

EXPERT
SYSTEM

实用专家系统指南

■ 谭东风 编译

■ 哈尔滨工业大学出版社

封面设计 陆荣斌

ISBN 7-81024-142-7

TP·27 定价：4.20元

实用专家系统指南

译东风 编译

国防科大出版社

内 容 简 介

本书根据美国学者 Paul Siegel 所著“Expert Systems (A Non-Programmer's Guide to Development and Applications)”编译而成。书中围绕不懂计算机编程的用户怎样建立自己的实用专家系统这一主题，详细介绍了专家系统开发的全过程——确定专家系统领域、开发表示知识的词汇、建立知识库和测试知识库四个阶段。全书采用通俗易懂的语言，辅以丰富的图示，深入浅出地讲述了重要的概念、方法和实用技巧。贯穿全书的专家系统实例——交通事故原型为读者全面理解和掌握书中内容提供了帮助，同时也为读者实践所学知识提供了可供借鉴的范本。

本书可供具有中等以上文化水平的读者阅读，或供大专院校的师生使用，亦可作大专院校有关专业的教材或教学参考书。

实用专家系统指南

谭东凤 编译

责任编辑 何 晋

责任校对 钟 平

国防科技大学出版社出版发行

新华书店总店科技发行所经销

国防科技大学印刷厂印装

*

开本：787×1092 1/32 印张：10 26/32 字数：250千

1991年5月第1版第1次印刷 印数：1—4 200册

ISBN 7-81024-142-7
TP·27 定价：4.20元

目 录

引 言

第一章 什么是专家系统?

1.1 存贮在磁盘上的专家经验.....	(5)
1.2 什么是知识?	(6)
1.3 知识的表示.....	(7)
1.4 经验知识.....	(11)
1.5 计算机推理.....	(13)
1.6 人机对话.....	(17)
1.7 不确定性.....	(19)
1.8 推理的解释.....	(19)
1.9 专家系统的主要特征.....	(20)

第二章 专家系统在经营管理领域的应用

2.1 问题的类型.....	(22)
2.2 经营管理中的分析与综合任务.....	(27)
2.3 执行层的应用.....	(27)
2.4 控制层的应用.....	(30)
2.5 计划层的应用.....	(31)
2.6 专家系统的应用领域.....	(32)
2.7 复合领域的应用.....	(35)

第三章 图形化规则语言

3.1 上下文树.....	(38)
3.2 事实模版.....	(40)
3.3 规则模版.....	(43)

第四章 开发专家系统的过程

4.1 开发计划.....	(47)
4.2 阶段Ⅰ：确定专家系统领域.....	(49)
4.3 阶段Ⅱ：开发表示知识的词汇.....	(51)
4.4 阶段Ⅲ：建立知识库.....	(54)
4.5 阶段Ⅳ：测试知识库.....	(56)

第五章 阶段Ⅰ：确定专家系统的应用领域

5.1 示范实例.....	(59)
5.2 分割应用领域.....	(60)
5.3 确定推理流程.....	(61)
5.4 合成人机对话.....	(62)
5.5 重新组织或构造.....	(65)
5.6 确定性能指标.....	(68)

第六章 阶段Ⅱ：开发表示知识的词汇

6.1 确定属性.....	(70)
6.2 定义属性的值.....	(75)
6.3 编写检验词汇的规则.....	(79)

第七章 专家系统的推理过程

7.1 逻辑的模拟.....	(82)
7.2 正向推理.....	(87)
7.3 反向推理.....	(88)
7.4 前件规则与后件规则.....	(91)
7.5 元规则.....	(91)
7.6 不确定性.....	(92)

第八章 专家系统开发工具——电脑顾问

8.1 概述.....	(96)
8.2 语法.....	(98)
8.3 输入知识库元素.....	(106)
8.4 人机对话.....	(108)
8.5 调试.....	(114)

第九章 阶段Ⅲ：建立知识库	
9.1 准备书面形式的规则库.....	(120)
9.2 输入知识库元素.....	(126)
9.3 检查知识库的逻辑正确性.....	(128)
9.4 检查可信因子.....	(137)
第十章 阶段Ⅳ：测试知识库	
10.1 准备一套测试案例	(140)
10.2 评价知识库	(142)
10.3 改善人机对话	(148)
10.4 “开门”评议	(159)
第十一章 开发专家系统：一条自我深造的途径	
11.1 专家系统开发计划	(161)
11.2 经营管理专家系统领域中的知识	(168)
11.3 专家系统开发工具	(173)
11.4 严谨、求实的工作作风	(180)
11.5 开发自我	(184)
附录A “电脑顾问”的命令 —— ARL形式	
A.1 系统定义的元素	(186)
A.2 开发者定义的元素	(189)
A.3 命令	(194)
A.4 实例	(197)
A.5 从ARL到自然语言的翻译	(200)
附录B “电脑顾问”的命令 —— LISP形式	
B.1 系统定义的元素	(204)
B.2 开发者定义的元素	(207)
B.3 命令	(207)
B.4 实例	(208)
B.5 用户定义的函数	(213)
附录C 交通事故原型专家系统的规则库	(215)
附录D 交通事故原型专家系统的规则和参数（属性）清单.....	(239)

附录E 可在个人计算机上运行的专家系统开发工具

E.1 中档产品	(318)
E.2 低档产品	(320)
E.3 高档产品	(321)

参考文献

索 引

引　　言

如果你是一位在经营管理方面经验丰富、颇有建树的人士，你希望有效地指导或培训那些刚刚从事经营活动的人们；如果你是一位整日忙于经营活动，希望有一个具备相当经验的助手随时帮助你的经理人员；如果你是一位对专家系统略有所闻，想进一步了解并尝试建立自己的专家系统的人；如果你是一位想对专家系统进行系统研究，但又被专业化的文献所困扰的学者；或者你是出于某种目的想了解专家系统的人，那么，阅读本书也许是有所裨益的。

由于开发专家系统需要使用计算机，因此你可能以为开发专家系统必须在一个大型研究机构里才能做到，其实不然。今天，任何拥有个人计算机的人都可以开发出自己的专家系统。不仅如此，由于专家系统具有集组织中各种专业知识于一体的机制，凭借这种崭新的工具，小企业家也有可能与大的跨国联合企业展开有效的竞争。

开发专家系统不一定必须具有程序设计的经历和经验，而只需熟悉个人计算机的使用就可以了。

本书有两个目的。首先，帮助你判断专家系统对你是否有益；其次，向你介绍怎样建立专家系统。这两个目标是相辅相成的，你很快就会发现，不了解专家系统的开发过程是难于判断开发专家系统的有关问题的。为此，我们将在本书之中为你解答以下问题：

“什么是专家系统？”

“你需要专家系统吗？”

“专家系统能为你做些什么？”

以及

“建立一个专家系统你应该做些什么？”

本书将介绍专家系统适用的经营管理领域，并说明专家系统为什么适用于经营管理领域。然后，还要介绍在微型计算机IBM PC及其兼容机上开发小型专家系统的全过程，这个过程由四个开发阶段组成。

本书在叙述上着力表述清晰明了的原理和简单实用的方法，而不象“菜谱”那样详尽地介绍每一道菜肴的配料细节和烹调过程。在本书中我们也介绍了利用一种称之为专家系统开发工具的软件，即德克萨斯仪器公司出品的“电脑顾问”，来建立一个专家系统的整个过程及其部分细节。我们之所以这样做，既不是想使你陷入细节之中，也不是企图给你指出一条“唯一正确的道路”，而是为了使抽象的概念生动具体、深入浅出。虽然我们专门介绍了“电脑顾问”这个软件，但本书所述内容对于同类个人计算机专家系统开发软件来说，同样也是适用的。

建立一个专家系统不是一件轻而易举的事情，但是只要你抱着实事求是的态度，用科学合理的方法，同时对你所从事的专业工作非常娴熟，而且准备集中才智、全力以赴的话，那么你一定能够成功。建立一个小型专家系统（大约含三百条规则）所需的时间根据问题领域的复杂程度而有所不同，一般来讲，一项工程计划大约需要六个月左右。

在着手进行正式系统的开发之前，应该先从所开发的领域中选择一个小而有代表性的原型进行试验。根据从原型开发中

获得的经验，你可以检验开发方法是否正确，同时也可以更准确地估计整个工程将持续的时间。

多数人认为开发专家系统需要一位知识工程师的帮助。知识工程师是从专家那里获得知识，并利用这些知识建造专家系统的行家。然而，知识工程师的三分之二时间花费在发掘专家系统使用的词汇和规则上面，为此，他必须与专家进行大量的讨论才能获得这些知识。这就意味着你必须为工程的进展付出大量宝贵的时间。

你需要知识工程师的帮助吗？如果是建立一个小系统，也许并不需要。当问题领域很大很复杂时，你才离不开知识工程师。此外，在开发大型专家系统时，你还可以通过建立工程小组的形式来获得同事们的帮助。

本书最好分三遍来阅读。第一遍，通过阅读可以获得一个对专家系统开发过程的大致印象，然后便可以判断有没有必要开发专家系统，若有必要开发，是否需要知识工程师和工程小组的帮助。在读第二遍时，要更仔细地了解开发专家系统的过 程和方法，并应用这些知识来指导原型系统的开发。原型系统完成之后，要进行全面的评价。在开始正式系统的开发之前，再进行第三遍阅读。

在阅读过程中，你也许并不认为自己对业务是很精通的，但你只要仔细地阅读本书，并亲手建立一个专家系统，这对你提高业务水平肯定是大有裨益的。通过开发专家系统并且用它来解决实际问题，你便不知不觉地开发了自己的知识。也许，有朝一日你也就成为专家了。

第一章 什么是专家系统？

你是专家吗？当然是。任何人都会在某一方面有所造诣。如果你是一位企业家，你就一定知道怎样建立一个组织。如果你是一位经理，你就是一批人的领导者。如果你是一位设计师，你可以创造新的想法和产品。如果你是故障检修员，你可以很快地确定问题的原因，并能及时地纠正或者修复它。如果你是研究人员，你一定拥有发掘新知识的诀窍。

如果你是一位擅长分析问题的人，你会把复杂的问题分解成较小、较容易解决的小问题，进而将它们解决。如果你是一名专业顾问，由于你对自己所从事的专业领域有着透彻的了解，所以你能够解答有关这个专业范围的问题，这个领域可能是财政、税务、法律、宗教、政治、教育、经济或其它专业领域。

难道你不愿意整理出自己的专业特长和经验，以便更好地利用或用来与别人进行交流吗？有些人把他们的专长和经验整理成书，但是，建立专家系统可能是一种更好的办法。你可以用规则的形式将你的知识输入到专家系统中，这个专家系统就可以同遇到问题的用户进行“交谈”，应用存储其中的规则进行推理，最后向用户提供咨询，甚至能向用户解释自己的推理方法和过程。这里的用户可以是任何人——你的同事、部下、学生、供货商以及现实的和潜在的顾客等等。

1.1 存贮在磁盘上的专家经验

什么是专家系统？简单地讲，专家系统就是存贮在磁盘上的专家经验。通常，先由专家系统的开发者把人类专家从多年经验中积累起来的核心知识存贮到磁盘上，以后，用户就可以通过个人计算机获得这些知识。他们只需先回答几个由专家系统提出的问题，然后就可得到专家系统对专业领域中问题的咨询建议。专家系统正是应用这些事先存贮在计算机中的知识进行推理并提供咨询建议的。

开发专家系统的设备是一种称作“专家系统开发工具”的奇妙软件。图 1-1 展示了专家系统开发工具的功能：它引导专家系统的开发者，也就是你，输入自己的知识并建立知识库，即图中的“黑箱”。这个黑箱连同专家系统开发工具中的推理机一块复制到另一个软磁盘上，这就形成了一个完整的专家系

