

第 5 篇

专家系统是如何实现的

艾博士导读

在人工智能发展的初期,由于对人工智能面临的困难估计不足,很快陷入了困境。在总结经验教训的过程中,研究者认识到知识的重要性。一个专家之所以成为该领域的专家,是因为专家具备了该领域的知识,可以熟练地运用知识解决该领域的问题。如果将专家的知识整理出来,以一种计算机可以使用的方式存放到计算机中,是不是计算机就可以使用这些知识像专家那样解决相关领域的问题了?在这样的背景下,诞生了专家系统。

所谓的专家系统,斯坦福大学的费根鲍姆教授将其定义为“一种智能计算机程序,它运用知识和推理来解决只有专家才能解决的复杂问题”。本篇内容介绍专家系统是如何实现的,结合一些实例介绍专家系统的基本概念和实现方法。

这天是周末,小明来到动物园游玩。小明看到一只可爱的动物,但是一时想不起来是什么动物,就打电话询问万能的艾博士。于是在小明和艾博士之间就产生了如下对话。

艾博士: 你看到的动物有羽毛吗?

小明: 有羽毛。

艾博士: 会飞吗?

小明: (经观察后)不会飞。

艾博士: 有长腿吗?

小明: 没有。

艾博士: 会游泳吗?

小明: (看到该动物在水中)会。

艾博士: 颜色是黑白的吗?

小明: 是。

艾博士: 这个动物是企鹅。

小明觉得这个问题挺有意思的,在从动物园回来的路上,就开始反复思考这个问题:艾博士怎么就想到了询问这几个问题?为什么最后就确认是企鹅?小明想在现实生活中也有很多类似的问题,比如识别花草、医生看病等,似乎都是差不多的过程。就拿医生看病为例,医生往往先问病人几个问题,根据病人的回答再问一些新的问题,经过一番诊断之后,会建议病人去做B超、CT等检查,再根据检查结果确认病人得的是什么病,从而给病人医疗建议。医生给病人看病的过程跟上面认识企鹅的过程基本差不多,虽然可能要复杂得多。小

明就想这个过程是否可以用人工智能方法实现呢？带着这样的问题，小明一回到家就又来请教艾博士。

5.1 什么是专家系统

明白小明的来意之后，艾博士解释说：这样的系统在人工智能中叫作专家系统，是人工智能研究的一个重要方向，在人工智能历史上起到过举足轻重的作用，人工智能技术应用于解决实际问题，就是从专家系统开始的。

小明：什么是专家系统呢？

艾博士：在人工智能发展初期，由于对实现人工智能的难度估计不足，人工智能的研究很快就陷入困境。在总结经验教训时，研究者逐渐认识到知识的重要性。一个专家之所以是专家，之所以能求解本领域的问题，重要的是具有该领域的专门知识。如果将专家的知识总结出来，并以计算机可以使用的方式表示出来，那么计算机不就可以像专家那样利用这些知识求解问题了吗？这就是专家系统的由来。

小明：专家系统原来是这么来的。

艾博士：在这样的思想指导下，1965年，斯坦福大学的费根鲍姆教授和化学家勒德贝格教授合作，研发了世界上第一个专家系统 DENDRAL，用于帮助化学家判断某待定物质的分子结构。之后，费根鲍姆教授领导的小组又研制了著名的专家系统 MYCIN，该系统可以帮助医生对住院的血液感染患者进行诊断和选用抗生素类药物进行治疗。可以说 MYCIN 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。XCON 是最早投入实际使用的专家系统，是由 R1 专家系统发展而来，该系统可以按照用户的需求，帮助 DEC 公司为其生产的 VAX 型计算机系统自动选择组件。

作者也曾经多年从事专家系统的研究工作，先后研制过火车编组站专家系统、货物轮船积载系统、雷达故障诊断专家系统和市场报告自动生成专家系统等，并在一些企业得到了应用。

小明：专家系统都有哪些特点呢？

艾博士：专家系统研究的先驱、图灵奖获得者费根鲍姆教授将专家系统定义为：一种智能的计算机程序，它运用知识和推理来解决只有专家才能解决的复杂问题。这里的知识和问题，都属于同一个特定领域。

