**学习目标：了解专家系统的定义，学习专家系统的组成，理解确定性推理和非确定性推理。**

**第四章 专家系统**

**导读**

在人工智能的发展过程中，研究者们逐渐认识到知识的重要性。如果计算机可以学习到人类在某个领域的专业知识，并且利用这些知识解决这个领域的专业问题，不是一种很好的人工智能的体现吗？所以，将一个专家在他本领域的专业知识总结出来，以计算机可以使用的形式表达，让计算机系统可以利用这种知识解决问题，这就是专家系统研究的动机。

世界上第一个专家系统是DENDRAL，诞生于1965年，由E.A.费根鲍姆等人在总结通用问题求解系统的成功与失败经验的基础上，结合化学领域的专门知识研制出来的，可以推断化学分子结构。后来，费根鲍姆领导的小组又研发了著名的专家系统MYCIN，它的功能是帮助医生对住院的血液感染患者进行诊断和选用抗生素HYPERLINK类药物进行治疗。这是个非常重要的专家系统，因为可以说它确定了专家系统的基本结构，为后来的专家系统研究奠定了基础。

**4.1 系统概述**

我们可以将专家系统定义为：一种智能的计算机程序，其内部含有大量的某个领域专家水平的知识与经验，它运用知识和推理来解决该领域内只有专家才能解决的复杂问题。需要注意的是，这里的知识和问题均属于同一个特定领域。简而言之，专家系统是一种模拟人类专家解决领域问题的计算机程序系统。

一个专家系统的基本结构如图4.1所示。主要有六个部分构成：人机交互界面、知识库、推理机、解释器、动态数据库、知识获取。人机界面是系统与用户进行交流时的界面。通过该界面，用户输入基本信息、回答系统提出的相关问题，并输出推理结果及相关的解释等；知识库用来存放专家提供的领域知识和事实等。专家系统的问题求解过程是通过知识库中的知识来模拟专家的思维方式的；推理机针对当前问题的条件或已知信息，反复匹配知识库中的规则，获得新的结论，以得到问题求解结果。推理方式可以有正向和反向推理两种。将在下一节中进行更详细的解释；解释器能够根据用户的提问，对结论、求解过程做出说明，因而使专家系统更像一个“人”；动态数据库专门用于存储推理过程中所需的原始数据、中间结果和最终结论，往往是作为暂时的存储区；知识获取是专家系统知识库是否优越的关键，也是专家系统设计的“瓶颈”问题，通过知识获取，可以扩充和修改知识库中的内容，也可以实现自动学习功能。

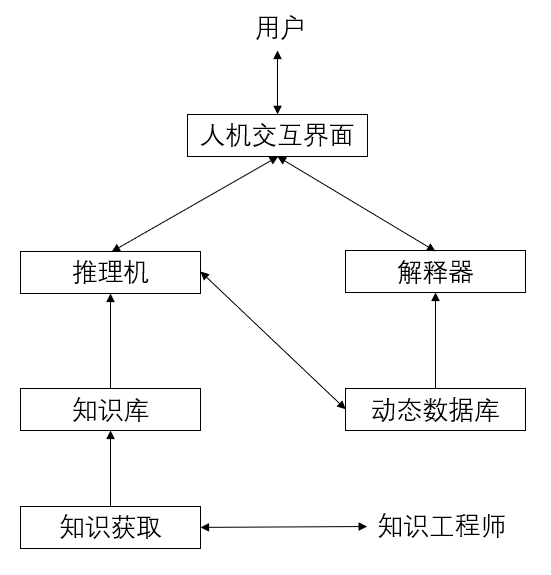


图4.1 专家系统基本结构

专家系统中最重要的部分是知识库和推理机。这也是专家系统不同于一般的计算机程序系统的所在。以这两个部分为核心，专家系统可以处理非确定性的问题，它的目标不是追求问题的最佳解，而是利用知识和推理得到一个满意解。一个好的专家系统应该强调知识库与其他子系统的分离，因为知识库是存储专业知识的，这些知识是与领域强相关的，不同领域的知识库肯定不同，然而推理机等其他的子系统在不同的领域也可能具有一定的通用性。