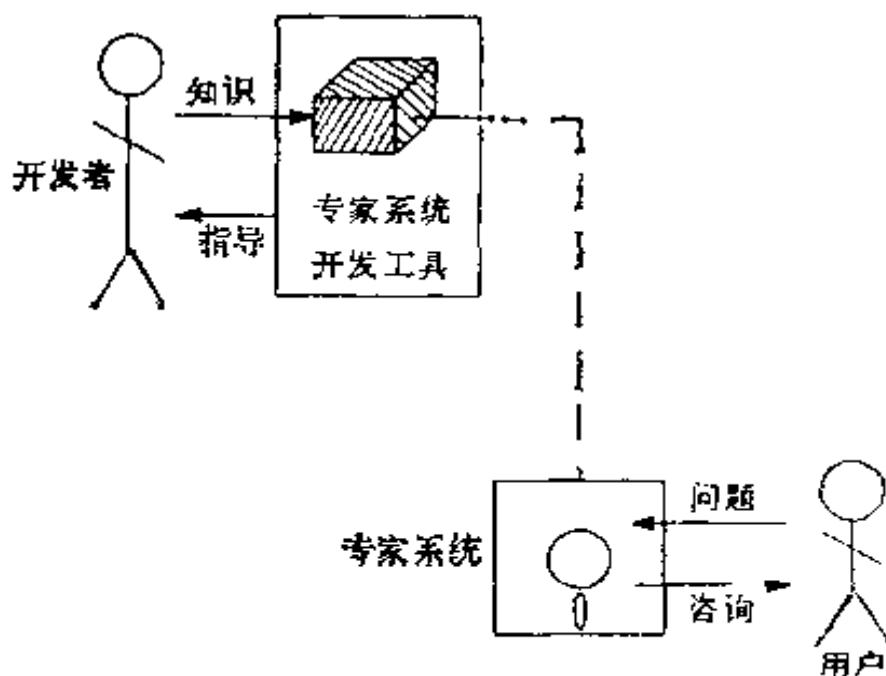


图 1-1 开发专家系统

统。这个专家系统利用以磁盘记录形式贮存的专业知识和经验来提供咨询、解决用户的问题。

下面，我们将一步一步地向你讲解，怎样把你的知识转换成象“电脑顾问”这样的专家系统开发工具能够理解和接受的形式。当然，我们也将解释“电脑顾问”或其它专家系统开发工具是怎样指导开发者输入知识以及怎样指导开发者开发自己的专家系统的。

1.2 什么是知识？

让我们用与数据和信息比较的方法来说明知识。在人类制定决策的活动中，这三者之间组成了一个以数据为基础、以知识为顶层的层次结构。

数据用来表示事实。例如，约翰身高1.70米，玛莉有双大眼睛。

信息则是数据的“浓缩”。数据经过组合、概括、校对、比较、分类、相关分析以及其它处理过程，从而变成更方便人们制定决策的形式。例如，成本-效益报表就是一种信息，因为它是由数据经过“浓缩”后，成为有助于人们判断一个企业经营水平的特有形式。

最顶端的是知识，它是人类制定决策的基础。人们能根据不同的成本-效益报表确定经营得最好的企业，而做这件事本身是需要知识的。

图1-2说明了数据、信息和知识处理与人类决策之间的关系。若计算机只做数据处理，那么人就得把数据加工成信息，然后据此做知识处理以便形成决策。如果计算机也做信息处理（包括数据处理），则人们只要进行知识处理后就能形成决策。当计算机能够进行知识处理时（包括数据处理和信息处理），留

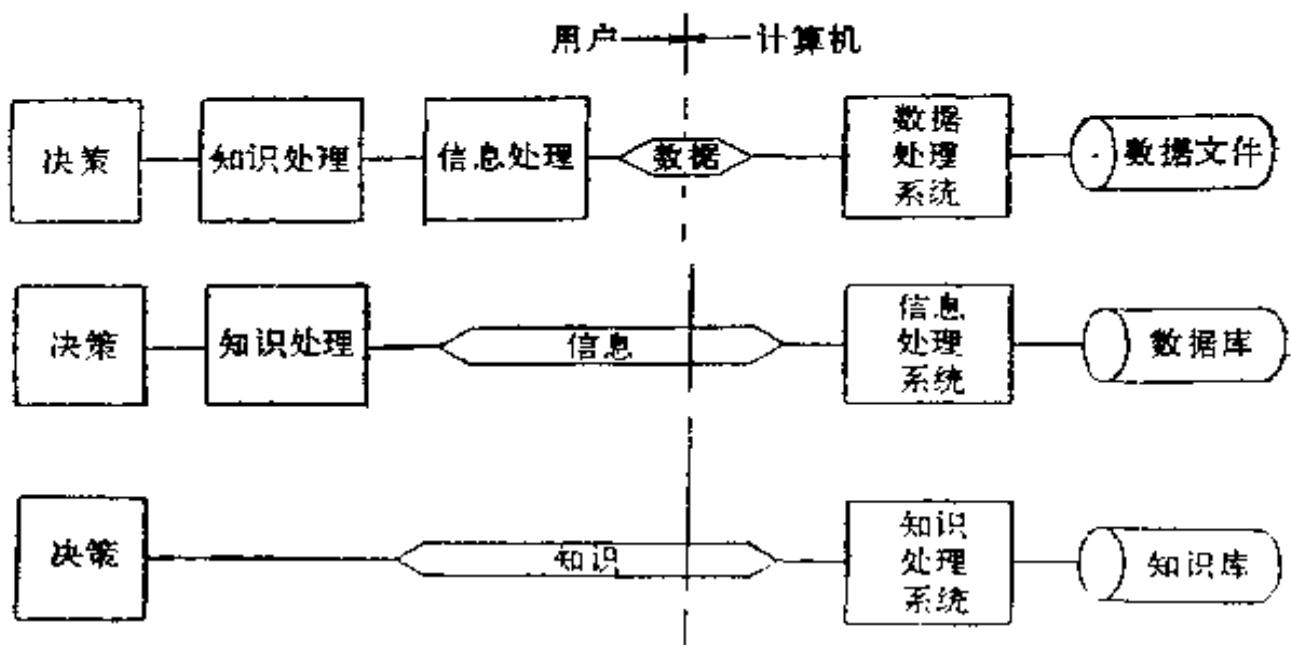


图 1-2 数据、信息和知识处理

给人们做的，除了把知识处理的结果应用于具体的任务以外，就相对少了。

1.3 知识的表示

尽管有上一节的介绍，知识这个词对我们来说仍是一个模糊的概念，你只是从直觉上知道了它的意思。但是知识应该怎样表示才能由计算机进行处理呢？在计算机科学界，这是一个多年来始终热门的研究课题。目前的研究主要集中在知识的描述、编程以及相应的数学技巧方面。在众多目前正在探索的途径中，语义网络、框架和规则是三种最能为广大非程序员用户接受的技术。

1.3.1 语义网络

语义网络能够反映不同个体之间的联系。一个简单的描述动物界内部联系的语义网络如图 1-3 所示，该图说明哺乳动物“是一种”动物，同样，食肉动物和鸟类也“是一种”动物；该图还说明哺乳动物的“表皮”类型是毛发，而它们的一种“活动”是产奶；类似地，该图还说明鸟类的表皮类型和活动特征，食肉动物的外貌特征、附属器官和活动特征。

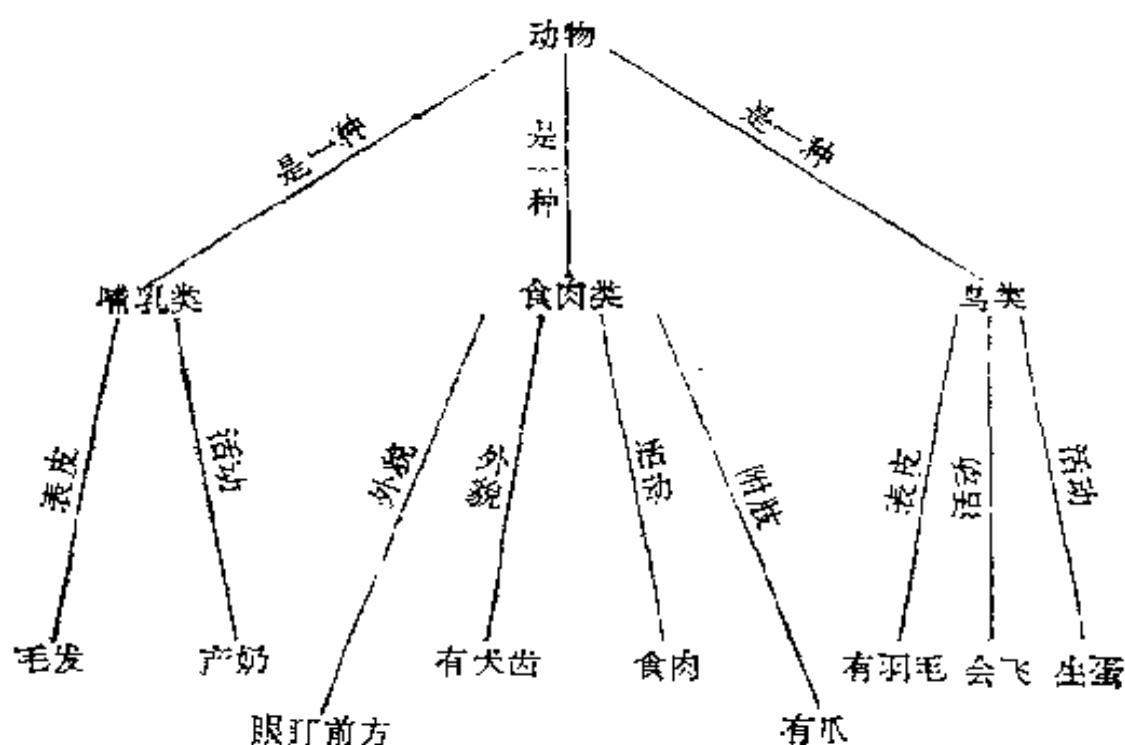


图 1-3 语义网络

在图示中，我们用线段来表示联系两个对象的“关系”，通常把关系标注在连线上。下面是三个关系的例子：

- A-KIND-OF (哺乳动物, 动物) —— 表示“哺乳动物是一种动物”；
- SKIN-COVER (毛发, 哺乳动物) —— 表示“哺乳动物的表皮是毛发”；

ACTIVITY (产奶, 哺乳动物) —— 表示“产奶是哺乳动物的一种活动”。

语义网络所描述的系统、概念、问题和关系都能由计算机进行处理。

1.3.2 框 架

另一种表示知识的方法是框架，其思想来源于著名的人工智能专家马翁·明斯基的人类怎样思维的理论。明斯基曾经说过“当一个人遇到一种新情况（或对他目前认识的问题有较大改变）时，他会从其记忆中选择一个称之为框架的实在结构，并根据需要改变这个记忆框架中的一些细节以适应变化了的实际问题。”

明斯基又说，“框架是一种数据结构，它可以用来表示那些已有陈规的情况，如身在一个起居室里或去参加一个孩子的生日晚会。每个框架附带着各种信息，有怎样使用框架的信息，有期望下一时刻发生什么事件的信息，有当这些期望都不成立时应该做什么的信息等等。”

在图 1-3 中用语义网络表示的数据可用图 1-4 中的三个框

哺乳类		食肉类		鸟类	
是一种:	动物	是一种:	动 物	是一种:	动 物
表皮:	毛发	外貌:	眼盯前方 有大齿	表皮:	羽 毛
活动:	产奶	附肢:	有 爪	活动:	会 飞
*	*	* 活动:	食 肉	*	生 蛋

*：除描述型数据之外，也可以包含过程型信息

图 1-4 框架

架来表示。第一个框架是用来表示哺乳动物的，其中顶端的关系表示哺乳动物是一种动物，这个关系起到指向动物框架的作用。表皮和活动关系也在哺乳动物框架中说明。食肉动物框架和鸟类动物框架之间的关系与语义网络中的类似。

