

基于关系数据库的知识库的设计

黄晓婧, 黄 艳

(湖北大学 数学与计算科学学院, 湖北 武汉 430062)

摘 要: 文章利用数据库和知识库各自的特点, 将数据库技术和专家系统相结合, 把知识存贮在关系数据库中。文章从知识的表示出发, 给出一种简单的基于关系数据库的知识库的设计。并且说明了怎么样利用 SQL 语言对知识库进行正向推理, 文章最后给出了一个利用这种知识库进行推理的例子, 并实现了编程。

关键词: 知识库; 关系数据库; 知识表示

中图分类号: TP311.13

文献标识码: A

文章编号: 1009 - 5160(2007) - 0013 - 04

在人工智能领域中, 专家系统是一个重要的组成部分, 而知识库又是专家系统的核心和基础, 因此, 一般把专家系统也称为基于知识库的系统, 知识库质量的好坏决定着整个专家系统的成败。

随着专家系统的不断发展, 在现代的知识库开发过程中, 需要面对大量的知识, 若以文件 I/O 的形式组织时局显得非常的复杂和困难, 并且不能简便有效的维护和扩充数据, 因此在专家系统中采用数据库是一种必然的选择。利用数据库技术, 将各种知识和规则系统化、规范化, 建立知识库, 随着经验的丰富可以不断的扩充知识库, 体现了知识库的健壮性, 并且有益于知识库维护的一致性和完整性。

本文首先利用数据库和知识库各自的特点, 对数据库和知识库进行比较, 将数据库和知识库技术相结合, 把知识存贮到数据库中, 然后从知识的表示出发, 给出了一种简单的基于关系数据库的知识库的设计, 并具体说明了怎样利用 SQL 实现正向推理。

1 知识库与数据库的比较

数据库指使用 DBMS 及其类似软件建立起来的, 并能存取和维护的数据及数据间逻辑关系的集合体。知识库是 AI 和数据库技术相结合的产物。它是以一致的形式存储知识的机构。知识库中的知识是高度结构化的符号数据。

关系数据库采用二维表格来表示实体类型及实体间联系的, 表示一些基本的事实数据, 其数据量非常大, 一般需要较频繁地修改以及需要并发的进行存取等操作, 但关系数据库中的数据缺乏规则的联系, 例如在某确定的规则、条件下, 关系数据库推导不出新的数据项, 它不支持专门的推理机制和带控制的搜索, 因此, 关系数据库的推理能力比较差。

从知识库的角度来看, 知识库的基本内容可描述如下: $KB=F+R$ 。其中 KB 代表知识库; F 代表事实集, 用数据库的术语来说, 相当于实体及其属性所对应的数据集合; R 代表规则集, 其中所包含领域知识或启发式知识, 在数据库中, 通常是隐含于数据结构或完整性约束中, 而在知识库中, 则主要是通过某种知识表示方式, 与事实一样显式地表达出来。

可见数据库与知识库既有区别又有联系, 它们中间没有一条泾渭分明的界限。数据库中所包含的明显表达的关系的约束条件, 可以视为某种低层次的知识; 知识库中所包含的有明确定义的事实、断言和通过规则表达的某些明显的关系实际上也是数据库的内容。

收稿日期: 2007-01-07

作者简介: 黄晓婧 (1982-), 女, 硕士研究生, 研究方向: 人工智能。

基于关系数据库技术的知识库的优越性主要表现在以下几个方面：

1)利用关系数据库成熟的管理技术能够对知识库中的各种知识进行集中管理,可以方便地对这此知识进行增加、删除、修改、浏览等操作,增强了知识对技术人员的透明度,并极大地简化了系统设计和维护人员对已有知识的访问过程和管理、维护难度。