我们一般将存放于知识库中的知识称为规则或者知识规则，一般以如下形式表示：

IF <前提> THEN <结论>

表示：当 <前提> 被满足时，可以得到 <结论>

例如：IF 阴天and湿度大 THEN 下雨

表示：如果阴天且湿度大，则会下雨。

规则的<结论>可以是类似上例中的“下雨”这样的结果，也可能是一个“动作”，例如：

IF 天黑 THEN 打开灯

也可能是其他类型，比如删除某个数据等。

推理机是一个执行结构，它负责对知识库中的知识进行解释，利用知识库进行推理。假设知识以规则的形式表示，推理机会根据某种策略对知识库中的规则进行预测，选择一个<前提>可以满足的规则，得到该规则的<结论>，并根据<结论>的不同类型执行不同操作。

**4.2 推理方法**

专家系统中的推理机是如何利用知识库进行推理的？这个答案会根据知识表示方法的不同而有所不同。在专家系统中，规则是最常用的知识表示方法，下面以规则为例进行说明。

按照推理的方向，推理方法可以分为正向推理和逆向推理。正向推理就是正向地使用规则，从已知条件触发向目标进行推理。其基本思想是：检验是否有规则的前提被动态数据库中的已知事实满足，如果被满足，则将该规则的结论放入动态数据库中，再检查其他的规则是否有前提被满足；反复该过程，直到目标被某个规则推出结束，或者再也没有新结论被推出为止。由于这种推理方法是从规则的前提向结论进行推理，所以称为正向推理。由于正向推理是通过动态数据库中的数据来“触发“规则进行推理的，所以又称为数据驱动的推理。

例9.1 设有规则：

R1：IF A and B THEN C

R2：IF C and D THEN E

R3：IF E THEN F

并且已知A、B、D成立，求证F成立

初始时已知A、B、D在动态数据库中，根据规则R1，推出C成立，所以将C加入到动态数据库中；根据规则R2，推出E成立，将E加入到动态数据库中；根据R3，推出F成立，将F加入到动态数据库中。由于F是求证的目标，结果成立，推理结束。

如果在推理过程中，有多个规则的前提同时成立，如何选择一条规则进行推理是冲突消解问题。最简单的办法是按照规则的自然顺序，选择第一个前提条件满足的推理优先执行。也可以对多个规则进行评估，哪条规则前提被满足的条件多，哪条规则就优先执行；或者从规则的结论距离要推导的结论的远近来考虑。

逆向推理又被称为反向推理，是逆向地使用规则，先将目标作为假设，反推是否有某条规则支持该假设，即规则的结论与假设是否一致，然后看结论与假设相关的规则其前提是否成立。如前提成立（在动态数据库中进行匹配），则假设得到求证，结论被放入动态数据库中；否则将该规则的前提加入到假设集中，一个一个的求证这些假设，直到目标假设被验证为止。由于逆向推理是从假设求解目标成立，是逆向使用规则进行推理的，所以又被称为目标驱动的推理。

例9.2 在例9.1中，如何使用逆向推理推导出F成立？

首先将F作为假设，发现规则R3的结论可以推导出F，然后检验R3的前提是否成立。目前动态数据库中还没有记录E是否成立，由于规则R2的结论可以得到E，依次检验R2的前提C和D是否成立。首先检验C，由于C也没有再动态数据库中，再次找结论含有C的规则，找到规则R1，发现其前提A、B均成立（在动态数据库中），从而推导出C成立，将C放入动态数据库中。再检验规则R2的另一个前提条件D，由于D在动态数据库中，所以D成立，从而R2的前提全部被满足，推出E成立，并将E放入动态数据库总。由于E已经被推出成立，所以规则R3的前提成立了，从而最终推出目标F成立。