与语义网络相比，框架有两个最大的优点。其一，可以分割复杂的领域；其二，除能存贮描述性数据外，还能存贮过程性知识。

1.3.3 规则

规则是一种最简便的知识表示法。用哺乳动物框架说明的主要知识成分可表示成规则：

如果 动物有毛发
且 动物产奶
则 该动物是哺乳动物

规则是目前用得最多知识表示方法，也是我们在本书的后续部分中将集中论述的主题。规则之所以得到如此广泛的应用，是由于它具有以下特点：

- 简洁性
- 模块性
- 规模恰当
- 既具有描述性又具有过程性

简洁性 知识的表示、理解和处理简单明了。

模块性 每条规则独立表示一个意思，对它的任何变化或修改都不会影响其它规则。

规模恰当 语义网络中的关系显得太琐碎，而框架又显得太粗旷。规则却能够做到规模恰当。例如，鸟类框架中的数据可分属两条规则，而不是一条：

- 1 - 如果 动物有羽毛
 则 该动物是鸟
- 2 - 如果 动物会飞
 且 动物生蛋
 则 该动物是鸟

既具有描述性又具有过程性 到目前为止，已经介绍过的规则都是描述性的。然而，当你阅读到后面，你就会发现规则也具有过程性。

1.4 经验知识

我们常说的经验知识，是指我们思维时所遵循的准则。例如，“药补不如食补”，“小洞不补，大洞吃苦”等都是经验知识。

经验知识很容易转换成用更形式化的方法表示的规则，这种形式化工作是用计算机处理规则所必须经历的步骤。在图1-5中，有三条关于风险投资的规则。

一条规则通常由两部分组成，如果部分和则部分。一条规则可以表述一个简单的逻辑关系：

如果 某个条件为真
 则 某给定结论成立

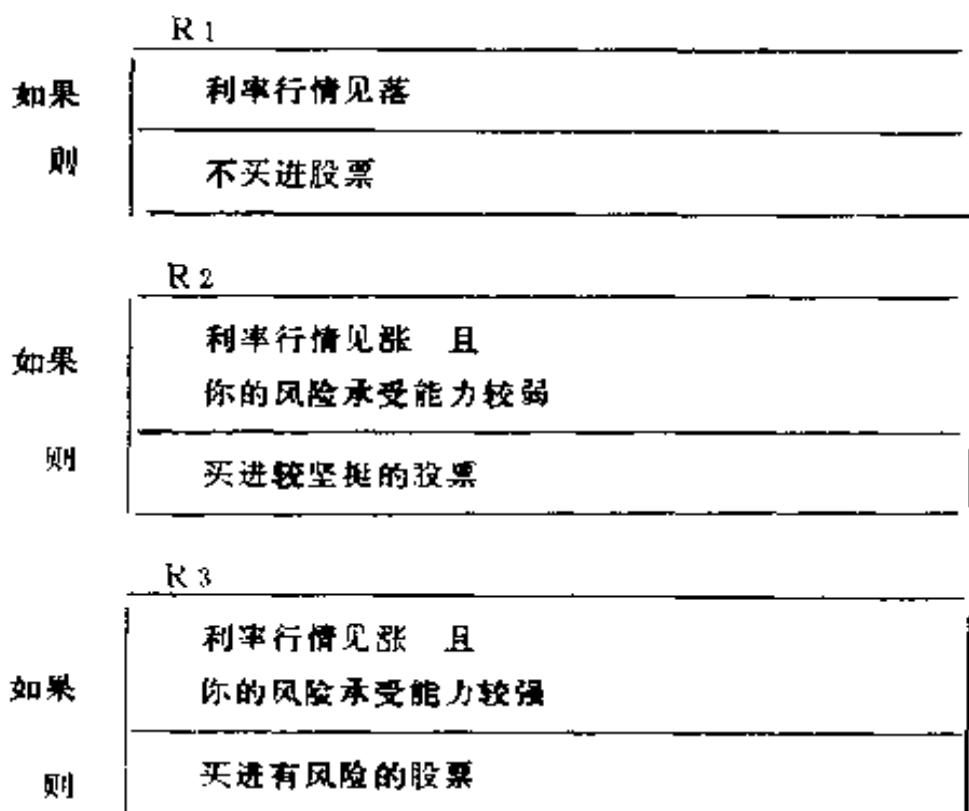


图 1-5 风险投资规则

譬如图 1-5 中的第一条规则 R 1，它表示如果子句“利率行情见落”为真，则结论“不买进股票”成立。规则 R 2 也类似，所不同的是它的如果部分含两个如果子句，它表示如果子句“利率行情见涨”且“你的风险承受能力较弱”都为真，则结论“买进一种较坚挺的股票”成立。规则 R 3 与规则 R 2 相似，所不同的是它表示如果利率行情见涨且你的风险承受能力较强，在这种情况下，你可以买进有风险的股票。

在图 1-6 中，我们给出了另外两条规则 R 4 和 R 5，它们与 R 1、R 2 和 R 3 的结构相似。然而，请注意 R 4 的则部分与 R 2 的第二个如果子句相同，R 5 的则部分与 R 3 的第二个如果子句相同。

通常，如果部分可以包含任意多个条件子句。同样，則部分也可由任意多个结论子句组成。然而，在实际应用中，为了

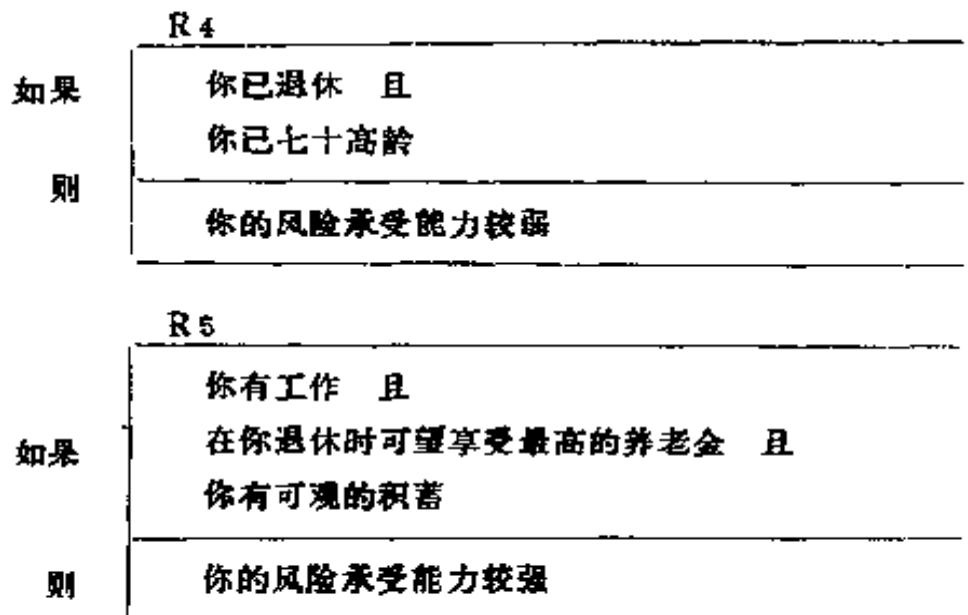


图 1-6 另外两条风险投资规则

保持逻辑描述的简洁，往往限制如果部分和则部分中子句的数目。

每条规则都是一个精练的知识模块，可以对它进行修改或替换而不会影响其它规则。规则是组成知识库的基本构件，你的经验可以用规则的形式存贮到知识库中去。建立在这种知识库之上的专家系统既能非常灵活地适应实际情况的变化，又能解决复杂的问题。

1.5 计算机推理

所谓计算机推理，就是计算机沿着知识库中的规则所经历的推理路径。若计算机从事实出发，沿着规则达到最终结论或目标，则称这个过程为正向推理或正向链接。若计算机由最终结论或目标出发沿着规则的引导向后达到已知事实，则这个过程称作反向推理或反向链接。

归纳推理是计算机能够进行的另外一种推理形式，即从大量的例子数据中抽象概括出有用的规律。归纳推理也是科学工

作者经常采用的一种思维形式，在证实了一组对已知现象的观察结果之后，科学家们往往要把这些观察结果概括成一条普遍适用的科学规律。计算机也能进行归纳推理，它先接受一组例子，然后形成一个能概括所有例子的一般性规则。

1.5.1 正向推理

正向推理的过程如图 1-7 所示，在这幅图中包含有五条用图形表示的规则。在这个图示中每条规则都经过简化，并用一个方框来表示，其中如果部分位于方框的上半部分，则部分位于其下半部分。从每个则部分中的结论子句引出的箭头，指向所有其条件子句与该结论子句相同的如果部分。因此，有从 R4 指向 R2 和 R5 指向 R3 的箭头。

在正向推理中（也称数据驱动的推理），计算机进行从已知事实跟踪到最终结论的逻辑推理过程。在本例中，已知事实是：

- 你有工作
- 你退休时可享受最高的养老金
- 你有可观的积蓄
- 利率行情见涨

由于规则 R5 的如果部分中的所有条件都为真，因而该规则的则部分——“风险承受能力强”——也为真。将这个结论应用到 R3 的如果部分，在那里另一条件——“利率行情见涨”——已为真，故规则 R3 被激活，使它的则部分——“买进有风险的股票”——也为真。

1.5.2 反向推理

在反向推理中，推理过程是从结论——“买进有风险的股

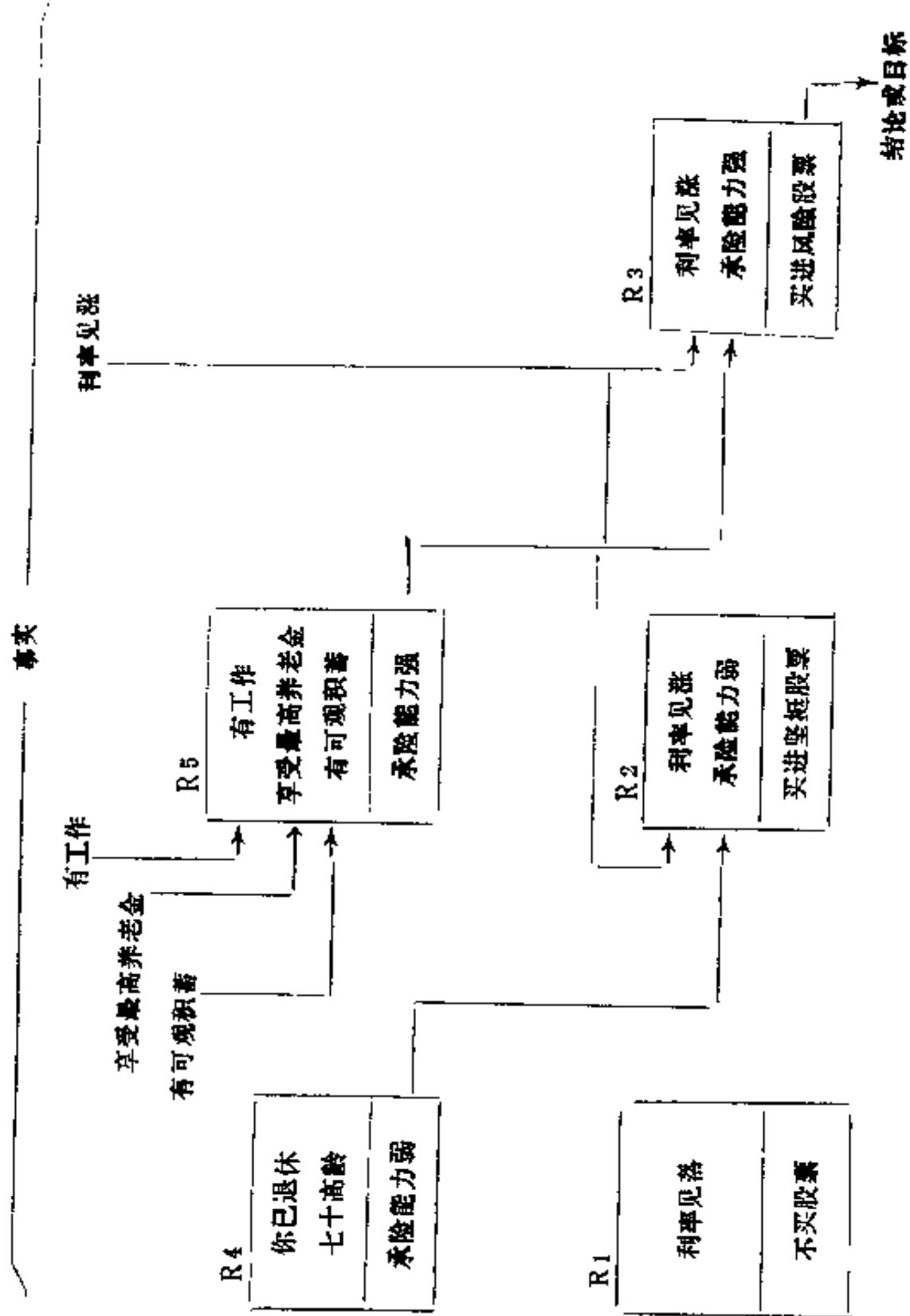


图 1-7 推理过程的图示

票”——退回到该结论所依赖的已知事实。请看图 1-7 中的规则 R3, R3 的如果部分的第一个条件——“利率行情见涨”——为真, 第二个条件依赖于规则 R5。在规则 R5 中, 全部条件都为真, 这样就找到了一条由结论到各已知事实的完整路径。依照这种方式计算机可做出判断: 你若买进有风险的股票, 你必须有承担风险的能力, 而且当时的利率行情正在上涨, 进一步你一定是有工作的, 而且在退休时可享受最高的养老金并有可观的积蓄。

1.5.3 归纳推理

为了理解计算机是怎样依据从观察得到的实例进行推理的, 不妨先看看图 1-8。在图 1-8 中, 有一组关于交通事故中,

观 察	车 速	司机的状态	生理损伤
1	50	清 醒	轻 微
2	55	半 醉	残 废
3	85	清 醒	残 废
4	65	大 醉	残 废
5	75	半 醉	死 亡
6	85	大 醉	死 亡
7	80	大 醉	死 亡
8	45	半 醉	无 事
9	70	半 醉	残 废

如果	车速超过 70 且
	司机喝醉酒
则	司机的生理损伤是死亡

图 1-8 归纳的例子

汽车速度、司机的状态以及司机生理损伤的统计结果。对这些统计数据进行细致的观察可概括出下述规则：

如果 汽车的速度大于70（公里/小时），
且 司机喝醉了酒，
则 司机遭受的生理损伤是死亡。

这个规则的真实性不是必然的，其必然性可用更多的统计结果以更准确的形式进行验证。

1.6 人机对话

当你请一位专家来为你做某项咨询时，你一定有这样的经历，这位专家首先总是详细地问这问那，以便确定你的问题和目的是什么。他一般很少只根据用户对问题的初步陈述就下定论。他会问一系列问题，也许还要做一些试验或测验，以便确认实质的（而不是表面的）问题所在，发现那些使问题难于解决的特殊障碍，找到解决问题的正确途径等等。医生、律师或会计师都是这样做的，任何一个提供咨询意见的人也是这样做的。