2)利用 ODBC 技术可以在多种编程环境(例如,VC,VB 等)中方便地实现对各种数据库系统的访问,使不同编程语言在数据库的基础上实现了统一,从而在专家系统开发过程中可以根据需要灵活地选择开发语言以降低开发难度、提高开发效率,例如用 VB 来实现专家系统中经常要求的计算功能,用 DELPHI 来实现对数据库的操作。

3)目前针对常用关系数据库系统(如 ORACLE)已推出许多专用开发上具(如 DELPHI),利用它们可以力便快捷地开发出很友好的图形用户界面(GUI),便于用户使用,从而降低了对用户计算机操作能力的要求。

2 基于关系数据库技术的知识库

2.1 设计知识库结构需要注意的问题

(1)认知的一致性,需要能够足够准确的表示出所获得的专家知识。

(2)知识的灵活性,由于获取知识的多变性,要求能够很容易的添加、删除和管理知识条目。

(3)可移动性,随着网络技术的发展,需要知识库具有平台无关性,能够很方便的将知识库从一个操作平台移到另一个。

(4)充分推理,有能力以所设计的推理方法来充分利用存储的知识。

(5)有效推理,有能力把附加的信息结合到结构中去,而这些附加的信息可用于把推理机的重点放到最有希望的方向上去。

2.2 知识的表示

专家系统的知识库系统中,一般采用所表示体系是用“概念—事实—规则”所表示的三级知识体系。一般的,知识库应包括概念、事实与规则,由于概念一般均包含在事实内,因此,实际上知识库中应包括事实与规则两个部分。

规则是由前提和结论构成,前提是一系列事实的连接所构成的假设,结论则由几个事实组成,为了避免规则解释的二义性和简化数据库的设计,可以定义规则的前提只能是合取运算,如果遇到析取运算则首先将它们转化为多条规则。例如:

IF A OR B THEN C => IF A THEN C 和 IF B THEN C

那么对于规则就可以表示成为前提和结论的二元关系,如果考虑置信度的话就是前提、结论和置信度的三元关系。这里,为了方便讨论这里先只考虑二元关系。因此,知识库的存贮结构就需要满足以下几个要求:可以方便的表示事实和事实之间的关系;存贮的知识可以很方便的被利用。

2.3 基于数据库的知识库表示

通过以上讨论,就可以用关系数据库的形式来存贮知识,在所设计的知识库表示方法中需要 3 张二维表。

(1)事实表(facts):包含事实序号(facts_id);事实的描述(facts_des)两个字段,其中事实序号为主关键字。

(2)规则表(rules):包含序号(id);规则号(rules_id);前提号(pre_id)三个字段,其中序号为主关键字。

(3)结论表(results):包含规则号(rules_id);结论的描述(results_des)两个字段,其中规则号为主关键字。

下面来简单说明一下访问知识库的控制流程:

假设现输入一系列事实,首先将这些事实分解成为只含有合取运算的形式,然后将分解后的事实与数据库中事实表中的事实相对应,找出其相应的事实序号,然后和规则表中的每一条规则的前提号相比较,由于每条规则可能有多个前提,这样就要求在进行比较前先要对规则表中存放的各项记录按规则号进行分组,将规则号相同的分为一组,然后才能将前提和其相应的规则匹配。找到对应的规则号后再从结论表中按规则号得出结论。最后将得出的结论与事实表中的事实比较,若事实表中存在有相同的,说明得出的结论不是最终结论,则需要