从该定义可以看出，首先专家系统是一个计算机程序，但又不同于一般的计算机程序，专家系统以知识库和推理机为核心，可以处理非确定性问题，不追求问题的最佳解，利用知识得到一个满意解是系统的求解目标。专家系统强调知识库与包括推理机在内的其他子系统的分离，一般来说知识库是与领域强相关的，而推理机等子系统具有一定的通用性。

一个专家系统的基本结构如图 5.1 所示。

小明：请艾博士具体解释一下这个专家系统的基本结构，各个组成部分都是什么含义。

艾博士：我们先简单介绍一下这个基本结构，让大家对专家系统有一个基本了解，后面还会结合具体内容做详细说明。

知识库用于存储求解问题所需要的领域知识和事实等，知识一般以如下形式的规则

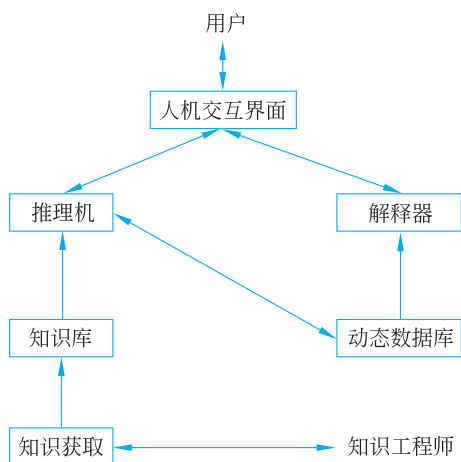


图 5.1 一个专家系统的基本结构

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 且 湿度大 then 下雨
```

这里的“阴天 and 湿度大”就是前提,“下雨”就是结论,表示“如果阴天 and 湿度大,则会下雨”这样一条知识。

当然这是一条确定性的规则,实际问题中规则往往不是确定性的,而是具有一定的非确定性,关于非确定性的规则表示问题,我们将在后面叙述。

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 下雨 then 带上雨伞
```

表示的是“如果下雨了出门要带上雨伞”。

也可能是其他的类型,比如删除某个数据、替换某个数据等。比如一个老年人健康护理专家系统,早上的时候可能记录的是老人没有吃药,一旦老人吃药后,就要从记录中删除“没有吃药”这条信息,并增加“已吃药”信息。

推理机是一个执行机构,它负责对知识库中的知识进行解释,利用知识进行推理,相当于人的大脑。例如,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检测,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的操作。

动态数据库是一个工作存储区,用于存放初始已知条件、已知事实和推理过程中得到的中间结果,以及最终结果等。知识库中的知识在推理过程中所用到的数据以及得到的结果,均存放在动态数据库中。

人机交互界面是系统与用户的交互接口,系统在运行过程中需要用户输入的数据,用户通过该交互接口输入到系统中,系统需要显示给用户的信息通过该交互接口显示给用户。

讲解到这里艾博士问小明：小明如果去看医生，你会信任医生的诊断结果吗？为什么信任他？

小明回答说：首先医生是看病的专家，对于医生的诊断结果我还是比较信任的，但是也会向医生提出一些问题，请医生解释说明。

艾博士：对，医生不仅会诊断你有什么病，还会向你解释为什么得的是这种病，病人之所以会信任医生的诊断结果，与医生的解释是分不开的。所以具有解释能力也是专家系统的重要特征。解释器是专家系统特有的负责解释的模块，也是与一般的计算机软件系统的区别之一。在专家系统与用户的交互过程中，如果用户有需要系统解释的内容，专家系统通过解释器对用户进行解释。解释一般分为 Why 解释和 How 解释两种，Why 解释回答“为什么”这样的解释，How 解释回答“如何得到的”这样的解释。例如，在一个医疗专家系统中，系统给出让病人验血的建议，如果病人想知道为什么让自己去验血，用户只要通过交互接口输入 Why，则系统会根据推理过程，给出为什么会让病人去验血，让用户明白验血的意义。如果专家系统最终诊断病人患有某种疾病，病人想了解专家系统是如何得出这个结果的，只要通过交互接口输入 How，则专家系统会根据推理过程，对用户做出解释，根据什么症状判断用户患有的是这种疾病。这样可以让用户对专家系统的推理结果有所了解，而不是盲目信任。“可解释”是专家系统中非常重要的组成部分。现在很多数据驱动的人工智能系统，大多是黑箱模型，对结果缺乏可解释性，可解释性也是目前人工智能领域一个重要的研究课题。