当然，专家系统也应该这样工作才对。下面，让我们来看一个使用反向推理方法工作的专家系统，这种专家系统可以从结论反推出一些规则，这些规则的如果部分是由一些无需其它规则说明，而且可以直接证实的事实组成的。为了证实这些如果子句是否为真，专家系统通过提问的形式，请求用户告知哪些如果子句是为真的，哪些如果子句是不为真的。下面这段文字描述了图1-7所示的推理过程。

假设系统从得出买进风险股票结论的规则 R3 开始推理。

为了判断 R3 的如果部分是否成立，它向用户提问：“今后六个月利率行情怎样？”

如果用户回答：“下降”或“平稳”，则计算机知道 R3 不可用。不过，若此时用户的回答是“上升”，为了确定下面这个条件——“承担风险能力强”——计算机向后考查规则 R5，并就下述几个问题向用户提出询问，以确定 R5 的如果部分是否为真：

“你已退休，还是正在工作？”用户回答：“工作”，

“在你退休时，是否有望享受最高的养老金？”用户回答：“是的”。

“你有可观的积蓄吗？”回答：“是的”。

系统现在已经获得了足够多的信息，能够做出买进风险股票的咨询决定了。

图 1-9 记录的就是上述对话的过程。

问 题	回 答
今后六个月利率行情怎样？	上 升
你已退休，还是正在工作？	工 作
在你退休时，是否有望享受最高的养老金？	是 的
你是否有可观的积蓄？	是 的

图 1-9 人机对话的例子

专家系统提出的所有问题都是按照你的事先安排进行的。有时，例如在医疗诊断中，专家系统也许提议用户做些必要的

试验以便确认有关的事实，然而，所有这些“建议”也是按照你事先编排做出的。

向用户询问不是专家系统收集有关事实数据的唯一方法。它还可以从数据库，电子表格和统计软件包直接检索数据。

1.7 不确定性

输入到专家系统的规则都是经验规则。作为你的经验知识的结晶，你很了解它们的作用。例如，其中有些总有效果，你对它们有百分之百的坚信。而对另一些规则，你可能只有百分之八十，百分之六十甚至更少的信任。就拿前一个例子中的 R₃ 来讲，你也许会说你对自己所做的咨询决定“买进风险股票”的有效性只有百分之八十的把握。

专家的经验尚且如此，那么用户的数据也就更不例外。他对所述事实并不是总有十全的把握。例如，有人只有百分之八十的把握在他退休时能够享受到最高的养老金。

有的专家系统通过给规则带一个确定性因子 (certainty factor)，并允许用户给每个事实一个可信程度的方法来处理上述不确定性。在下面的介绍中，我们将看到专家系统可以综合这些可信因子，以便以一定的确定性程度给出咨询建议。

尽管我们在本书中主要讨论可信因子 (confidence factors)，有的专家系统开发工具可能出于同样的目的，但却采用诸如概率统计等更为严格的技术。

1.8 推理的解释

一位专家除了能处理实用知识，能分析推理，能用这些知识与人们进行“交谈”之外，还应该对得出结论的推理过程作出解释。理所当然，专家系统也应该如此。

当然，专家系统不可能按人类专家采用的方式解释自己的推理，但它可以回答“为什么”和“怎样推”两类用户提问。

如果用户想问专家系统为什么要要求他提供某项数据，专家系统就应该解释因为所要求的数据可以导出某个结论。例如，若用户想知道专家系统为什么问“你有工作吗？”这个问题，专家系统就会显示规则 R5 来做为回答，以解释有工作是规则 R5 的如果部分中的一个条件。

又如，用户还可以向专家系统询问它是怎样推得结论的，专家系统就会把推理过程中用到的规则顺序地列出来，以此来向用户解释它的推理过程。在上面这个例子中，如果用户问“（用户）应该买进风险股票”这个咨询意见是怎样推出来的，专家系统就可以用显示规则 R5 和 R3 来解释它的推理过程。

1.9 专家系统的主要特征

下面讨论基于规则的专家系统的主要特征，这种专家系统可以在类似于“电脑顾问”的专家系统开发工具的帮助下，无需做任何程序设计工作就能实现。

1.9.1 知识库

知识库是由规则和表示问题的数据组成的，其中规则包含表示条件的如果部分和给出结论的则部分。

1.9.2 独立的推理机

推理机是与知识库分离的。推理机一般由专家系统开发工具内部提供，对于“电脑顾问”来说主要是反向推理机。它可以作为专家系统的一个组成部分，在建立专家系统时，由专家系统开发工具自动提供。

1.9.3 人机对话

使专家系统激活合适规则的方法是与用户进行一段对话。

与规则产生的提问信息和给开发者的有关引导信息一样，为响应用户对提问的回答而产生的新的提问信息都必须事先存入专家系统。专家系统通过人机对话提供咨询，专家系统也可直接与其它软件包通信。

1.9.4 推理的解释

专家系统可以对试图达到或已推得某个结论的推理过程进行解释，前者通过解答用户的“为什么”提问来实现，后者由对用户的“怎样推”提问进行解答来提供。

1.9.5 不确定性

专家系统通过使用确定性因子或其它方法来处理有关事实以及与这些事实相关的规则的不确定性。

第二章 专家系统在经营 管理领域的应用

作为一种智能倍增器，基于规则的专家系统可以成为一种能够改变经营管理领域面貌的先进工具。与以往的任何一项发明相比，专家系统更能够有效地改变智力工作的性质。它能改变管理工作的性质，研究、工程和生产活动的性质，销售和分配的性质，软件的性质，职业活动的性质以及教育工作的性质等等。

专家系统是一种新型的问题求解工具。如同其它任何工具一样，它并不适用于所有知识领域。然而，值得庆幸的是，对于大多数有着最为困难的问题需要解决的经营管理领域来说，专家系统是完全适用的。目前解决这些问题尚没有现成有效的方法，而需要大量从实际经验中积累起来的问题求解知识。

2.1 问题的类型

解决“困难”问题的一个最好办法是先把它分解成若干个相对简单的小问题，然后再逐个解决这些简单的小问题，这种解决问题的策略称之为“分而治之”或“各个击破”。如果这些较简单的小问题解决起来仍不够方便的话，则可以把它们进一步分解，直到你认为能够比较容易地解决时为止。最后，按照这样的分解路线返回来解决最初的“困难”问题。

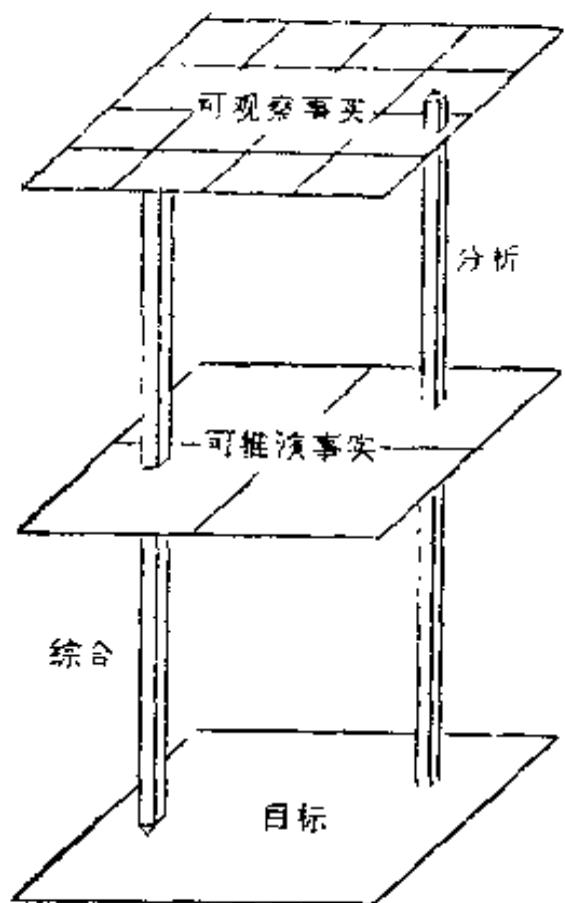


图 2-1 分析与综合

图 2-1 给出了这种分解的三个有代表性的层次，其中处在顶层的是细节丰富的可观察事实，中层是较一般化的可推演事实，处在底层的是最一般化的目标。除此以外，还有许多其它的术语可用来描述这三个层次，它们是：

子子问题—子问题—问题

- 详细的原因—粗略的原因—结果
- 部分—部件—系统
- 步骤—例程—过程

为了找到问题的答案，而把一个“困难”问题分解成若干部分的过程称为“分析”，而与此相反的过程则称为“综合”。

2.1.1 分析

在分析过程中，你的工作次序为：目标—可推演事实—可观察事实。分析通常包含如下内容：寻找组成整体的部件，它们可以是想法、原理、原因、对象或例程等。

分析有多种形式。形式之一称作“分类”它是一个给被分析对象按照某种条件“贴上”相应标签的过程。例如，在邮政部门的信件分检中，邮政工人按照寄往地址将各种信件分别投入不同的柜子，这就是一个典型的分类工作。生物学家往往按照一定的特征来辨别动物，他们采用的是动物分类规则，如图 2-2 就是一条这样的规则。

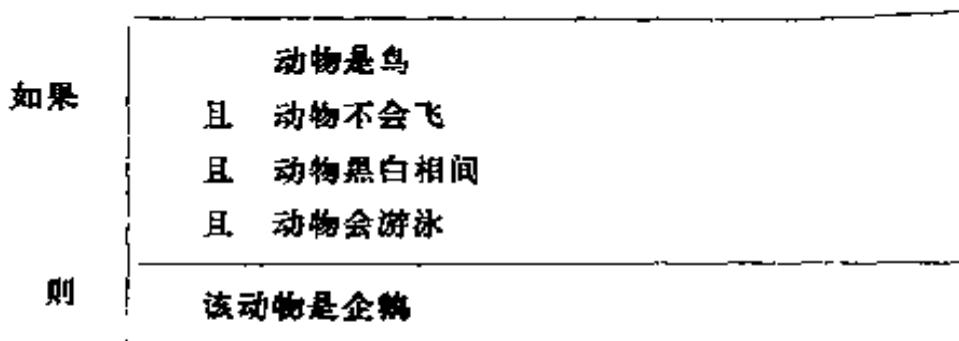


图 2-2 动物分类规则

另一种分析形式是“搜索”。在搜索时，要求找到满足给定条件的相应用对象。石油勘探人员为了确定钻井位置需要进行搜索。为了完成这项任务，他可以使用专门帮助地质勘探的专

家系统PROSPECTOR(勘探者)。在PROSPECTOR中贮存有表示矿藏模型的规则网络，它可以向钻探人员询问有关岩石、矿物和其它地质构造的年代、理化成分、岩石纹理以及规模等信息，然后根据其知识进行推理，并提供有关出油概率的钻井咨询。

分析的其它形式还有“测试”、“预测”和“诊断”。测试用来判断一个物体、部件或组织是否满足一组给定的准则。对于从阿斯匹林到拉链等大批量消费品的生产活动，这种测试是至关重要的。

与其它形式的分析不同，诊断主要解决因果关系问题。假设原因发生在某个小的部件中，而结果则体现于整个物体。在辅助诊断的专家系统中，最著名的例子是MYCIN，它是一个帮助医生诊断感染性疾病的研究原型。在向医生询问了一系列有关病人的一般健康情况，医学检查的结果以及其他问题之后，MYCIN应用类似于图2-3中的规则确定病人是否已被感染或带有引起感染的细菌以及其他生物，并向医生提供一张对抗感染有效的药物清单，而且推荐一种对病人最好的药物。

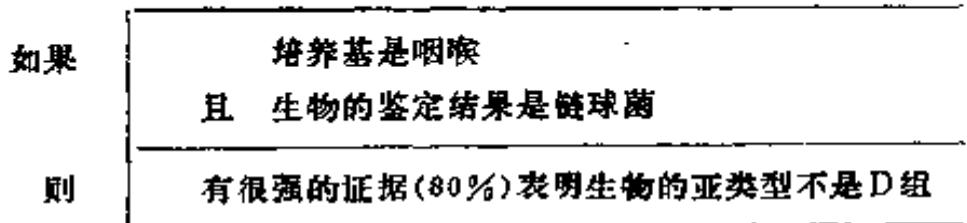


图 2-3 MYCIN的规则

2.1.2 综合

综合比分析困难得多。因为你得去发明、建立、设计甚至设想各种各样的想法、体系和机器。做综合时，应该遵循从上到下的次序开展你的工作：可观察事实—可推演事实—目

标。

“配置”是一项综合任务。配置就是为了达到某些设计准则而将各部件组织成一个完整系统的过程。过去，无线电爱好者将分离的元件凑到一起装配成自己的收音机、电视机甚至“高保真”音响系统。今天，有些个人计算机爱好者在用计算机组件来做同样的事情。

