

用逆向链方法开发的专家系统

陆 佃 刘梅彦 (北京机械工业学院)

摘要:使用面向用户的逆向链方法来处理规则,在具有内置式推理机的开发工具 *Vp-Expert Shell* 环境下,较大地减少了在微机上建立专家系统的工作量,有较好的普及性及用户友好性。从理论上讨论了专家系统的质量取决于关键因素的全面性、原因的充分性、以及建议的准确性。

1. 引言

在计算机科学的人工智能——Artificial Intelligence (AI)中,专家系统(Expert System 或 KBS——knowledge Based System)是其中的一个分支。从六十年代开始研究,目前已得到广泛应用。

专家系统主要由三部分组成:知识库/推理机和解释器。知识库是信息的集合,它存储着所开发领域的专门知识。推理机是专家系统的程序部分,负责处理推理的逻辑过程,按照一定算法适时地索取知识库中的规则,适时地按约定的方式向用户提出问题。按照用户的回答,利用知识库中的知识推理后向用户提出结论。解释器用于存储用户咨询过程中的历史文件,系统根据历史文

件确切报告如何得出一个结论及原因。

2. 逆向链

可以使用各种计算机语言来开发专家系统。例如可以使用 Lisp、Prolog、C...等语言。但是程序员在用这些语言建立知识库和解释器,构造推理机制时,不但工作量很大,而且若要使所开发的系统能应用于多种领域难度就更大。为克服这些困难计算机工作者开发工具已经具有了知识库、解释器和推理等机制,专家系统的建立者只要把某领域的知识输入进去,就可以得到该领域的专家系统,提供给用户使用。

这些开发工具有适用于各种大、中、小型计算机,有的适用于微型计算机。

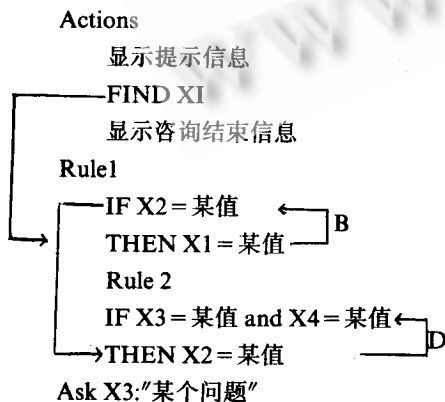
Vp-Expert 是在微机环境下,开发专家系统的一种

工具软件。它有内置式的推理机制,开发人员利用 Vp-Expert Shell 提供的操作命令书写“知识库文件”(KBS),就可建立起他所选定领域的专家系统。例如人体健康诊断、汽车故障诊断、投资判断系统……。

知识库文件由三部分组成:活动块(Actions)、规则块(Rules)、问题块(Questions)。每部分均由相应的语句组成。Vp-Expert 有丰富的语句,例如活动块中的 FIND 语句、规则块中的 IF-THEN 语句、问题块中的 ASK 语句和 CHOICE 语句。另外它还提供了各种函数和运算符。

活动块相当于高级语言主程序,活动块中 FIND 关键字后的变量称为目标变量。规则块由大量规则组成。每个规则由条件部分和结论部分组成,如果规则的条件得到满足,则此规则被“激活”。问题块用来回答目标变量或子目标变量,并给这些变量赋值,从而得到推理结果。

专家系统的“内置式”推理机工作时要构造逆向链。推理机在用户咨询阶段处理规则的机制称为“逆向链”,其形式如图中的 A→B→C→D。通过构造逆向链及目标变量被赋值的过程,可以看到 Vp-Expert Shell 的推理机制。从解决目标变量逆向链开始构成,得到目标变量后,推理机会到规则块中去寻找结论部分有此目标变量的规则,逆向链延伸到此规则的条件部分,将此规则的条件部分的变量作为子目标变量。继续这样的寻找,直到找不到这样的规则,此时逆向链构造完毕。推理机会跳到问题块中去寻找包含需知其值的子目标变量的 ask 语句,以从用户那里得到此子目标变量的值。如果仍未得到此子目标变量的值,则此次咨询过程失败。只有所有子目标变量的值确定后,结论部分包含目标变量的规则才能“激活”,并且控制会返回到“活动块”中。用图 1 某屏的显示可以说明逆向链构成。



CHOICE X3: Yes,No ;选择值

Ask X4: "某个问题"

CHOICE X4: Yes,No ;选择值

图 1 逆向链

解开链,用户使用此专家系统是系统运行解开链的过程,借助于与用户交互的 CHOICE 语句根据用户对 Yes 或 No 的选择,激活规则块中的相应规则,子目标变量得到赋值,执行其结论部分,再自动沿逆向链的相反方向,去激活与其相关的下一个规则,如此连续下去,直到目标变量得到赋值,此时程序会自动返回到 Actions 块中,执行 display 语句,向用户显示结果。