听艾博士这样讲解后小明点点头说：具有解释能力确实是专家系统的重要特征，不能盲目信任专家系统的结论，必须给出合理的解释才可以获得信任。

艾博士：知识获取模块是专家系统与知识工程师的交互接口，知识工程师通过知识获取模块将整理的领域知识加入到知识库中，也通过知识获取模块对知识进行管理和维护。专家系统主要是依靠人工整理获取知识。

小明：那么专家系统是怎样一个工作流程呢？

艾博士：专家系统一般都是某个领域的专用系统，即便是医疗领域的专家系统，也会像医生看病一样，划分为几个专科，每个专科看专门的疾病。比如我国曾经建造过一个“关幼波肝病诊疗程序”的专家系统，就是根据著名肝病诊疗专家关幼波大夫的经验建造的一个专家系统。

对于一个已经建造好的专家系统，因应用领域的不同其工作流程可能会有一些差别，一个基本流程是这样的。

用户根据自己的需要选定一个专家系统，输入一些基本情况。比如以看病为例，可能要先告知自己身体哪里不舒服，有哪些症状等，专家系统会根据用户提供的基本信息做出一些判断，询问用户一些更详细的问题，或者让用户做些必要的检查。经过几轮交互之后，最终专家系统会给出一个结果，确诊是什么疾病，给出治疗方案。如果在这个过程中用户有哪些疑问，均可以通过解释器与系统做交互，得到专家系统的解答。其他的应用场景也是类似的过程。

小明：这个流程确实跟我们看病过程差不多。

艾博士：是这样的，专家系统就是某种程度上对人类专家的模仿。前面咱们两个关于

识别动物的那个对话过程,就是一个典型的专家系统工作流程,你完全可以把我当作一个动物识别专家系统看待。

小明: 谢谢艾博士的耐心解答,我对专家系统有了初步的认识。

艾博士: 下面详细地介绍每个部分的具体实现方法,由于有很多不同的实现方法,我们选择一些相对简单又有代表性的方法加以介绍。

小明读书笔记

一个领域的专家,之所以能成为该领域的专家,是因为掌握了该领域的相关知识,可以利用这些知识解决该领域的问题。如果能将专家的知识整理出来,以计算机可以使用形式存储到计算机中,那么计算机也可以使用这些知识像专家一样解决该领域的问题。这就是专家系统。

一个专家系统基本由知识库、推理机、动态数据库、解释器、人机交互界面等几部分组成。

知识库用于存储知识,最常用的知识表示形式是规则。规则具有如下形式:

if <前提> then <结论>

表示当<前提>成立时,<结论>成立。

例如:

if 阴天 and 湿度大 then 下雨

表示的是“如果阴天并且湿度大,则会下雨”这样的一条规则。

推理机是专家系统的执行机构,负责对知识库中的知识进行解释,利用知识进行推理。

动态数据库是一个工作存储区,专家系统工作中获得的数据,包括初始的已知条件、已知事实、用户的输入、推理过程中获得的结论等,均存放在动态数据库中。

人机交互界面负责人与计算机的交互,系统运行过程中需要用户输入的数据通过人机交互界面输入,系统需要显示给用户的信息也通过人机交互界面显示给用户。

解释器是专家系统特有的组成部分,可以对专家系统获得的结论进行解释。一般至少有 Why 解释和 How 解释两种解释功能。Why 解释负责回答类似于“为什么这样做”这样的问题,How 解释负责回答类似于“如何得到的”这样的问题。

5.2 推理方法

艾博士: 专家系统的推理机就相当于我们的大脑,具有一定的通用性,与具体的任务领域无关。就像我们人类一样,具有了哪个领域的知识就可以求解哪个领域的问题,也就成为了哪个领域的专家,但是大脑都是一样的,具有通用性,只是应用的知识不一样。

小明: 专家系统中的推理机是如何利用知识库进行推理的呢?