配置大、中型计算机系统需要大量的技术秘诀。例如，一个典型的VAX-11计算机系统是由90多个部件组成的，而这些部件又是从一个更大的部件集中挑选出来的，并且要求这些被选出来的部件放到一起后必须协调一致地工作。美国数据设备公司(DEC)开发了一个称为XCON/R1的专家系统，它可以帮助该公司的人员完成这项任务。图2-4介绍的是它所使用的一条规则。目前，这家公司的推销员们正在利用XCON/R1的改进型XSEL为顾客配置系统或为特殊的应用提供建议。

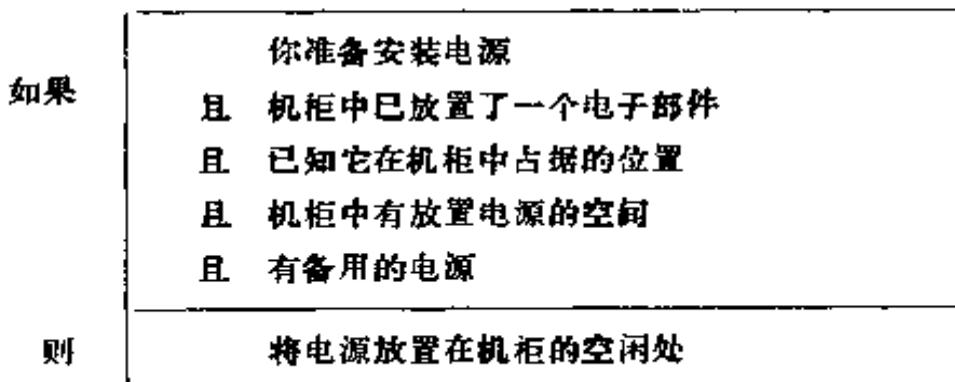


图2-4 XCON/R1的配置规则

“设计”是一种更为复杂的配置工作。配置主要是针对实际部件的；而设计则主要是针对概念的，如在设计通信系统时，常常使用诸如信号变换、放大、音频等概念。因此，可能要使用一些具有某种处理功能的小部件，如话筒、放大器、或扬声器等。但仅仅将这些部件堆砌在一起并不能构成一个通信

系统。可以相信建立一个设计型专家系统肯定比构造配置型专家系统困难得多。因此，我们暂时不准备尝试这类工作，除非它非常简单。

2.2 经营管理中的分析与综合任务

一个公司或企业是一个为了实现某个特定目标，而由人员、机器、材料和服务设施组成的系统。它是非常复杂的。现代企业从经营上看至少可分为三个层次。首先是执行层，它负责完成具体工作：研究、工程、生产、销售、采购和推销。其次是控制层，在这一层，管理人员协调进度、控制执行层进行的各项工作。最后是计划层，在这一层，由行政主管做出预测和计划并负责判断计划目标是否实现。

在上述三个层次中，都需要分析，没有分析的工具，现代企业就不可能有效率地运转。分析的工具对于企业与社会、经济的有机结合，对于诊断组织中的各种流向，对于帮助其职工更好地完成种类繁多的任务来说，都是非常必要的。

仅有分析，而没有相应的行动，不会给企业带来任何有益的效用。行动是一种综合任务。经过分析，你对过程或系统获得了清晰的了解，从而有可能设计一个系统或修复一个功能欠缺的系统。

在一个企业的所有层次上也同样需要综合，在执行层，需要综合来创新服务、产品、生产和销售的质量与技术；在控制层，用于控制整个系统；在计划层，用于开发具有企业特色的计划过程。

2.3 执行层的应用

在执行层有许多分析和综合任务可以从基于经验规则的专

采购:

电子器件的测试

机械部件的测试

生产:

机械部件的诊断

计算机故障的诊断

电子开关系统的诊断

生产线的咨询

机床刀具的选择

销售和推销:

市场分析

信用评估

财务:

贷款分析和审批

税务咨询

投资咨询

选择保险政策和比率

设备出租咨询

法律事务:

寻找司法判例

合同要求的法律分析

维护:

原子能反应堆的事故诊断

质量保证

职业技能:

语言障碍病理咨询

营养建议

园艺和植物的护理

宠物护理

摄影咨询

气象预报

图 2-5 目前已开发的一些执行层分析型专家系统

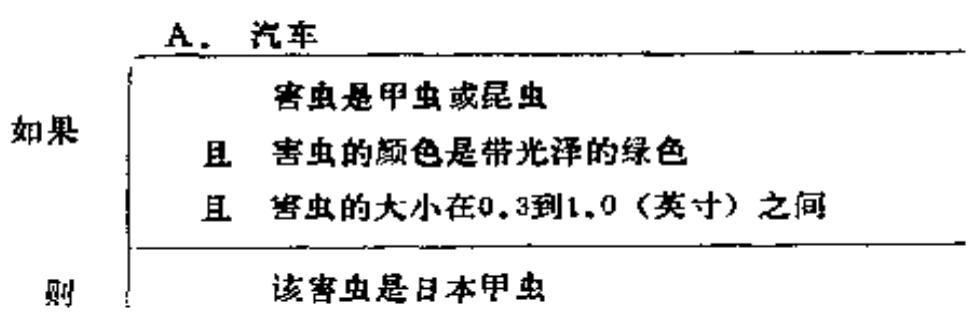
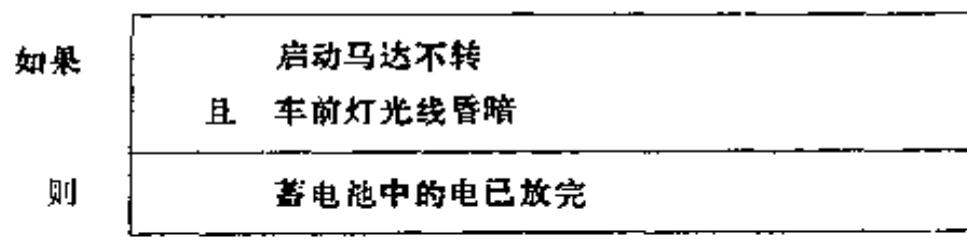
家系统中获益。

2.3.1 分析任务

专家系统能做哪些分析任务呢？回答这个问题的最好办法也许是列举出一些成功的专家系统，正如图 2-5 所示。也许你已经发现，这些应用包括了执行层的全部内容：采购、生产、销售和推销，财务管理、法律事务和维护服务等等。

诊断问题是生产领域的分析任务中最为重要的一种。图 2-6 给出了两条诊断规则。其中第一条是有关汽车故障的；第二条是判断园艺问题的。

租赁是另一种经营形式。在图 2-7 中给出了一条租赁规则。



B. 园艺

图 2-6 经营活动中的诊断规则

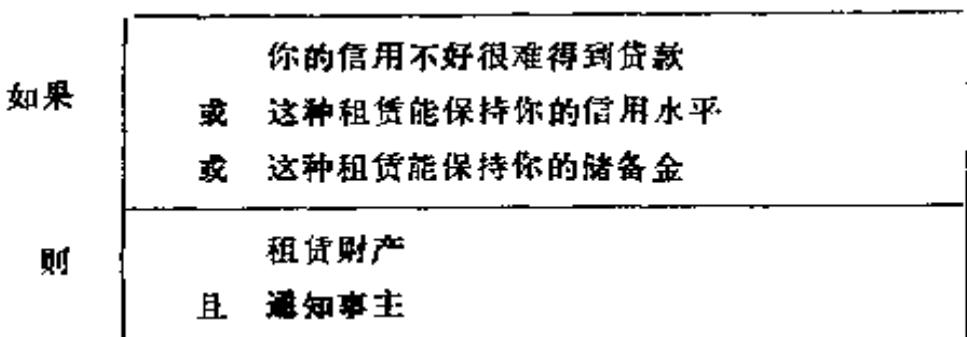


图 2-7 租赁规则

2.3.2 综合任务

专家系统不仅能够帮助你配置计算机系统，而且可以是通信系统，也可以辅助其它的配置任务，例如在车间里布置机器，在生产线上布置工作站，或在商场陈列商品等等。

专家系统还可以辅助机械产品的设计，它可以估计原材料和成品的数量，建立生产成品的机械化工艺流程，甚至为每个操作步骤提供详细的指令。

2.4 控制层的应用

控制层对机器的运行，也包括对人员的行动进行鼓励、监控和调整。这里面存在人员的问题、机器的问题以及培训解决上述问题的人员的问题等等。

2.4.1 人员问题

已有一些成功的专家系统能够分析人的心理反应，筛选求职申请，判断是否存在利益冲突，核准开支需求和辅助审计等。

在那些对安全、洁净或健康等要求十分严格的领域，如生产医疗用的、自动化的和用于研究的产品，质量控制人员也可以建立专家系统以保证其产品的质量。

2.4.2 机器问题

机器、包括机器人都是不能自行运转的。必须由人来对它们进行安装、调试、测试和维修。可以借助专家系统来布置机器和使机器正常运转。例如，最近专家系统被用于大型汤料制造厂，它能使非熟练工人辨别产生问题的原因，并使“炊具”正常地运行。

2.4.3 培训人员

如果精明的行政领导能够指导和培训他的部下、同事都不浪费时间的话，经营管理也许会更顺利些。刚好这正是专家系

统可以大显身手的地方。

专家系统可用于培训，在培训中既需要分析也需要综合。主管和培训经理们可以利用专家系统来训练新工具的使用者、机器的操作手，以及尖端电子或其它高技术设备的运行和维护人员。

在美国政府的劳工部里，有这样一个专家系统，它能够指导失业保险部门的官员。保险部门的官员们可以利用这个专家系统接待与劳资纠纷有关的失业保险人，以确定该申请者要求的补偿是否合法。由于专家系统告诉了官员应该问那些问题，而不问那些问题，因此使接待工作效率非常高。

可以将所有表示企业系统和过程的规则存贮到专家系统中，这样企业里的任何职员只要回答专家系统几个问题，专家系统就能够立刻给出应该遵守的合理工作步骤。同样，主管人事的官员可以建立人事政策专家系统；工厂的保卫人员可以建立安全专家系统；主辩律师可以建立法律事务专家系统；还有主任医生可以建立健康专家系统等等。

2.5 计划层的应用

由于目前尚未很好地掌握计划过程本身的运行机制，因此建立有关计划应用的专家系统就显得较为困难。当然，也许某些必要的分析可在专家系统帮助下做出。但是，在计划层中大的问题都是综合型的，需要通盘考虑所有因素，才能制定指导公司或企业未来发展的计划。

尽管我们不可能完全了解计划过程中的所有因素，但在做计划时，至少应对你希望大家都遵守的那些要求有清楚的认识。如果你的方案有助于处理大量的日常事务，那么，存贮了你的这些方案的专家系统将使你受益匪浅。专家系统还能用来

存贮预测标准，以避免预测的结果因人而异。

如果你是一位涉足一项新的风险事业的企业家，你对应该怎样工作，心里还没有谱。因而你希望以一种已经习惯了的工作方式从事经营，你希望新的雇员也能象你的老部下那样与你合作，那么，如要不经过面对面的传授就能达到目的的最好办法就是利用专家系统。在建立这样的专家系统方面，需要花费的时间也许是相当可观的，但却是非常值得的：除去建立专家系统带来的直接便利之外，建立专家系统的过程本身会帮助你提炼自己经营企业的目标和方法等方面的经验。

2.6 专家系统的应用领域

并不是经营管理活动中的每个领域都是专家系统领域。所谓专家系统领域，即按当前的技术发展水平，基于规则的专家系统所能应用的领域。然而，只要问题领域满足下述两条准则，就有理由肯定它是一个专家系统领域：首先是检验领域本身的准则；其次是确定领域中专家技术秘诀范围的准则。

2.6.1 检验领域的准则

- 领域的有界性
- 问题的可分解性
- 没有严格的求解方法
- 关系和数据的不确定性
- 易变性

领域的有界性 为了建立能在给定领域提供咨询的实用专家系统，应能明确地划定该领域的范围。该领域应该只具有有限几个目标、解决问题的途径，关系以及其他相关因素。而

且，目标和解决问题途径的数目应尽可能地小，即关系和因素的数目不能大到容易引起混乱的地步。

领域的边界应该是明确的，这种明确不是哲学意义上的，而应是容易确认领域中包含什么和不包含什么。

问题的可分解性 可分解性意味着领域中的问题可分割为较小的子问题，而这些子问题可以独立地获得解决。可分解问题的一个典型例子是汽车故障的诊断，在那里，每个过程由一系列确定的步骤组成。