在逆向推理中也存在冲突消解问题，可采用与正向推理一样的方法解决。

一般的结论及推理都是确定性的，也就是说前提成立，结论一定成立。比如在几何定理证明中，如果两个同位角相等，则两条直线一定是平行的。但是在很多实际问题中，推理往往具有模糊性、不确定性。比如“如果阴天则可能下雨“，但我们都知道阴天了不一定就会下雨，这就属于非确定性推理问题。关于非确定性推理问题，我们将会在之后详细介绍。

**确定性推理**

假如你是一位动物学家，可以识别各种动物。你的朋友FD周末带小孩去动物园游玩并见到了一个动物，FD不知道该动物是什么，于是给你打电话咨询，你们之间有了以下的对话：

你：你看到的动物有羽毛吗？

FD：有羽毛

你：会飞吗？

FD：（经观察后）不会飞。

你：有长腿吗？

FD：没有

你：会游泳吗？

FD：（看到该动物在水中）会。

你：颜色是黑白吗？

FD：是。

你：这个动物是企鹅。

在以上对话中，当得知动物有羽毛后，你就知道了该动物属于鸟类，于是你提问是否会飞；当得知不会飞后，你开始假定这可能是鸵鸟，于是提问是否有长腿；在得到否定回答后，你马上想到了可能是企鹅，于是询问是否会游泳；然后为了进一步确认是否是企鹅，又问颜色是否是黑白的；得知是黑白颜色后，马上就确认该动物是企鹅。



图4.2 会游泳的短腿黑白色羽毛的鸟——企鹅

我们也希望一个动物识别专家系统能像你一样完成以上过程，通过与用户的交互回答用户有关动物的问题。

为了实现这样的专家系统，首先要把你有关识别动物的知识总结出来，并以计算机可以使用的方式存放在计算机中。可以用规则表示这些知识，为此，我们设计一些谓词以方便地表达知识。首先是same，表示动物具有某种属性，如可以用（same 有羽毛 yes）表示是否具有羽毛，当动物有羽毛时为真，否则为假。而notsame与same相反，当动物不具有某种属性时为真，如（notsame 会飞 yes），当动物不会飞时为真。