艾博士: 推理机制与具体的知识表示方法有关,根据知识表示方法的不同推理方法也会有所不同。在专家系统中,规则是最常用的一种知识表示方法,下面以规则为例展开说明。

小明：规则就是前面提到过的如下这种形式吗？

```
if <前提> then <结论>
```

艾博士：是的，规则就是这种形式，其中<前提>是一些条件的逻辑组合，包括“与”“或”“非”等，而<结论>是某种可能的结果或者某些动作等。

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，从已知条件出发，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到动态数据库中，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，是规则的一种正向使用形式，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。

例如，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。

其中 r1、r2 等是规则名，“A and B”“C and D”等是规则的前提，“A and B”表示“A 与 B 同时成立”，“C and D”表示“C 与 D 同时成立”。

我们看看，如果采用正向推理的方法，如何根据这些规则，从已知条件推导出目标 F 成立。

初始时 A、B、D 在动态数据库中，由于 A 与 B 均成立，规则 r1 的前提成立，所以由规则 r1 推导出 C 成立，并将 C 加入动态数据库中。由于 D 是已知成立的，刚刚又由规则 r1 推导出了 C 成立，所以规则 r2 的前提成立，根据规则 r2，推出 E 成立，将 E 加入动态数据库中。这样规则 r3 的前提也是成立的，根据规则 r3，推出 F 成立，将 F 加入动态数据库中。由于 F 就是求证的目标，所以结论成立，推理结束。这就是正向推理过程，图 5.2 给出了该过程的示意图。

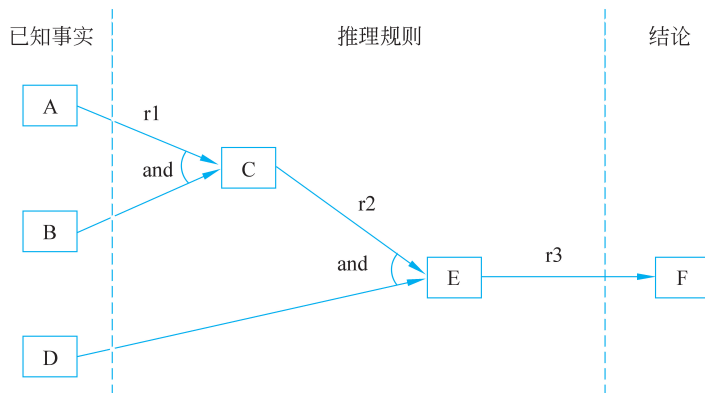


图 5.2 正向推理示意图

小明：我有一个问题，如果在推理过程中，同时有多个规则的前提都成立，这时如何选择规则呢？

艾博士：如果在推理过程中，有多个规则的前提同时成立，如何选择规则称为冲突消解问题。最简单的办法是按照规则的自然顺序，选择第一个前提被满足的规则执行。也可以对多个满足条件的规则进行评估，优先选择前提条件多的规则执行。

小明：这是为什么呢？

艾博士：由于这样的规则涉及的前提条件比较多，不容易被满足，一旦前提条件被满足，其结论可能是一个比较重要的结果。

也可以从规则的结论距离要推导的结论的远近来考虑，这里说的“远近”是指一旦有了该结果，还需要应用多少条规则才能推导出最终结论。距离目标越近的规则越是要优先执行。也可以人为地对每条规则的重要程度做出规定，重要的规则具有较高的优先级。比如说有如下 3 个规则：

```
r1: if 生病 then 休息
r2: if 病重 then 去看医生
r3: if 昏厥 then 打电话叫 120
```

这 3 条规则的重要性程度显然是不一样的，当 3 条规则的前提均被满足时，应该优先执行规则 r3，打电话叫 120 抢救。这种情况下，可以在构建知识库时，对规则的重要性进行评价，给出优先级，当发生冲突时按照优先级执行规则。