可分解性还意味着你可以从一个有明确定义的状态转移到另一个有明确定义的状态。这也意味着没有反馈现象，即某一状态的输出又回送到它的输入状态中去。反馈现象经常出现在电子、自动机械和经济系统中，以及许多生物和其它自然系统中。反馈从根本上破坏了状态的可分解性。

在许多领域中，反馈是起决定作用的，这些领域一般不是专家系统领域。

没有严格的求解方法 如果在一个领域中，解决问题的方法必须象科学的研究那样，使用严格的定律和方程，则这种知识领域一般也不是专家系统领域。在这些领域中，应用严格的数据技巧，人们可以比用专家系统更迅速、更可靠地得到解答。但如果解决问题的方法是利用从多年实践经验中积累起来的不严格的经验知识，那么该领域就很可能是专家系统领域。

关系和数据的不确定性 在有的专家系统领域中，我们对所掌握的知识有绝对的把握；而在另一些专家系统领域中，我们对所了解的知识并不是总有百分之百的把握，我们必须考虑其中的不确定因素。

也不是所有需要考虑不确定性的领域都是专家系统领域。例如，在许多研究工作中，数据是以时间序列的形式给出的，

其统计规律需要较严格的求解方法。统计在许多其它经营领域中也发挥着很大的作用。然而，如果能够获得当前情况的“快照”，并能用这些“快照”数据形成成功概率很高的结论的话，那么可以考虑使用专家系统。

在许多经营管理领域中，诸如预测、计划、销售和人事问题等，事实是不确定的，专家系统特别适用于这些有模糊数据的应用。

易变性 大多数问题求解系统都不太适应迅速变化的环境，而变化实在是所有经营管理领域的主要特征。明天的情况肯定不同于今天的情况，未来的经营环境和约束企业的因素，例如税收、利率等，都将与现在不同，明天的方法、技术以及办事程序也将不同于今天。

如果你从事的经营领域处于快速变化的环境之中，那么这可能是专家系统领域。

2.6.2 评价专家经验的准则

下面三条准则可帮助你判断能否建立专家系统：

- 因素的可确认性
- 关系的可表述性
- 经验的可交流性

因素的可确认性 那些在解决问题时必须考虑的因素应该是能够确认的。你必须知道这些因素是什么，它们起什么作用，以及它们各自的相对重要性等等。

关系的可表述性 除了可确认的因素以外，还应该很好地了解所考查的领域，以便能够清晰地表述因素之间的各种关系。这项工作的目的完全是为了实用，而不是为了表述理论，

除非表述理论就是专家系统的目标。例如，假设你想建立医疗诊断专家系统的话，那么，你就应该在正确地表述各种疾病及其原因和治疗方法的同时，又不致于陷入高深的医学理论的困扰之中。

可表述性意味着你所掌握的知识不是基于某种模糊的直观，而是遵循一定的关系和组织方法的。同时，你可以把这些知识以书面文字的形式表述出来。

经验的可交流性 你不仅要了解事实，能表述它们之间的关系，而且应该能与非专业人员讨论你的知识，并用语言的形式为他们提供咨询。进行这种可交流性检验的一种简单办法是：利用电话，你能否在20分钟时间之内为你的顾客解答一个有代表性的问题。在电话交谈中，可以先由顾客描述情况，然后顾客回答几个你为了进一步弄清情况而提出的问题，最后，由你根据情况给出咨询意见。

2.7 复合领域的应用

如果在考查了上述两个检测准则之后，你发现你的应用不

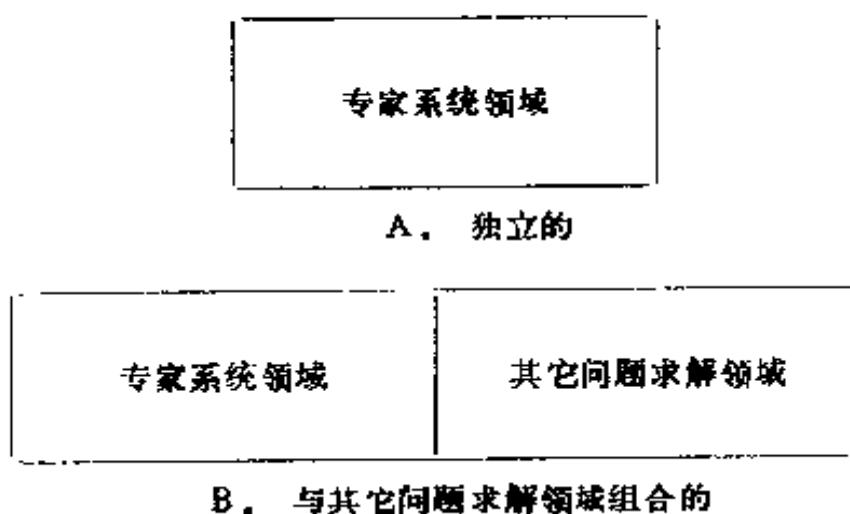


图 2-8 完整的问题领域和专家系统领域

是专家系统领域的话，你可以将你的应用分解成两个协同工作的问题求解领域，即专家系统领域和另一种问题求解领域，见图2-8。通过这种分解，你可以用不同的技术解决不同的问题。

可以通过专家系统与别的问题求解系统合作，或为怎样使用其它类型的问题求解系统提供咨询，这两种方式来实现专家系统与其它类型问题求解系统的协同工作。

2.7.1 组合型系统

目前在软件市场上已有许多各式各样的软件包，它们分别以各自特有的方式处理数据。在这些软件包中，有数据库管理系统，如dBASE；电子表格，如Lotus 1-2-3；图形设计软件包，如AutoCAD；以及统计软件包，如SAS等。如果专家系统不向用户索取数据，就可以与上述软件联结，以便直接从它们那里获得数据。当然，在专家系统执行推理任务时，数据处理软件包仍可以执行它们自己的工作。这种“二重奏”式的工作方式可以产生极强的组合功能。

2.7.2 咨询型系统

每个软件包产品一般都配有详尽的用户手册。但是无论这种用户手册编得怎样详尽，当用户需要立刻了解软件中的某项内容时，总会遇到这样或那样的困难。也许答案就隐藏在那厚厚手册中的某一页里，但是要马上找出来却不是件容易的事情。为什么不可以利用专家系统来帮助软件包的用户，特别是新用户，来解决上述问题呢？使用这种专家系统，用户只需回答几个简单的问题，专家系统就能够告诉他怎样解决他在使用软件包中遇到的问题。

专家系统能够提供的咨询范围十分广阔，如研究原型SACON所做的那样。SACON是一个为结构工程师设计的咨询型专家系统。它本身并不直接解决具体的结构设计问题，而是通

过向结构工程师提供有关如何使用大型仿真程序包的咨询建议，来间接地解决问题。例如，一位飞机结构工程师想评价波音747的机翼在飞行负荷下的结构整体性，他首先必须描述机翼的数学性质，SACON然后建议怎样使用仿真软件，以便确定机翼在各种压力情况下的性态。图2-9中有一条SACON的规则。

仿真不仅在工程上被广泛应用，而且还在预报、销售、生产、工程项目管理、财务计划和其它许多经营管理活动中得到应用。

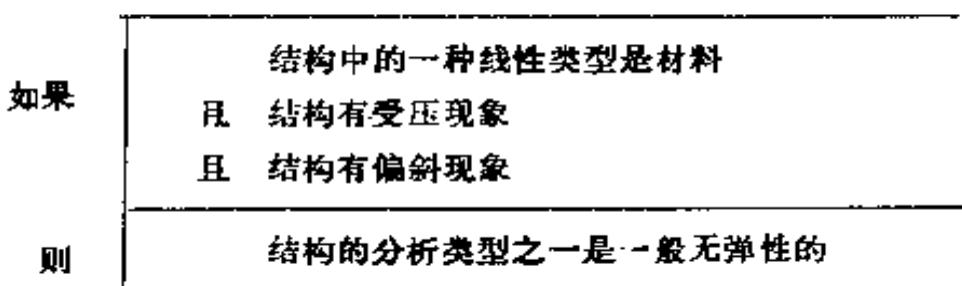


图 2-9 一条SACON规则

第三章 图形化规则语言

到目前为止，你已经知道了专家系统的一般概念和它的一些经营管理应用。由于专家系统开发工具不同于其它类型的软件包，为了了解怎样建立专家系统，你必须先学习一种新的语言。通常，在学习使用一种软件包时，只需要学会一组命令。然而，专家系统开发工具要求掌握的不仅是命令，更重要的是语言。你必须努力学习用这种语言描述问题，这些问题的解决办法以及获得结论或答案所必须考虑的条件等等。

我们将要学习的这种语言是十分简单的，它具有良好的结构，且拥有一种近似自然语言的风格。

在建立专家系统时，完全使用自然语言（如汉语或英语）是有一定困难的。自然语言在表达不定事实和逻辑关系方面是十分有效的。但是，由于自然语言在形式上不够简洁和严谨，使它不是处理上述事实和关系的理想工具。正是由于这个原因，本章我们将介绍一种专门的图形化规则语言。

顾名思义，图形化规则语言能够很容易地表述规则间的关系。借助简单的规则模版可构造规则。此外，规则还将与另一种称作上下文树的图形结构发生联系。

3.1 上下文树

专家系统领域中的问题可以分解成若干子问题，这些子问题又可以进一步分解成更小的子问题。这样，领域的结构或轮

廓就有点象一棵倒挂着的树，因此我们称之为上下文树。

3.1.1 上下文的概念

上下文这个术语究竟是什么意思呢？按词典的解释，上下文是“一个讲话或一段文字的前面或后面的某一段内容，它与一个特定的意思或词组联系在一起”。“上下文是一种约束结构，它被用来使各个意思之间发生关联。同样的一段话在不同的上下文中会有不同的意思。在讨论某人的言论时，你常会问“他说这些话的上下文是什么？”在与别人辩论时，你也会常说：“请记住我的上下文，不要断章取义地理解我的话……”。

因此，上下文是这样的概念，你可以用它来组织你想思考的观点、目标、问题、过程、注释或其它任何项目。在第一章的风险投资规则中，规则是表示投资者应该做什么的。换句话说，规则以“投资者”做为它的上下文。

3.1.2 根上下文

在风险投资领域中，基本的上下文显然是“投资者”。由于它是上下文树出发的根，因此它被称作根上下文。所有描述投资者受经济形势影响的财务状态的规则都应置于“投资者”上下文中。

3.1.3 子辈上下文

如果你的规则不多，或这些规则都只与一个大的主题有关，那么，只要根上下文就足够了。然而，当你建立有许多条规则的结构时，你可能希望增加从属上下文。如果把根上下文视为父辈，那么这些从属上下文就是“子辈”。图 3-1 给出了一个由根上下文“投资者”及其四个子辈——“债券”、“股票”、“存款凭证”和“金融基金”组成的上下文树的例子。

增加第三层分枝可产生第三代子辈。例如，假若将这种上下文加入到“债券”上下文之下，它们就是“债券”的子辈。

总之，上下文树描绘了专家系统领域的轮廓。

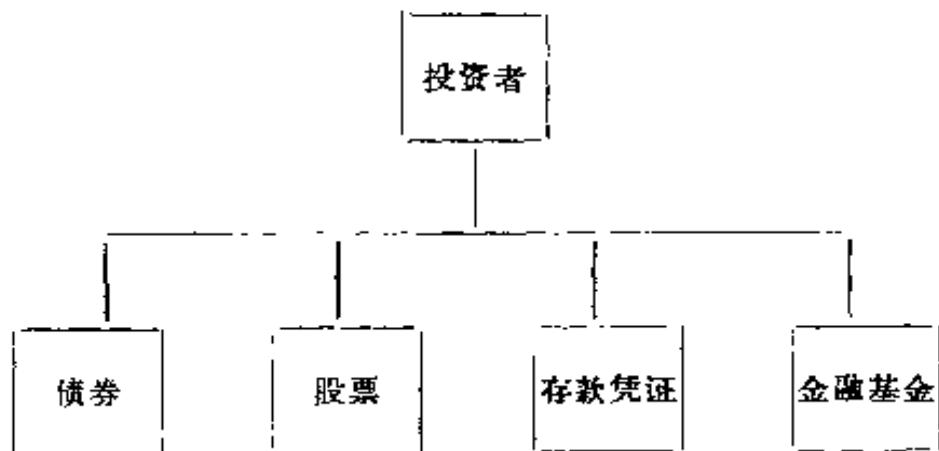


图 3-1 风险投资领域的上下文树

3.1.4 上下文实例

有时需要区别不同的“投资者”，以便对两个或更多的“投资者”进行比较。甚至可以允许一个“投资者”有多个“债券”子辈。为了实现这种功能，应该允许有上下文实例，如“投资者-1”、“投资者-2”等等。