一个规则，具有如下的格式：

（rule <规则名>

（if <前提>）

（then <结论>））

如“如果有羽毛则是鸟类“可以表示为：

（rule <规则名>

（if（same 有羽毛 yes））

（then （类 鸟类）））

其中R3是规则名，（same 有羽毛 yes）是规则的前提，（类 鸟类）是规则的结论。

如果前提有多个条件，则将多个谓词并列即可。如“如果是鸟类且不会飞且会游泳且是黑白色则是企鹅“可以表示为：

（rule R12

（if（same 类 鸟类）

（notsame 会飞 yes）

（same 会游泳 yes）

（same 黑白色 yes））

（then （动物 企鹅）））

也可以用（or <谓词> <谓词>）表示“或“的关系，”如果是哺乳类且（有蹄或者反刍）则属于偶蹄子类“可以表示为：

（rule R6

（if（same 有羽毛 yes）

（or（same有蹄 yes）（same 反刍 yes）））

（then （类 偶蹄类）））

这样，我们可以总结出如下规则组成知识库：

（rule R1

（if（same 有毛发 yes））

（then （类 哺乳类）））

（rule R2

（if（same 有奶 yes））

（then （类 哺乳类）））

（rule R3

（if（same 有羽毛 yes））

（then （类 鸟类）））

（rule R4

（if（same 会飞 yes）

（same 下蛋 yes））

（then （类 鸟类）））

（rule R5

（if（same 类 哺乳类）

（or（same 吃肉 yes）（same 有犬齿 yes））

（same 眼睛前视 yes）

（same 有爪 yes））

（then （子类 食肉类）））

（rule R6

（if（same 有羽毛 yes）

（or（same有蹄 yes）（same 反刍 yes）））

（then （类 偶蹄类）））

（rule R7

（if（same 子类 食肉类）

（same 黄褐色 yes）

（same 有暗斑点 yes））

（then （动物 豹）））

（rule R8

（if（same 子类 食肉类）

（same 黄褐色 yes）

（same 有黑条纹 yes））

（then （动物 虎）））

（rule R9

（if（same 子类 偶蹄类）

（same 有长腿 yes）

（same 有长颈 yes）

（same 黄褐色 yes）

（same 有暗斑点 yes））

（then （动物 长颈鹿）））

（rule R10

（if（same 子类 偶蹄类）

（same 有白色 yes）

（same 有黑条纹 yes））

（then （动物 斑马）））

（rule R11

（if（same 类 鸟类）

（notsame 会飞 yes）

（same 有长腿 yes）

（same 有长颈 yes）

（same 黑白色 yes））

（then （动物 鸵鸟）））

（rule R12

（if（same 类 鸟类）

（notsame 会飞 yes）

（same 会游泳 yes）

（same 黑白色 yes））

（then （动物 企鹅）））

（rule R13

（if（same 类 鸟类）

（same 善飞 yes））

（then （动物 信天翁）））

推理机是如何利用这些知识进行推理的呢？我们假设采用逆向推理进行求解。

首先，系统提出一个假设。由于一开始没有任何信息，系统只能把规则的结论部分含有（动物 x）的全部内容作为假设，并按照一定顺序进行验证。在验证的过程中，如果一个事实是已知的，比如已经在动态数据库中有记录，则直接使用该事实。动态数据库中的事实是在推理过程中由用户输入的或者是某个规则的结论。如过动态数据库中对该事实没有记录，则查看是否是某个规则的结论，如果是某个规则的结论，则检验该规则的前提是否成立，实际上就是用该规则的前提当作子假设进行验证，是一个递归调用的过程；如果不是某个规则的结论，则向用户询问，由用户通过人机交互接口获得。在以上过程中，一旦某个结论得到了验证——由用户输入的或者是规则的前提成立推出的——就将该结果加入动态数据库中，直至在动态数据库中得到最终的结果（动物是什么）结束，或者推导不出任何结果结束。

假定系统首先提出的假设是鸵鸟，则推理过程如图4.3所示。根据规则R11，需要验证其前提条件“是鸟类 且 不会飞 且 有长腿 且 有长颈 且 黑白色”。首先验证“是鸟类”，动态数据库中还没有相关信息，所以查找结论含有“（类 鸟类）”的规则R3，其前提是“有羽毛”。该结果在动态数据库中也没有相关信息，也没有哪个规则的结论含有该结果，所以向用户提出询问是否有羽毛，用户回答“Yes”，得到该动物有羽毛的结论。由于R3的前提条件只有这一个，所以由规则R3得出该动物属于鸟类，并将“是鸟类”这个结果加入到动态数据库中。R11的第一个条件得到满足，接下来验证第二个条件“不会飞”。同样，动态数据库中没有记载，也没有哪个规则可以得到该结论，还是询问用户，得到回答“Yes”后，将“不会飞”加入到动态数据库中。在验证“有长腿”，这时由于用户回答是“No”，表示该动物没有长腿，“没有长腿”也被放入到动态数据库中。由于“有长腿”得到了否定回答，所以R11的前提不被满足，假设“鸵鸟”不能成立。系统再次提出新的假设动物是“企鹅”，得到如图9.3所示的推理过程。根据规则R12，要验证规则的前提条件“是鸟类 且 不会飞 且 会游泳 且 黑白色”，由于动态数据库中已经记录了当前动物“是鸟类”“不会飞”，所以规则R12的前两个条件均被满足。直接验证第三个条件“会游泳”和第四个条件“黑白色”，这两个条件都需要用户回答，在得到肯定的答案后，系统得出结论——这个动物是企鹅。