小明：有正向推理，是不是也有逆向推理呢？

艾博士：与正向推理对应的就是逆向推理。逆向推理又被称为反向推理，这种推理方法的特点是逆向使用规则。

小明：我不是太明白，逆向使用规则是什么意思呢？

艾博士：我们举例说明。假设有规则：

```
if A and B then C
```

我们想知道 C 是否成立，就看该规则的前提是否成立，为此就要看 A、B 是否成立。为了知道 A 是否成立，就要看是否某个规则的结论为 A，然后看该规则的前提是否成立。这就是逆向使用规则。

小明：我明白了，所谓逆向使用规则，就是从规则的结论出发，反过来看规则的前提是否成立，一步一步由后向前推，看是否满足条件。

艾博士：在逆向推理中，按照逆向使用规则的思想，首先将求证的目标作为假设放入假设集中，查看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，然后看结论与假设一致的规则其前提是否成立。如果前提成立（在动态数据库中进行匹配），则假设被验证，结论放入动态数据库中，否则将该规则的前提加入到假设集中，一个一个地验证这些假设，直到目标假设被验证为止。由于逆向推理是先假设目标成立，逆向使用规则进行推理的，所以这种推理方法又称为目标驱动的推理。

例如，在前面正向推理的例子中，如何使用逆向推理推导出目标 F 成立呢？小明我们

看一下推导过程。

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E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的结论可以推出E,依次检验r2的前提C和D是否成立。首先检验C,由于C也没有在动态数据库中,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的前提条件全部被满足,推出E成立,并将E放入动态数据库中。由于E已经被推出成立,所以规则r3的前提也成立了,从而最终推出目标F成立。这就是逆向推理过程,图5.3给出了该过程的示意图。

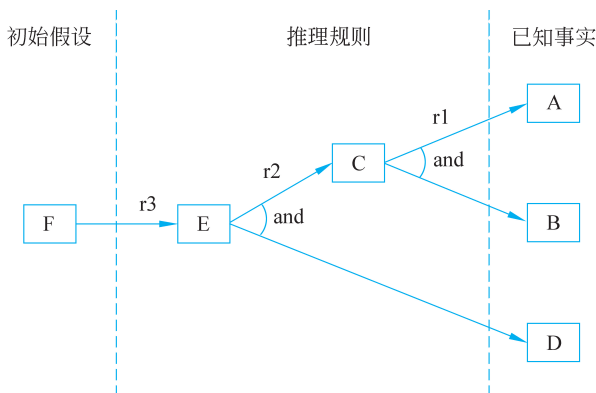


图 5.3 逆向推理示意图

小明：逆向推理是不是也和正向推理一样，存在冲突消解问题呢？

艾博士：在逆向推理中也同样存在冲突消解问题，想验证某个假设是否成立时，可能多个规则的结论与该假设有关，优先选择哪个规则呢？可采用与正向推理一样的方法解决。

在具体实现时，正向推理一般采用类似宽度优先搜索的方式，一步一步由已知条件向目标结论推进，而逆向推理则一般采用类似深度优先搜索的方式，一步一步产生假设，从目标结论开始反向使用规则，对假设进行验证。这里的宽度优先、深度优先搜索与第3篇介绍的方法会有些不同，因为规则的前提中会有“与”“或”等逻辑运算，在搜索过程中需要考虑这些因素，但基本思想是一样的，很容易扩展过来。

一般的逻辑推理都是确定性的，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、非确定性。比如，如果阴天则可能下雨，阴天了不一定就肯定下雨。这就属于非确定性推理问题。关于非确定性推理问题，我们将在后面做详细介绍。

小明读书笔记

最基本的推理方法包括正向推理和逆向推理两种。所谓的正向推理是从事实出发，正向使用规则，一步一步地推出结论。正向推理又称作是数据驱动的推理。

逆向推理是先提出假设，逆向使用规则验证能得出该假设的规则前提是否成立。如果不能直接验证相关规则的前提是否成立，则进一步将相关规则的前提作为新的假设添加到假设集中，直到所有的假设被验证得出结论，或者一些假设被否定得不出结论为止。