上下文实例的引入极大地扩展了专家系统的问题求解能力。然而，并非所有专家系统都具有这种功能。

3.2 事实模版

图形化规则语言的语法是用模版来描述的。通过往模版中的相应位置填写合法的词汇就可以描述一个事实。图 3-2A 中是一个事实模版，它由下述三个语言成分组成：

- 属性
- 谓词
- 值

只要理解这些语言成分的意义，就可以利用事实模版书写事实。

3.2.1 属性

属性，也称参数，是描述相关上下文中有关内容的数据项。因此，（培养）“基”、“税金”和“信用”都是属性，因为它们分别描述了上下文“培养”、“公民”和“借贷者”。属性可以描述状态、性质、动作或事物等。

属性也可以是一个长的叙述句的缩写。例如在图 2-7 中，句子“保持你的储备金”可用简写“保持现金”来表示，这样可以带来表述上的方便。

3.2.2 值

一个属性可以取一些确定的值。如属性“会飞”可以取值“是”或“不是”；即一个“动物”也许“会飞”或“不”“会飞”。这种属性称作是非型属性。

属性“信用”可以取几个可能的值，如“好的”、“可以”和“不行”。属性“税金”可以有许多数量值。上述属性称作单值的，因为任何时刻，这些属性都只能取其可能取值中的一个值。

象“读过的学校”这样的属性称作多值的，因为在同一时刻，这种属性可以取多个值。例如，一个“申请者”可以输入他曾就读过的几所学校的名称。

3.2.3 谓词

值与属性的联系是由谓词来完成的。下面是几个常用的谓

词：

- = (是)
NOT= (不是)
< (小于)
> (大于)

3.2.4 事实模版的例子

图3-2B 中有四个事实。其中第一个事实是“基” = “咽

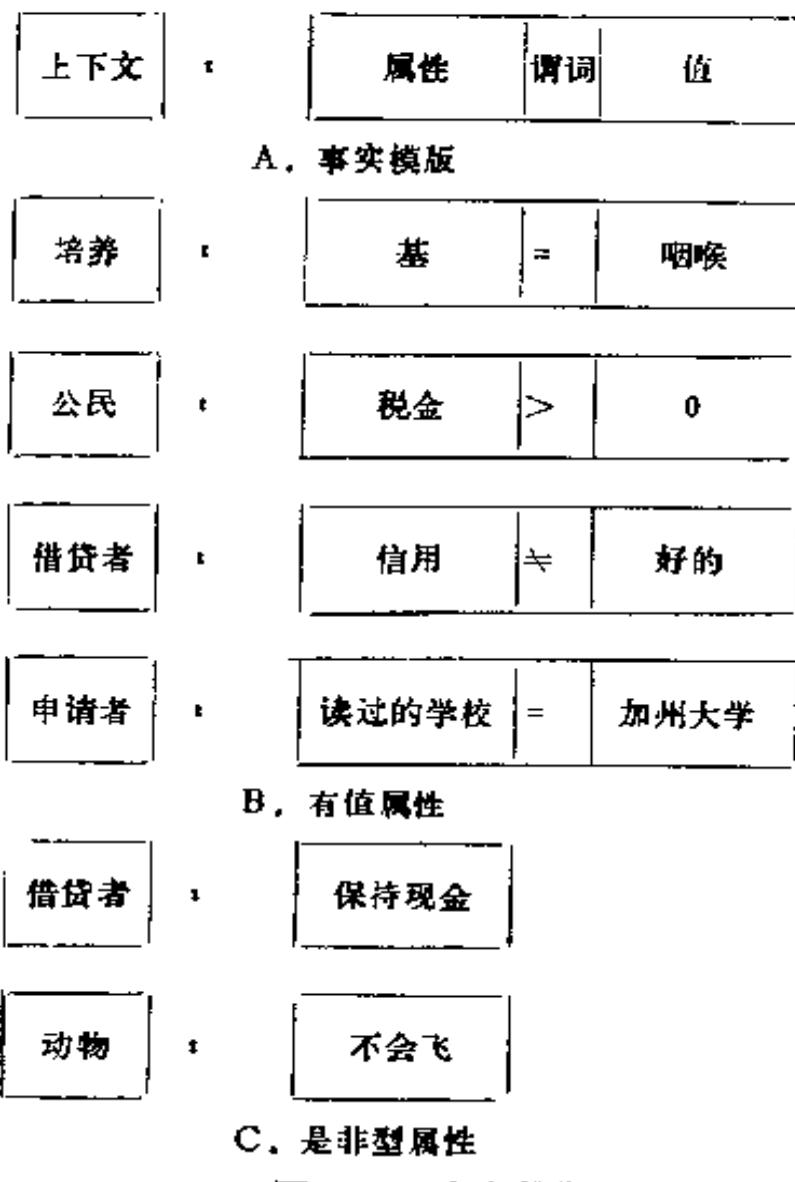


图 3-2 事实模版

喉”。通过与上下文“培养”的联系，这个陈述的意思就是：“培养”“基”是“咽喉”。

第二个事实“税金” >0 。同样，通过与“公民”上下文联系，它的意思就是：“公民”的“税金”大于零。

第三个事实“信用”NOT=“好的”。它的意思是某“借贷者”的“信用”不是“好的”；而是“可以”或“不行”。

第四个事实“读过的学校”=“加州大学”。它的意思是“申请者”曾“读过的学校”之一是“加州大学”。

图3-2C告诉我们对于是非型属性怎样简化模版的表示。对于肯定的情况，如同“保持现金”的形式一样，只需表示出属性就可以了。对于否定的情况，需将“不”加到是非型属性之前，如“不”“会飞”所示。

3.3 规则模版

如图3-3A所示，利用规则模版可以将事实组织起来形成规则。每条规则有一个如果部分和一个则部分。这两个部分都是由若干个表示事实的子句通过函数联结起来的。其中则部分中的每个子句还可以有一个可信因子CF与之联系。

规则模版中用到的主要语言成分有：

- 函数
- 属性
- 谓词
- 值
- CF

所有这些成分都与上下文以及语言的其它成分有联系。

	函数	属性	谓词	值	CF
如果		子子	句句·	1 2	
则		子子	句句·	1 2	

A. 规则模版

如果	且	状态 社会保险 有积蓄	=	受雇 最高	
则	且	风险性	=	高	80%

B. 风险投资规则

状态 = 受雇
社会保险 = 最高
有积蓄
风险性 = 高(80%)

C. 图形化规则

图 3-3 规则模版

3.3.1 子 句

规则模版中如果部分的每个子句都遵循下述事实模版：

属性 - 谓词 - 值

在则部分中还需增加CF：

属性 - 谓词 - 值 - CF

3.3.2 函 数

函数能够联结子句中的属性和其它元素，也能联结子句。

这些联结符分算术的和逻辑的两种。

子句中的元素可用算术函数来联结，如：

+	(加)
-	(减)
*	(乘)
/	(除)

自然，这些子句中的属性一定都是数量型属性。

逻辑函数用来联结子句。逻辑函数有“与”和“或”等。大多数子句都是用“与”来联结的，但是有时也用“或”函数联结。

有些专家系统开发工具不支持“或”函数，它们用若干条分离的规则来完成“或”函数的功能。在有的系统中也有用别的名称和方法实现其它函数的。

3.3.3 可信因子 (CF)

可用CF来表示你对结论子句的信任程度，CF是一个可在+100%到-100%之间取值的量。如果你给CF的值为+100%，则表示你对子句的成立是肯定的或确信的；反之，若你给CF的值是-100%，则表示你对结论不成立是确信的。如果你对子句并不十分肯定，或者子句在已知条件下并不是总能成立，则你可以在90%到1%之间选择一个CF值，但究竟是哪一个值，这依赖于你对子句的肯定程度。同样，如果由于信息不完全或你无法确信你对子句的否定感觉，这时，可以在-99%到-1%之间选择一个CF值。

正如图3-4所示，则部分中的每个子句都可以有自己的CF值。

白血球数量<2.5
细菌 = 大肠杆菌 (75%)
细菌 = 绿脓杆菌 (50%)
细菌 = KEBSTELLA 肺炎菌 (50%)

图 3-4 不同则-子句的CF值

3.3.4 规则例子

如图3-3B所示，这是一个用规则模版表示的风险投资规则。参考这种模版，你就可以编写专家系统的任何规则。

图3-3C告诉我们，一条规则是怎样用图形来表示的。

第四章 开发专家系统的过程

现在，你可以纵览开发专家系统的整个过程了。专家系统的开发过程是这样的：首先，必须制订一个开发计划，然后按照下述四个开发阶段逐步进行：

- I. 确定专家系统领域
- II. 开发表示知识的词汇
- III. 建立知识库
- IV. 测试知识库

阶段 I 和 II 与待解决问题的性质和解决问题时所使用的概念和词汇有关。这部分工作不需要软件工具的帮助也能完成。在阶段 III 和 IV 中，专家系统开发工具将帮助你把知识转化成可执行的软件。

需要指出的是，介绍这样一个有四个阶段的方法仅仅是为了使一些基本概念容易理解，为了说明一些必须经过的重要步骤以及一些可能遇到困难等等。一旦你进入了自己所扮演的工程角色，毫无疑问，你肯定会展形成自己特有的工作模式。当然，在你自己的专家系统开发实践中，你可以借鉴本书介绍的各种方法和技术。

4.1 开发计划

在决定开发专家系统之前，你必须首先从主观和客观两方

面考虑，以便弄清楚是否真正需要这样的系统，以及你是否有足够的时间来开发专家系统。如果你的问题领域非常简单，那么一个月时间也许足够，但若是较复杂的领域，则可能需要一年甚至更长的时间。对于一个规则数目不超过三百条的小型专家系统来说，你至少应该计划六个月的时间来完成它。

任何事先制定的时间表毕竟是一种臆测。为了确认整个工程究竟需要多长时间，可以先花一个月左右时间，在领域中选择一个小而有代表性的原型，并加以开发，开发原型有助于你加深对问题领域、专家系统开发工具以及专家系统开发过程的全面了解，从而使你能够进一步判断专家系统是否真能为你的目标服务。

开发原型也有助于你判断是否需要知识工程师的帮助。知识工程师通常是有计算机编程经验的技术人员，他在如下两个方面都造诣很深：首先，他知道怎样从专家们（其中包括你）那里“搜刮”知识，并把获得的知识转变成规则。另一方面，由于他受过良好的工程训练，他知道怎样测试和精炼知识库。

若是开发小型专家系统，你也许不需要请知识工程师帮忙。但是对于大而复杂的系统，你最好请一位。若是更大型的工程，你可能还得组建一个工程小组参加才行。

如果把开发专家系统原型比作在游泳池中练习水性的话，那么，开发真正实用的专家系统才是一项需要在江河中披波斩浪、奋勇直前的艰巨任务。尽管如此，在完成了原型的开发之后，你会比以前对整个工程的关键部分和困难有更全面、更深入的了解，从而能做出更切实际的计划。

本章下面将介绍的四个开发阶段，无论对开发原型系统还是最终的实用系统都是适用的。

4.2 阶段 I：确定专家系统领域(图4-1)