如果把推理过程记录下来，则专家系统的解释器就可以根据推理过程对结果进行解释。比如用户可能会问为什么不是“鸵鸟”？解释器可以回答：根据规则R11，鸵鸟具有长腿，而你的回答该动物没有长腿，所以不是鸵鸟。如果问为什么是“企鹅”？解释器可以回答：根据你的回答，该动物有羽毛，根据规则R3可以得出该动物属于鸟类；根据你的回答该动物不会飞、会游泳、黑白色，则根据规则R12可以得出该动物是企鹅。

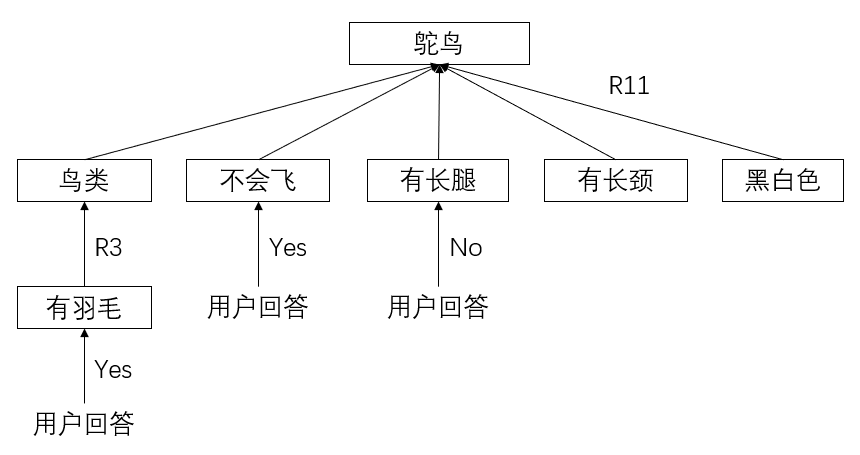


图4.3 判断是否为鸵鸟的推理过程

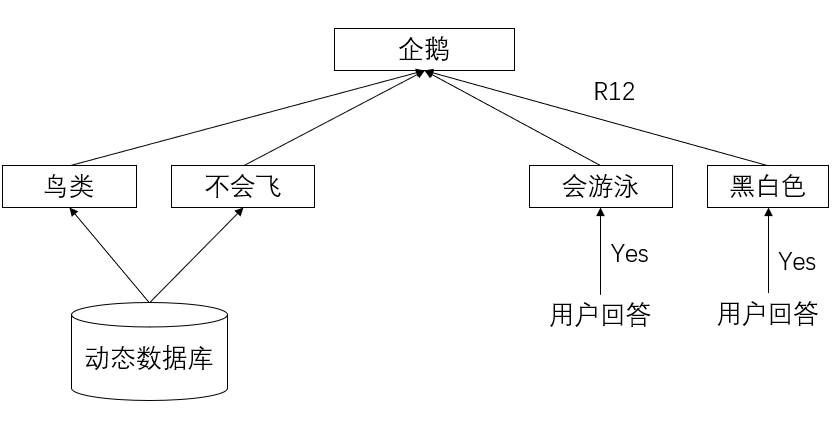


图4.4 判断是否为企鹅的推理过程

以上我们给出了一个简单的专家系统实例以及它是如何工作的。实际的系统中，为了提高效率，可能要比这复杂得多，如何提高匹配速度以提高系统的工作效率？如何提出假设以便系统尽快地得出答案？这都是需要解决的问题。更重要的一点是，现实的问题和知识往往是不确定的，如何解决不确定推理问题将在下一节介绍。