逆向推理又被称作目标驱动的推理。

无论是正向推理还是逆向推理均存在冲突消解问题。所谓的冲突消解,就是当同时有多个规则满足触发条件时,如何选择一个规则执行。可以有多个冲突消解解决策略,比如按照规则的排列顺序、按照规则前件已经满足的程度,按照规则优先级等。

5.3 一个简单的专家系统

艾博士：下面通过一个简单的实例,说明一下专家系统是如何构建和工作的。

我们再来看看本篇开始时有关动物识别的一段对话。

艾博士：你看到的动物有羽毛吗？

小明：有羽毛。

艾博士：会飞吗？

小明：（经观察后）不会飞。

艾博士：有长腿吗？

小明：没有。

艾博士：会游泳吗？

小明：（看到该动物在水中）会。

艾博士：颜色是黑白的吗？

小明：是。

艾博士：这个动物是企鹅。

在以上对话中,当小明告诉我动物有羽毛后,依据所掌握的知识,我就知道了该动物是鸟类,于是我就向小明提问该动物是否会飞,当小明回答说不会飞后,我最先猜到的是这个动物可能是鸵鸟,于是再次问小明该动物是否有长腿,因为如果有长腿的话,很大可能就是鸵鸟了。在得到小明的否定回答后,我马上意识到鸵鸟这个猜测是错的,于是就想到了可能是企鹅。就接着询问小明动物是否会游泳,在得到小明的肯定答复后,我已经感觉到这个动物大概率是企鹅了。为了进一步确认是否是企鹅,又继续问小明动物的颜色是否是黑白的。小明回答是黑白颜色后,我马上就确认了该动物是企鹅。

在我和小明的这个对话过程中,我首先提问是否有羽毛,目的是先区分出是否鸟类。是否有羽毛这个问题既容易观察,又可以比较大规模地缩小猜测范围,因为无论小明回答是否有羽毛,都可以排除掉很多不相关的动物。如果一上来就提问是否是长腿,就很难达到这样的效果。然后再一步步地,根据小明已有的回答,猜测可能是什么动物,再逐一确认或者否定。这个过程就是一个“猜测—提问—回答—再猜测—再回答”的循环过程。

一个动物识别专家系统,我们也希望能像上面的对话一样,系统通过与用户的交互,回应用户有关动物的问题。

小明：如何实现这样的专家系统呢？

艾博士：为了实现这样的专家系统,首先要把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。这些知识包括我们自己掌握的、书本上学来的以及向动物专家请教来的等。可以用规则表示这些知识,为此,我们设计一些表达式,以便方便地表达知识。

小明不解地问道:表达式?这是什么意思呢?

艾博士: 为了方便表达规则的前提和结论而设计的一些表达式,一般具有如下三元组的形式:

```
(<名称> <属性> <值>)
```

在实际表达规则时,其中的<>部分要用具体的内容代替。下面给出几个表达式的例子就容易明白了。

首先是 same,表示动物具有某种属性,比如,可以用(same 羽毛 有)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,比如(notsame 飞翔 会),当动物不会飞翔时为真。

一条规则,一般表达为如下形式:

```
(rule <规则名>
  (if <前提>)
  (then <结论>))
```

其中的<前提>、<结论>均可以用表达式表示。

比如“如果有羽毛则是鸟类”,可以表示为:

```
(rule r3
  (if (same 羽毛 有))
  (then (动物 类别 鸟类)))
```

其中 r3 是规则名,(same 羽毛 有)是规则的前提,(动物 类别 鸟类)是规则的结论。这里用到的“same”“动物”等是表达式的名称,“羽毛”“类别”等是表达式的属性,而“有”“鸟类”等则是属性的值。

如果前提有多个条件,则可以通过多个表达式的逻辑组合表示。

比如“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 类别 鸟类)
      (notsame 飞翔 会)
      (same 游泳 会)
      (same 黑白色 是))
  (then (动物 是 企鹅)))
```

也可以用(or <表达式> <表达式>)表示“或”的关系,比如:

```
(rule r6
  (if (same 类别 哺乳类)
      (or (same 蹄 有) (same 反刍 是)))
  (then (动物 子类 偶蹄类)))
```