(g(x_1))}{x_3}, \frac{f(x_2)}{x_4} \right\}$, $\theta_2 = \left\{ \frac{x_4}{x_3}, \frac{g(x_1)}{x_2} \right\}$, 则 $\theta = \{\theta_1, \theta_2\}$ 是一致的。
- (3) 设 $\theta_1 = \left\{ \frac{a}{x} \right\}$, $\theta_2 = \left\{ \frac{b}{x} \right\}$, 则 $\theta = \{\theta_1, \theta_2\}$ 是不一致的。

(4) 设 $\theta_1 = \left\{ \frac{g(y)}{x} \right\}$, $\theta_2 = \left\{ \frac{f(x)}{y} \right\}$, 则 $\theta = \{\theta_1, \theta_2\}$ 是不一致的。

与或型演绎推理不必把公式化为子句集,保留了蕴含连接词“ $\rightarrow$ ”,这样就可以直观地表达出因果关系,比较自然。但是,与或型正向演绎推理把目标表达式限制为文字的析取式,与或型逆向演绎推理把已知事实表达式限制为文字的合取式。与或型双向演绎推理虽然可以克服以上限制,但是其终止时机与判断却难于被掌握。

3.5 案例：家庭财务分配管理系统

为了更好地说明如何用谓词公式描述问题以及如何针对问题领域进行逻辑推理,本节设计一个简单的家庭财务分配管理系统,将其作为例子来表明经典逻辑推理在若干实际问题中的应用。

本系统帮助用户判定是将资金存在银行还是去买股票,或者既存银行也买股票。家庭资金的使用方式取决于家庭收入和当前银行存款总额,可以按下述标准判断:

① 不论家庭收入如何,银行存款不足的家庭应优先考虑增加存款额。

② 有足够银行存款和足够收入的家庭应考虑买股票,这种情况下可以承担买股票的风险,使投资利润更大化。

③ 已有足够银行存款的低收入家庭可考虑将收入的余额用于银行存款和买股票,这样既能增加必要时便于使用的积蓄,也能通过买股票增加家庭收入。

存款额和家庭收入是否足够,由家庭必须赡养的人数确定。这里要求人均银行存款至少为 5000 元。足够的收入必须是稳定收入,且每年至少应有 50000 元再加上各赡养人的年均消费 4000 元。

我们使用一元谓词符号 S 和 I 分别描述家庭银行存款和收入是否充足,它们的变元都为 yes 或 no。从而, S(yes)、S(no)、I(yes) 和 I(no) 分别描述有充足、不足的银行存款,以及有充足、不足的家庭收入。咨询系统的结论用一元谓词 INVESTMENT 描述,变元可为 stocks、savings 或 combination。故 INVESTMENT(stocks)、INVESTMENT(savings) 和 INVESTMENT(combination) 分别描述资金用于买股票、存入银行及既买股票也存银行。

使用这些谓词,可用蕴含式表示资金使用的不同策略。上面给出的标准可表示如下:

① $S(\text{no}) \rightarrow \text{INVESTMENT}(\text{savings})$

② $S(\text{yes}) \wedge I(\text{yes}) \rightarrow \text{INVESTMENT}(\text{stocks})$

③ $S(\text{yes}) \wedge I(\text{no}) \rightarrow \text{INVESTMENT}(\text{combination})$

其次,系统必须判定银行存款和收入怎样才算足够或不够。需定义一元函数 mS 来确定最小足够存款额,其变元为家庭赡养人数,且 $mS(y) = 5000 \times y$ 。用该函数,存款是否足够可由下述句子确定:

④ $(\text{SUM}(x) \wedge D(y) \wedge \text{GREATER}(x, mS(y))) \rightarrow S(\text{yes})$

⑤ $(\text{SUM}(x) \wedge D(y) \wedge \neg \text{GREATER}(x, mS(y))) \rightarrow S(\text{no})$

其中, SUM(x) 和 D(y) 分别表示当前家庭银行存款总额为 x 及家庭赡养人数为 y, 句子中出现的变量皆为全称量化变量。同样,定义函数 minI 为 $\text{minI}(x) = 50000 + 4000 \times x$, 该函数用于计算当家庭赡养人数为 x 时,家庭最小足够收入。下述句子判定家庭收入是否足够:

$$\textcircled{6} (E(x, \text{steady}) \wedge D(y) \wedge \text{GREATER}(x, \text{minI}(y))) \rightarrow I(\text{yes})$$

$$\textcircled{7} (E(x, \text{steady}) \wedge D(y) \wedge \neg \text{GREATER}(x, \text{minI}(y))) \rightarrow I(\text{no})$$

$$\textcircled{8} E(x, \text{unsteady}) \rightarrow I(\text{no})$$

其中,二元谓词 $E(x, y)$ 表示家庭当前总收入为 x , y 表示 x 是否为稳定收入, y 只能取 steady 或 unsteady 。句子中出现的变量皆为全称量化变量。