**4.4 非确定性推理**

在前面给出的一个专家系统的简单例子中，每个规则都是确定性的，也就是说满足了什么条件，结果就一定是什么。用户给出的事实也是确定性的，有羽毛就是有羽毛，会游泳就是会游泳。但现实生活中的很多实际问题是非确定性问题。比如如果阴天则下雨。阴天就是一个非确定性的东西，是有些云彩就算阴天呢？还是乌云密布算是阴天？即便是乌云密布也不确定就一定下雨，知识天阴得越厉害，下雨的可能性就越大，但不能说阴天就一定下雨。这就是非确定性的问题，回答这种问题就需要非确定性推理方法。

随机性、模糊性和不完全性均可导致非确定性。解决非确定性推理问题至少要解决以下几个问题：

\*事实的表示

\*规则的表示

\*逻辑运算

\*规则运算

\*规则的合成

目前有不少非确定性推理方法，各有优缺点，下面我们以著名的专家系统MYCIN中使用的可信度方法（certainty factor，CF方法）为例进行说明。

**4.4.1事实的表示**

事实A为真的可信度用CF（A）表示，取值范围为[-1，1]，当CF（A）=1时，表示A肯定为真；当CF（A）=-1时，表示A为真的可信度为-1，也就是A肯定为假。CF（A）>0表示A以一定的可信度为真；CF（A）<0表示A以一定的可信度且值为(-CF（A）)为假；或者说A为真的可信度为CF（A），由于此时CF（A）为负，实际上A为假；CF（A）=0表示对A一无所知。在实际使用时，一般会给出一个绝对值比较小的区间，只要在这个区间就表示对A一无所知，这个区间一般取[-0.2，0.2]CF（A）。

例如：

CF（阴天）=0.7，表示阴天的可信度为0.7。

CF（阴天）=-0.7，表示阴天的可信度为-0.7，也就是晴天的可信度为0.7。

**4.4.2规则的表示**

具有可信度的规则表示为如下形式：

IF A THEN B CF（B，A）

其中A是规则的前提；B是规则的结论；CF（B，A）是规则的可信度，又称规则的强度，表示当前A为真时，结论B为真的可信度。同样，规则的可信度CF（B，A）取值范围也是[-1，1]，取值大于0表示规则的前提和结论是正相关的，取值小于0表示规则的前提和结论是负相关的，即前提越是成立则结论越不成立。

一条规则的可信度可以理解为当前提肯定为真时，结论为真的可信度

例如：

IF 阴天 THEN 下雨 0.7

表示：如果阴天，则下雨的可信度为0.7。

IF 晴天 THEN 下雨 -0.7

表示：如果晴天，则下雨的可信度为-0.7，即如果晴天，则不下雨的可信度为0.7。

若规则的可信度CF（B，A）=0，则表示规则的前提和结论之间没有任何相关性。

例如：

IF 上班 THEN 下雨 0

表示：上班和下雨之间没有任何联系。

规则的前提也可以是复合条件。

例如：

IF 阴天and湿度大 THEN 下雨 0.6

表示：如果阴天且湿度大，则下雨的可信度为0.6。

**4.4.3逻辑运算**

规则前提可以是复合条件，复合条件可以通过逻辑运算表示。常用的逻辑运算有“与”“或”“非”，在规则中可以分别用“and”“or”“not”表示。在可信度方法中，具有可信度的逻辑运算规则如下：

1. CF（A and B）=min {CF（A），CF（B）}
2. CF（A or B）=max {CF（A），CF（B）}
3. CF（not A）=-CF（A）

1表示“A and B”的可信度，等于CF（A）和CF（B）中最小的一个；2表示“A or B”的可信度，等于CF（A）和CF（B）中最大的一个；3表示“not A”的可信度等于A的可信度的负值。

例如，已知：

CF（阴天）=0.7

CF（湿度大）=0.5

则：

CF（阴天 and 湿度大）=0.5

CF（阴天 or 湿度大）=0.7

CF（not 阴天）=-0.7

**4.4.4规则运算**

前面提到过，规则的可信度可以理解为当规则的前提肯定为真时，结论的可信度。如果已知的事实不是肯定为真，也就是事实的可信度不是1时，如何从规则得到结论的可信度呢？在可信度方法中，规则运算的规则按照如下方式计算：