再进行一轮推理,直到得出的结论与事实表中的事实没有相同即停止。这时得到的结论就为最终的推理结论。

2.4 正向推理机的程序实现

```

DataTable fact=dm.SelectDataSet("select * from facts").Tables[0];
int Num=1;
string content=rule.Text.ToString();
do
{
int index=-1; int resultId=-1; string RESULT=null;
content=content.Replace(" ", "@", "");
string sql1="select facts_id from facts where facts_des in ('"+content+"') order by facts_id";
DataTable factId=dm.SelectDataSet(sql1).Tables[0];
DataTable Result=dm.SelectDataSet("select * from results").Tables[0];
int count=Result.Rows.Count;
for(int i=0;((i<Result.Rows.Count)&&(resultId== -1));i++)
{
int m=Convert.ToInt32(Result.Rows[i][0]); int num=0;
string sql2="select rules_id,pre_id from rules where rules_id="+m+" group by rules_id,pre_id";
DataTable fenzu=dm.SelectDataSet(sql2).Tables[0];
int Count1=fenzu.Rows.Count;
int id1=-1,id2=-2;
if(factId.Rows.Count==fenzu.Rows.Count)
{
for(int j=0;j<Count1;j++)
{
id1=Convert.ToInt32(factId.Rows[j][0]);
id2=Convert.ToInt32(fenzu.Rows[j][1]);
if(id1==id2) then {num++;}
}
if(num==factId.Rows.Count)
{
resultId=Convert.ToInt32(fenzu.Rows[0][0]);
}
}
fenzu=null;
}
for(int n=0;n<Result.Rows.Count;n++)
{
if(resultId==Convert.ToInt32(Result.Rows[n][0]))
{
RESULT=Result.Rows[n][1].ToString();
}
}
if(RESULT==null&&Num==1)
{
MessageBox.Show("输入前提错误");
}
}

```

```
    }  
    result.Text=RESULT;    for(int s=0;s<fact.Rows.Count;s++)  
    {  
        if(RESULT==Convert.ToString(fact.Rows[s][1]))  
        {  
            index=Convert.ToInt32(fact.Rows[s][0]); break;  
        }  
    }  
}
```

3 总结

以上设计的简单的知识库支持多个前提,多条规则的正向推理,得到最终的一个结论,它还可以有更深更广的拓展。在现实生活中的结论往往不会是单一的一个事实,可能有多条事实以各种关系联系起来的总的结论,因此对与结论表的设计可以再进一步改进。同时,还可以对此系统增加一些内容,使其能更有效的,正确的完成反向推理等工作。在编程的实现上,还没有完全考虑到程序运行时的效率问题,在以后的研究中还可以做进一步的研究。

总的来看,利用关系数据库的特点可以快速地修改及浏览知识库,同时也方便专家对知识库中的知识进行检查及更新;同时将知识存贮在关系数据库中,可以很容易得实现知识库、推理机及应该程序的分离,这样有助于维护整个专家系统,并且不同的推理机可以共享同一个知识库。而且利用 SQL 语言的强大功能,可实现复杂的推理功能,从而能省略和简化代码的编制。

参考文献

- [1] Walczak Steven. Knowledge acquisition and knowledge representation with class[Z]. Expert Systems with Applications, 1998.
- [2] R R Hoffman. A survey of methods for eliciting the knowledge of experts[M]. SIGART Newsletter, 1998.
- [3] 徐洁磐, 马玉书, 范明. 知识库系统导论[M]. 北京: 科学出版社, 2000.
- [4] 刘星成, 汤庸译. 专家系统原理与编程[M]. 北京: 机械工业出版社, 2000.
- [5] 韩建超, 史忠植. 知识库系统研究[J]. 计算机科学, 1998.
- [6] 萨师煊, 王珊. 数据库系统概论[M]. 北京: 高等教育出版社, 2002

The Design of KBS Based on RDB

HUANG Xiao-jing, HUANG Yan

(Faculty of Mathematics and Computer Science, Hubei University, Wuhan Hubei 430073, China)

Abstract: In this paper, we combine the relation database and knowledge base with the features of their own, and knowledge is stored into relation database. After studying the requirements of KBS and the compositions of knowledge, this paper gives out a simple and convenient structure of KBS, which based on RDB. We also explain that how to use SQL for the forward reasoning of database with the reference of reasoning mechanism. Finally, an example is given as well.

Keywords: Knowledge Bases; Relational Data Base; Information representation