应当指出每个专家系统中,FIND 语句中的目标变量可能是多个,规则量很大,至少要建立几千条才能达到实用的水平。

3.用 Vp-Expert Shell 开发 KBS 的步骤

用开发工具开发专家系统的主要工作是获取专家的知识,并加以整理、分析,使之成为计算机可以接受的形式。开发专家系统需要计算机工程人员根据专家和知识构造知识库文件,并最终在计算机上实现专家系统。

Vp-Expert Shell 非常适合于开发在微机上使用的小型专家系统;也适用于在开发大系统时,先用它开发出原型系统,以测试一个大系统的实用性及可行性。常用来开发专家系统的领域有商业、医学、化工、计算机等。假如拟开发一个微机故障诊断专家系统,为遇到微机问题的用户提供帮助,首先要对微机可能出现的多种故障范围逐个进行分析、综合,可从中选出一种来开发一个原型系统。例如选定“系统已经启动但没有 C:/ >提示”。首先分析引起故障的原因,可能是由电源、监视器、RAM、硬盘、系统设置、软件等故障引起的,而每个部分出故障的原因又可能有很多。对所有情况综合、归纳,最后给出解决这种故障的建议。依此对可能出现的故障均予以解决,就得到了完整的微机故障诊断专家系统。

开发专家系统可以划分为三个阶段 / 八个步骤:

(1)系统分析阶段

- 确定要开发专家系统的领域和目标;
- 划分功能块、生成区域块图;
- 确定可原型化的目标;

(2)系统设计阶段

- 生成依赖图;

- 生成决策表;
- 写出知识规则;
- 构造用户界面;

(3)系统实现阶段

把以上在纸上形成的专家系统素材,在微机上予以实现。

在确定了开发领域和目标后,要进行的第二步工作是划分功能块,生成区域块图。

首先对所选定领域的运行机制进行概括,分析,生成一张功能模块图,并从中选择可用来开发最初原型的区域块。功能模块图如图2。

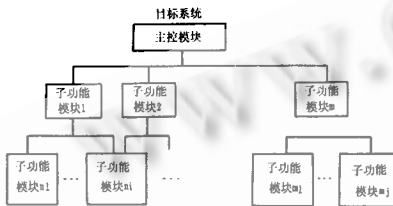


图2 划分模块图

第三步:确定可原型化的目标

一旦一个领域被划分出来进行KBS的开发,必须准确地确定此领域中存在的问题和各种可能出现的情况。一般称与要解决的问题有直接关系的原因或元素为关键因素。关键因素通常不只一个,这里用K来代表关键因素,即可能有K1,K2,...,Km共m个关键因素。而且与每个关键因素相关的情况或原因通常也不只一个,用R代表原因,即可能有R1,R2,...,Rn共n个原因。对于用户的询问,原型系统最终提供给用户一些建议,用S1,S2,...,Sp表示。

使用软件开发界面的图解功能,画出原型化的决策情况图如图3,可使专家系统的开发工作清晰明了。

第四步:生成依赖图

依赖图是由第二步中的原型化的决策情况图转化来的。图4就是由图3转化来的。依赖图明确了关键因素、输入的问题、规则、数值及由KBS原型给出的建议等之间的依赖关系。它是KBS专家系统的一个完全的图形描述,为书写实际的知识库代码提供了一个纸上模型。

通过画依赖图,可以迫使KBS开发的每个疑难问题得到解决,也许这会引起第三步的开发工作推倒重来,但无论经过多少次的反复,最终生成一个可信的依赖图,将为下一步的开发打下良好的基础。

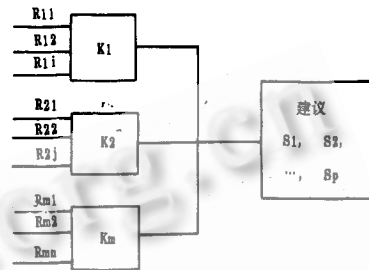


图3 原型化的决策情况图

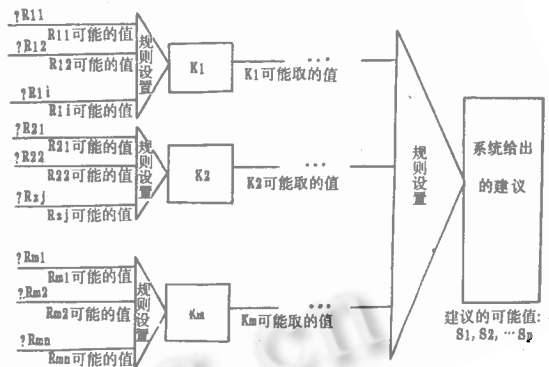


图4 依赖图

第五步:生成决策表

为依赖图中的每一个三角形生成一个决策表,是建模过程的最后一步;它是下一步制订规则的基础,决策表对于指示专家系统的最后结论以及任何中间过程的结果都是必要的。

假设上步中的依赖图中的三角形的个数为X个,则要开发的专家系统原型中决策表的个数为X个。制作决策表的过程如下:依赖图中每个三角形对应一个决策表。首先确定每个决策表的行数(用ROW表示)。

$$ROW = \text{规则的条件数} \times \text{每个条件可能的值}$$

而后按此行数画出一张只有行号的表格。如下表,其中最后一列为标识和组合的可能的结果值。即:

规则	条件 1	条件 2 ...	条件 m	关键因素 K_i
A1	yes	yes ...	yes	OK 或 not-OK
A2	yes	yes ...	No	OK 或 not-OK
:	:	:	:	:
An				

上述表中若存在可以合并的条件,就必须进行简化工作。

第六步:写出 IF-THEN 规则

知识库的规则语句中包括关键字 Rule、IF、THEN、AND、OR、ELSE 等,还包括变量和运算符。当规则中的所有条件满足后,这条规则就被“激活”。规则是知识库的重要组成部分,是 Vp-Expert 的推理机进行推理的依据。推理机利用知识库中的规则构造逆向链,从而得到目标变量的值。

书写规则时,尽量避免使用 ELSE 子句,因为涉及到具体情况,并不总是即此非彼,情况往往很复杂。如果使用 ELSE 子句,则 ELSE 的条件被省掉,使产生可能结果的条件变得含糊不清,并且可能在用户咨询过程中产生不合逻辑的问题序列。

对于缺乏开发 KBS 经验的人来说,一个好的方法是为简化的决策表中每一行书写一个规则,随着开发经验的积累,可以探索出一条提高书写规则和执行一次知识库查询效率的捷径。

第七步:构造用户界面

用户接口主要包括系统开始的信息、系统结束的信息和在咨询过程中系统提出的问题。

在知识库的活动块中包含有系统开始信息、目标变量和系统结束信息。Vp-Expert 提供了关键字 display 来显示系统提示信息。

构造在咨询过程中提出的问题,可从系统期望的输入中引出来。问题构造好后,还要在系统运行的咨询阶段加以测试,以考察其合理性和必要性。把一个问题转化为知识库中的代码需要两个新的关键字:ask 和 choice,系统提出的问题用变量来表示。ask 后紧跟问题变量,choice 与 ask 一一对应,choice 为 ask 中变量提供初始值。ask 和 choice 语句共同构成知识库的一个独立的问题块。

第八步:系统的实现

通过以上七步,一个纸上的专家系统已经形成。将写好的知识库文件通过文本编辑软件输入到计算机中,形成一个 KBS 文件。在 Vp-Expert 环境中可以运行 KBS 文件,从而完成用户咨询工作。

至此,已经生成了一个专家系统的知识库代码,一个专家系统已经基本上建立了,但系统性能及使用的有效性,需要实践来检验。一个真正实用的系统往往需要经过反复修改,不断完善。

4.专家系统的质量

专家系统的质量取决于多方面,此处讨论其中一个主要的方面。一般而言,实际问题的原因可达成千上万个,应该讨论决策图与依赖图(图 3-2,图 3-3)中关键因素 K 的全面性/原因 R 的充分性和建议 S 的准确性。

这里假设解决问题 Q 的专家系统为 CV,符号“~”表示相关,A~B 表示 A 与 B 相关。

对于问题 Q 首先存在关键因素集合:

$$K = \{K_1, K_2, \dots, K_m\}$$

其中每一个 K_i 都对应原因集合:

$$R_i = \{R_{i1}, R_{i2}, \dots, R_{im}\}$$

其中 $R_{ij} \rightarrow K_i \quad R_{ij} \in R_i \quad j=1..m$

即原因 R_{ij} 可导致 K_i 。

令

$$R = R_1 \cup R_2 \cup R_3 \cup \dots \cup R_m$$

则 R 与对应的建议集:

$$S = \{S_1, S_2, \dots, S_p\}$$

有图 3 的关系。

因此问题 Q 的专家系统 V 建立的过程可归结为确定关键因素集合 K,原因集合 R,并建立建议集合 S。所以专家系统 V 的质量取决于 K 的全面性、R 的充分性以及 S 的准确性。

5.结语

使用面向用户的逆向链方法来运行的开发工具来建立专家系统,比用高级语言要便利得多。能使开发者集中精力去提高专家系统的实用性和质量。而且开发工具可以开发不同领域的各种系统。

参考文献:

[1] Dr. Dologite: Using Vp-Developing Knowledge-based System MACMILLAN 1993

[2] 施鸿宝等:专家系统