已知：

IF A THEN B CF（B，A）

CF（A）

则：CF（B）=max {0，CF(A)} \*CF（B，A）

由于只有当规则的前提为真时，才有可能推出规则的结论，而前提为真意味着CF（A）必须大于0；CF（A）<0的规则，意味着规则的前提不成立，不能从该规则推导出任何与结论B有关的信息。所以在可信度的规则运算中，通过max {0，CF（A）}筛选出前提为真的规则，并通过规则前提的可信度CF（A）与规则的可信度CF（B，A）相乘的方式得到规则的结论B的可信度CF（B）。如果一条规则的前提不是真，即CF（A）<0，则通过该规则得到CF（B）=0，表示该规则得不出任何与结论B有关的信息。注意！这里CF（B）=0，知识表示通过该规则得不出任何与结论B有关的信息，并不表示对B就一定一无所知，因为还有可能通过其他的规则推导出与B有关的信息。

例如，已知：

IF 阴天 THEN 下雨 0.7

CF（阴天）=0.5

则：CF（下雨）=0.5\*0.7=0.35，即从该规则得到下雨的可信度为0.35.已知：

IF 湿度大 THEN 下雨 0.7

CF（湿度大）=-0.5

则：CF（下雨）=0，即通过该规则得不到下雨的信息。现实中可以理解为现在不能得到湿度大的前提，所以也无法通过依靠湿度大的规则来得到下雨的信息。

4.4.5规则合成

通常情况下，得到同一个结论的规则不止一条，也就是说可能会有多个规则得出同一个结论，但是从不同规则得到同一个结论的可信度可能并不相同。

例如，有以下两条规则：

IF 阴天 THEN 下雨 0.8

IF 湿度大 THEN 下雨 0.5

且已知：

CF（阴天）=0.5

CF（湿度大）=0.4

从第一条规则，可以得到CF（下雨）=0.5\*0.8=0.4

从第二条规则，可以得到CF（下雨）=0.4\*0.5=0.2

那么究竟CF（下雨）应该时多少呢？这就是规则合成问题。

在可信度方法中，规则的合成计算如下：

设：从规则1得到CF1（B），从规则2得到CF2（B），则合成后有：

CF（B）=CF1（B）+CF2（B）-CF1（B）\*CF2（B），当CF1（B）和CF2（B）均大于零

=CF1（B）+CF2（B）+CF1（B）\*CF2（B），当CF1（B）和CF2（B）均小于零

=CF1（B）+CF2（B），其他

这样，上面的例子合成后的结果为：

CF（下雨）=0.4+0.2-0.4\*0.2=0.52

如果是三个及三个以上的规则合成，则采用两个规则先合成一个，在与第三个合成的办法，以此类推，实现多个规则的合成。

下面给出一个用可信度方法实现非确定性推理的例子。

已知：

R1：IF A1 THEN B1 CF（B1，A1）=0.8

R2：IF A2 THEN B1 CF（B1，A2）=0.5

R3：IF B1 andA3 THEN B2 CF（B2，B1andA3）=0.8

CF（A1）=CF（A2）=CF（A3）=1

计算：CF（B1），CF（B2）

由R1：CF1（B1）=CF（A1）\*CF（B1，A1）=1\*0.8=0.8

由R2：CF2（B1）=CF（A2）\*CF（B1，A2）=1\*0.5=0.5

合成得到：CF（B1）=CF1（B1）+CF2（B1）-CF（B1）\*CF2（B1）=0.8+0.5-0.8\*0.5=0.9

CF（B1 and A3）=min{CF（B1），CF（A3）}=min{0.9，1}=0.9

由R3：CF（B2）=CF（B1 and A3）\*CF（B2，B1 and A3）=0.9\*0.8=0.72

答：CF（B1）=0.9，CF（B2）=0.72

**4.5专家系统工具**

专家系统的一个特点是知识库与其他部分的分离，知识库与求解的问题领域密切相关，而，推理机等则与具体领域独立，具有通用性。为此，人们开发了一些专家系统工具用于快速建造专家系统。