用户咨询该系统时需用谓词 SUM、E 和 D 描述他的家庭情况。

设用户家庭当前银行存款总额为 22000 元,有 25000 元的稳定收入,赡养人数为 3,可描述如下:

$$\textcircled{9} \text{SUM}(22000)$$

$$\textcircled{10} E(25000, \text{steady})$$

$$\textcircled{11} D(3)$$

上述①~⑪构成的句子集合 S 描述了问题领域。使用一致化算法和假言推理可推断出该用户正确使用资金的策略,策略是 S 的逻辑推论。

首先考虑蕴含式⑦的前件中第一成分 $E(x, \text{steady})$ 与⑩,它们是可一致的,其最一般合一为 $\left\{\frac{25000}{x}\right\}$ 。⑦的前件中第二成分 $D(y)$ 与⑪ 是可一致的,其最一般合一为 $\left\{\frac{3}{y}\right\}$ 。

将这两个最一般合一合成为 $\left\{\frac{25000}{x}, \frac{3}{y}\right\}$ 作用到⑦上得出 $I(\text{no})$,即

$$(E(25000, \text{steady}) \wedge D(3) \wedge \neg \text{GREATER}(25000, \text{minI}(3))) \rightarrow I(\text{no})$$

计算出函数 $\text{minI}(3)$ 后得出

$$(E(25000, \text{steady}) \wedge D(3) \wedge \neg \text{GREATER}(25000, 27000)) \rightarrow I(\text{no})$$

前件中 $\neg \text{GREATER}(25000, 27000)$ 为真,且 $E(25000, \text{steady}) \wedge D(3)$ 与⑩及⑪的合取相匹配,按假言推理可推断出 $I(\text{no})$,将其加入 S 可得出。

$$\textcircled{12} I(\text{no})$$

同样,⑨及⑪的合取 $\text{SUM}(22000) \wedge D(3)$ 与④的前件中前两成分可一致化,将得到的两个最一般合一 $\left\{\frac{22000}{x}\right\}$ 与 $\left\{\frac{3}{y}\right\}$ 进行合成可得 $\left\{\frac{22000}{x}, \frac{3}{y}\right\}$ 。

$$(\text{SUM}(22000) \wedge D(3) \wedge \text{GREATER}(22000, \text{mS}(3))) \rightarrow S(\text{yes})$$

计算出 $\text{mS}(3)$ 的值后,前件中 $\text{GREATER}(22000, 50000)$ 为真,其余部分与⑨和⑪的合取相匹配,按假言推理推断出 $S(\text{yes})$,将其加入 S 可得:

$$\textcircled{13} S(\text{yes})$$

蕴含式③的前件与⑬和⑫的合取完全匹配,再由假言推理得出结论

$$\textcircled{14} \text{INVESTMENT}(\text{combination})$$

这便是系统提供给用户的资金使用建议,将家庭资金进行储蓄和购买股票。

使用 Python 实现的家庭财务分配管理系统大致如代码清单 3-1 所示。

代码清单 3-1 家庭财务分配管理系统

```
def rule01():
    global S, I, INVESTMENT, deposit, people, income, status
    if S == 'no':
```

```
INVESTMENT = 'savings'

def rule02():
    global S, I, INVESTMENT, deposit, people, income, status
    if S == 'yes' and I == 'yes':
        INVESTMENT = 'stocks'

def rule03():
    global S, I, INVESTMENT, deposit, people, income, status
    if S == 'yes' and I == 'no':
        INVESTMENT = 'combination'

def rule04():
    global S, I, INVESTMENT, deposit, people, income, status
    x = deposit
    y = people
    if x is None or y is None:
        return
    if SUM(x) and D(y) and GREATER(x, mS(y)):
        S = 'yes'

def rule05():
    global S, I, INVESTMENT, deposit, people, income, status
    x = deposit
    y = people
    if x is None or y is None:
        return
    if SUM(x) and D(y) and not GREATER(x, mS(y)):
        S = 'no'

def rule06():
    global S, I, INVESTMENT, deposit, people, income, status
    x = income
    y = people
    if x is None or y is None:
        return
    if status == 'steady' and D(y) and GREATER(x, minI(y)):
        I = 'yes'

def rule07():
    global S, I, INVESTMENT, deposit, people, income, status
    x = income
    y = people
    if x is None or y is None:
        return
    if status == 'steady' and D(y) and not GREATER(x, minI(y)):
        I = 'no'

def rule08():
    global S, I, INVESTMENT, deposit, people, income, status
    if status == 'unsteady':
        I = 'no'
```

习题

一、选择题

1. 设 $C_1 = P \vee Q \vee R$, $C_2 = \neg P \vee S$, 则 C_1 与 C_2 的归结式 C_{12} 为()。
 - A. $Q \vee R \vee S$
 - B. $Q \wedge R \wedge S$
 - C. $Q \vee R \wedge S$
 - D. $Q \wedge R \vee S$
2. 如果命题 P 为真、命题 Q 为假, 则下述复合命题() 为真命题。
 - A. P 且 Q
 - B. 如果 Q 则 P
 - C. 非 P
 - D. 如果 P 则 Q
3. 下面逻辑等价关系不成立的是()。
 - A. $\forall x \neg P(x) \equiv \neg \exists x P(x)$
 - B. $\exists x P(x) \equiv \neg \forall x P(x)$
 - C. $\forall x P(x) \equiv \neg \exists x \neg P(x)$
 - D. $\neg \forall x P(x) \equiv \exists x \neg P(x)$
4. 下面() 对命题逻辑中的归结(resolution)规则的描述是不正确的。
 - A. 对命题 Q 及其反命题应用归结法, 所得到的命题为假命题
 - B. 对命题 Q 及其反命题应用归结法, 所得到的命题为空命题
 - C. 在两个析取复合命题中, 如果命题 Q 及其反命题分别出现在这两个析取复合命题中, 则通过归结法可得到一个新的析取复合命题, 只是在析取复合命题中要去除命题 Q 及其反命题。
 - D. 如果命题 Q 出现在一个析取复合命题中, 命题 Q 的反命题单独存在, 则通过归结法可得到一个新的析取复合命题, 只是在析取复合命题中要去除命题 Q 及其反命题
5. 下面() 命题范式的描述是不正确的。
 - A. 一个合取范式是成立的, 当且仅当它的每个简单析取式都是成立的
 - B. 有限个简单析取式构成的合取式称为合取范式
 - C. 一个析取范式是不成立的, 当且仅当它包含一个不成立的简单合取式
 - D. 有限个简单合取式构成的析取式称为析取范式
6. 设 $P(x): x$ 是鸟, $Q(x): x$ 会飞, 命题“有的鸟不会飞”可符号化为()。
 - A. $\neg(\forall x)(P(x) \rightarrow Q(x))$
 - B. $\neg(\forall x)(P(x) \wedge Q(x))$
 - C. $\neg(\exists x)(P(x) \rightarrow Q(x))$
 - D. $\neg(\exists x)(P(x) \wedge Q(x))$
7. $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ 是()。
 - A. 德·摩根律
 - B. 吸收律
 - C. 补余律
 - D. 结合律
8. $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$ 是()。
 - A. 结合律
 - B. 连接词化规律
 - C. 分配律
 - D. 德·摩根律
9. 下列命题公式不是永真式的是()。
 - A. $(P \rightarrow Q) \rightarrow P$
 - B. $P \rightarrow (Q \rightarrow P)$
 - C. $\neg P \vee (Q \rightarrow P)$
 - D. $(P \rightarrow Q) \vee P$

10. 下列谓词公式中是前束范式的是()。

A. $\forall x F(x) \wedge \neg(\exists x) G(y)$

B. $\forall x P(x) \wedge \forall y G(y)$

C. $\forall x (P(x) \rightarrow \exists y Q(x, y))$

D. $\forall x \exists y (P(x) \rightarrow Q(x, y))$

二、判断题

1. 演绎推理的核心是三段论,常用三段论由一个大前提、一个小前提和一个结论 3 部分组成。()

2. 归纳推理是从一类事物的大量特殊事例出发,去推出该类事务的一般性结论。()

3. 演绎推理是在已知领域内的一般性知识前提下,通过演绎求解一个具体问题或证明一个给定的结论。()

4. 在归纳推理中,所推理出的结论包含在前提内容中。()

5. 混沌推理是将正向推理与反向推理结合起来的一种推理。()

6. 应用归结原理证明定理的过程称为归结反演。()

7. 谓词逻辑的归结反演与命题逻辑的归结反演最大区别是两种方法中每个步骤的处理对象是不同的,所以两种归结反演的主要思想也是不同的。()

8. 归结演绎推理世界上就是从字句集中不断寻找可进行归结的子句对,并通过对这些子句对的归结,最终得到一个空子句。()

9. 自然演绎推理是指从一组已知为真的事实出发,直接运用经典逻辑的推理规则推出结论的过程称为自然演绎推理。()

10. 与或型正向演绎推理要求目标公式是文字的合取式,与或型逆向演绎推理要求事实公式是文字的析取式。()

三、问答题

1. 什么是推理?请从多种角度阐述推理的分类。

2. 什么是逆向推理?它的基本过程是什么?

3. 什么是置换?什么是合一?

4. 请阐述鲁宾逊归结原理的基本思想。

5. 将下列谓词公式化成对应的子句集:

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

6. 设已知:

(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。

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

7. 设已知:

(1) 能阅读的人是识字的;

(2) 小狗不识字;

(3) 有些小狗是很聪明的。

请用归结演绎推理证明:有些很聪明的人并不识字。