借助之前开发好的专家系统，将描述领域知识的规则等从原系统中“挖掉”，只保留其知识表示方法与领域无关的推理机等部分，就得到了一个专家系统工具，这样的工具称为骨架型工具，因为它保留了原有系统的主要框架。最早的专家系统工具EMYCIN（empty MYCIN）就是一个典型的骨架型专家系统工具，从名称就可以看出它是来自于著名的专家系统MYCIN。

骨架型专家系统工具具有使用简单方便的特点，只需将具体的领域知识按照工具规定的格式表达出来就可以了，可以有效提高专家系统的构建效率。但是灵活性不够，除了知识库以外，使用者不能改变其他任何东西。

另一种专家系统工具是语言性工具，提供给用户的构建专家系统需要的基本机制。除了知识库以外，使用者还可以使用系统提供的基本机制，根据需要构建具体的推理机等，使用起来更加灵活方便，使用范围也更广泛。著名的OPS5就是这样的工具系统，它以产生式系统为基础，综合了通用的控制和表示机制，为用户提供建立专家系统所需要的基本功能。在OPS5中，预先没有设定任何符号的含义以及符号之间的关系，所有符号的含义以及它们的关系均可由用户定义。其推理机制、控制策略也作为一种知识对待，用户可以通过规则的形式影响推理过程。这样做的好处是构建系统更加灵活方便，虽增加了构建专家系统的难度，但比起直接用计算机语言从头构建专家系统要方便得多。

**4.6 专家系统的应用**

专家系统是最早走向实用的人工智能技术。世界上第一个实现商用并带来经济效益的专家系统是DEC公司的XCON系统，该系统拥有1000多条人工整理的规则，帮助为新计算机系统配置订单，1982年开始正式在DEC公司使用，据估计它为公司每年节省了4000万美元。在1991年的海湾危机中，美国军队使用专家系统用于自动的后勤规划和运输日程安排，这项工作同时涉及5万个车辆、货物和人，而且必须考虑起点、目的地、路径以及解决所有参数之间的冲突。AI规划技术使得一个计划可以在几小时内产生，而用旧的方法则需要花费几个星期。

清华大学于1996年开发的一个市场调查报告自动生成专家系统也在某企业得到应用，该系统可以根据市场数据自动生成一份市场调查报告。该专家系统知识库由两部分组成，一部分知识是有关市场数据分析的，来自企业的专业人员，根据这些知识对市场上相关产品的市场形势进行分析，包括市场行情、竞争态势、动态、预测发展趋势等；另一部分知识是有关报告自动生成的，根据分析出的不同市场形势撰写出不同内容的图、文、表并茂的市场报告，并通过多种不同的语言表达生成丰富多彩的市场报告。

相比于专家系统在其他领域的应用，医学领域是较早应用专家系统的领域，像著名的MYCIN就是一个帮助医生对血液感染患者进行诊断和治疗的专家系统。我国也开发过一些中医诊断专家系统，如总结著名中医专家关幼波先生的学术思想和临床经验基础上研制的“关幼波胃脘病专家系统”等。在农业方面，专家系统也有很好的应用，在国家“863”计划的支持下，我国有针对性地开发出一系列适合我国不同地区生产条件的实用经济型农业专家系统，为农技工作者和农民提供方便、全面、实用的农业生产技术咨询和决策服务，包括蔬菜生产、果树管理、作物栽培、花卉栽培、畜禽饲养、水产养殖、牧草种植等多种不同类型的专家系统。

**4.7 课后思考题**

1. 什么是专家系统？它由哪几部分构成？

2. 专家系统的特点和优点是什么？

3. 简述正向推理和逆向推理的流程。