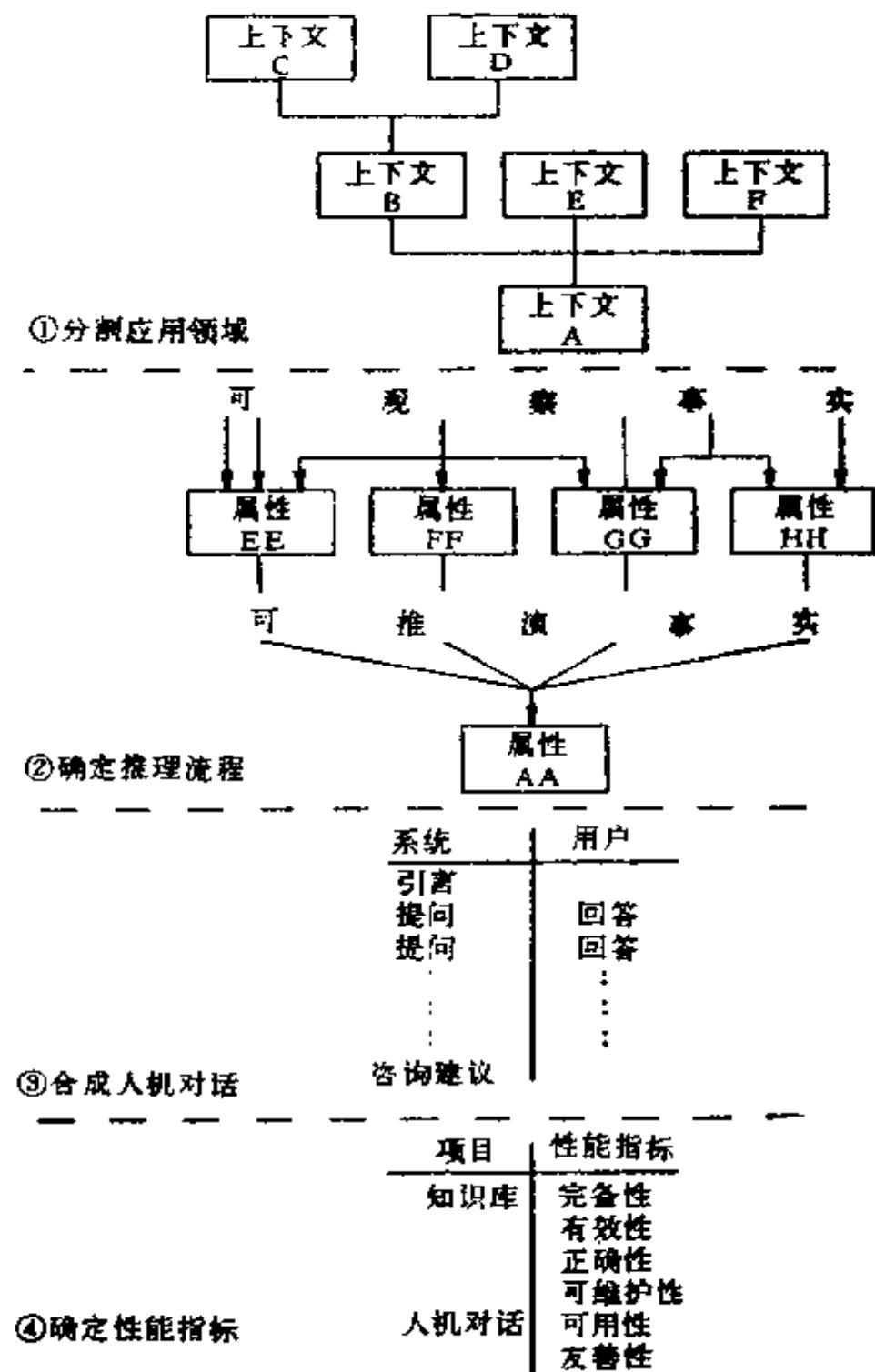


图 4-1 阶段 I：确定专家系统领域

首先需要进行的工作是确定专家的经验知识领域。这个过程包含下述四个步骤：

- 分割应用领域
- 确定推理流程
- 合成人机对话
- 确定性能指标

4.2.1 分割应用领域

如果应用领域很复杂，应该先把它分割成若干容易理解的部分，把问题分割成若干子问题，把大而粗的想法细分成小而精的主意，把实在设备分离成它的各部件，把一系列事件看作是若干较小的相关事件的组合。

在分析的基础上，可以画出表示问题领域的上下文树。一个企业或团体的组织结构图就是说明上下文树概念的极好例子。例如，一个公司由若干个部组成，每个部又分成几个处，等等。组织结构有效地分解了企业管理这个复杂的问题。

建立上下文树最重要的原则是：简化。不到必要的时候不应随意增加层次。

尽管人们通常都把上下文树的根画在上方。但由于以后将解释的原因，我们将上下文树的根画在树的下方，因而它的所有分枝都是向上生长的。

虽然，我们在这里只讨论了上下文树，应该指出的是，应用框架和将规则分组也能够获得类似的分割效果。

4.2.2 确定推理流程

如果说上下文树反映了问题领域的一个侧面的话，那么推理流程图则反映了领域的另一个侧面。推理流程图用来表示处

在上方的可观察事实与处在下方的目标之间众多属性的关系。在问题的这两个极端之间可能有若干层可推演事实。从形式上看，推理流程图也很象是一棵根在下方的树，与上下文树不同之处，是它具有更多的细节。

正如我们已在第二章中介绍过的那样，从目标到可推演事实到可观察事实的推理称作分析，而从可观察事实到可推演事实到目标的推理称作综合。由此可见，推理流程图只从总体上定义了推理的路径，并没有限定领域需要的问题求解方法是分析还是综合。

首先，我们可以确定一个（或几个）表示目标的属性，然后，确定表示事实的属性，最后，画出所有反映从目标到可观察事实之间属性关系的联线，即推理流程图。

4.2.3 合成人机对话

确定专家系统领域的另一种方法是设想你建立的专家系统与它的用户之间的对话过程。首先，考虑一下这样一些必须解决的实际问题：你希望系统提出哪些问题，你期望从用户那里获得的回答类型以及你设想的系统应该提供怎样的最后咨询建议。

在记录下上述对话之后，可以转而考虑另一类问题，并记录下你为它所设想的对话。用这种方法记录了若干组有典型意义的对话之后，你就拥有了充足的素材，可以用它们来确定专家系统领域。

4.2.4 确定性能指标

为将来测试整个知识库确定专家系统的性能指标。

4.3 阶段II：开发表示知识的词汇

保证专家系统成功的一个最好办法是在领域确定之后，首先开发一个清楚明了、客观实用的知识词汇表（见图4-2）。

属性表

上下文	属性	可观察 / 推演	翻译
① 确定属性	A	AA BB CC	观察 推演 推演
	B	GG HH	观察 观察

属性—值表

上下文 A

属性	可预先确定			不可预先确定				
	是/非	单值	多值	期望值	文本	数值	单位	范围
AA	√							
BB		√		好的 一般 较差				
CC						正数	寸	0— 100
DD					√			
EE			√	X Y Z				

② 定义属性的值

函数	属性	谓词	值	C F
③ 编写检验词汇的规则	AA			
	CC	=	80	
	BB	=	好的	80%

图 4-2 阶段Ⅱ：开发表示知识的词汇

因为它们是你在整个专家系统开发工程中都将使用的元素，所以，必须仔细地加以选择。阶段Ⅰ的三个主要步骤是：

- 确定属性
- 定义属性的值
- 编写检验词汇的规则

4.3.1 确定属性

阶段Ⅰ的工作可以在很大程度上决定你的知识库的主要属性元素以及它们与各上下文之间的联系方式。可以从两个方面来说明每一个属性：给属性一个自然语言翻译；使它隶属于上下文树中的某一个上下文。为此，可以把全部属性列在一张属性表中，这是一种简便，但非常有效的方法。

4.3.2 定义属性的值

可以采用属性一值表来定义每个属性的值。属性中有些可能是是非型的，有些在任一时刻仅能取一个值，而另一些则可同时取若干个值。有些属性需要数值型值，有些则需要字符型值，另一些可以从一特定值表中有选择地取值。

同理，为了简化，应尽可能地使用是非型或多选择型属性。

4.3.3 编写检验词汇的规则

借助规则模版的帮助，编写规则以判断属性及其值是怎样定义的。在这里，还不必写出所有规则，只需编写足够数量的规则便可，且以能够判断所选择的词汇是否有效为准。

为了满足编写规则的需要，可能要修改个别属性的定义。不仅如此，根据需要，有时还可能得返回到阶段Ⅰ，并重新设计上下文树及其相应的推理流程图。这类反复可能需要进行多次，直到你对领域定义和知识库词汇感到满意时为止。

4.4 阶段III：建立知识库（图4-3）

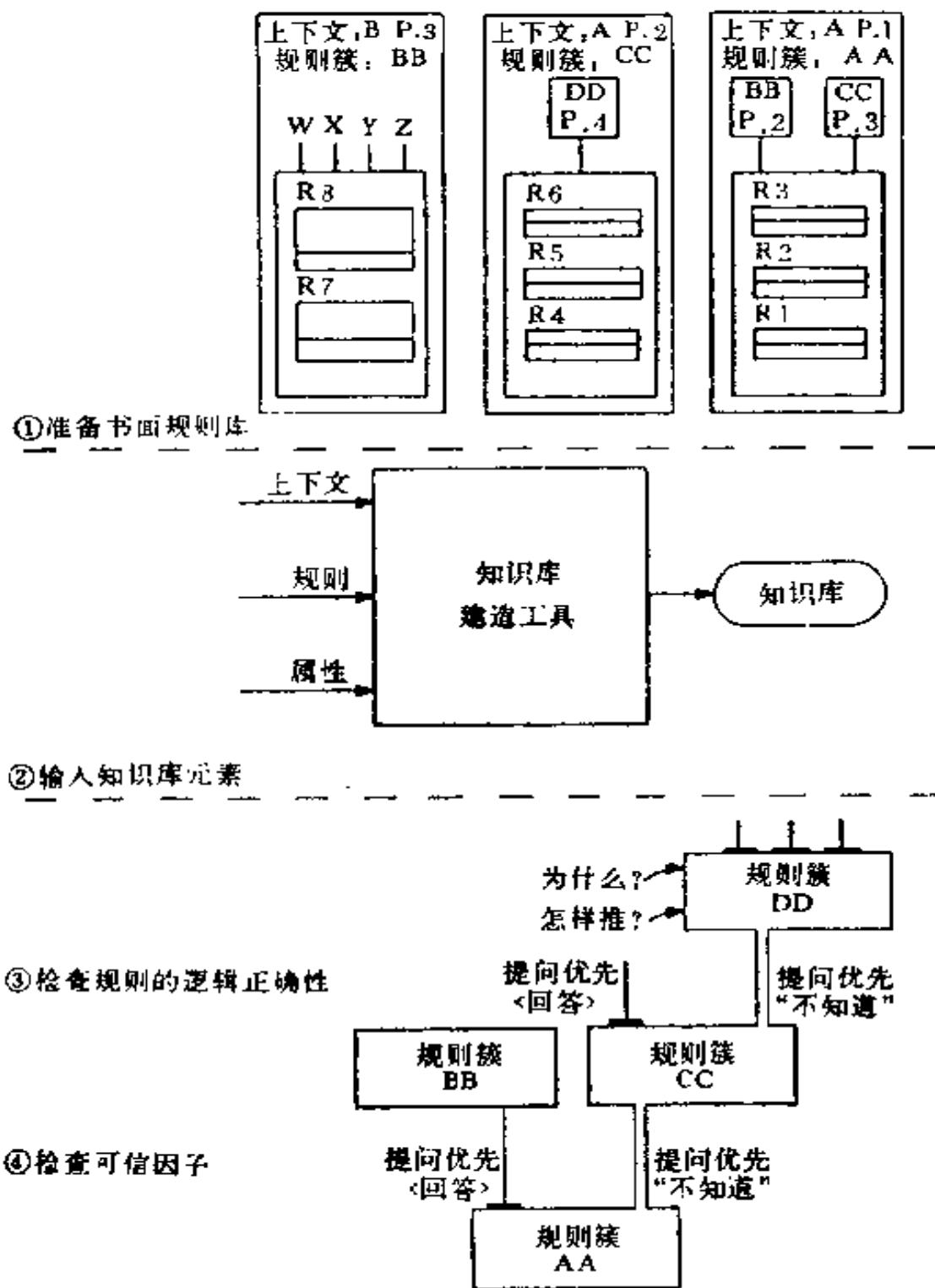


图 4-3 阶段III：建立知识库

现在，你可以利用“电脑顾问”或其它专家系统开发工具建立自己的知识库了。建立知识库是一个包含如下四个步骤的过程。

- 准备书面形式的规则库
- 输入知识库元素
- 检查知识库的逻辑正确性
- 检查可信因子

4.4.1 准备书面形式的规则库

在将上下文、规则及属性等知识库元素用专家系统开发工具输入到你的知识库中去之前，最好先将你的规则库以书面语言的形式记录在纸面上。规则库中包含专家系统的所有规则。有关构造规则库的详细内容将在后续章节中介绍。

当然，假若你的专家系统开发工具能够做归纳推理，那么，你就可能不必进行这项工作。你只需输入表示事例的表格数据，专家系统开发工具就会自动产生通用的规则。尽管如此，把一些粗略的结构数据记录在案仍然是极有益处的。

4.4.2 输入知识库元素

在利用专家系统开发工具向知识库输入知识库元素之前，最好再检查一下属性表和属性一值表，以确认每个属性是否都在合适的上下文中。接下来便可以参照规则库图输入知识库元素。在定义和输入上下文时，应该对各上下文进行仔细选择并保证它们之间的联系相对合理；在输入每个规则时，应该检查它与本规则簇以及别的规则簇中的其它规则之间的关系；在定义每个属性时，要确保它隶属于合适的上下文，确保它的唯一性以及能在规则中被合理地引用。

4.4.3 检查规则的逻辑正确性

以书面记录的规则结构做引导，利用专家系统开发工具提供的调试命令，你可以对输入的规则进行检查，以保证其逻辑正确性。通过人机咨询过程，可以找出遗漏的、冗余的或逻辑上不正确的规则。

4.4.4 检查可信因子

在编写规则时，就应该为每条规则选择合适的可信因子。然而，在你检查规则的逻辑正确性时，还应注意引入 CF 后对系统行为的影响。要注意检查在不同的条件组合时逻辑的情况，看最终结论的可信因子是否与你对该问题的理解相吻合。

类似的要求也适用于用概率统计或其它方法处理不确定性的专家系统开发工具。

在进行上述各步骤工作的过程中，不应回避为了适应已变化的情况而进行的必要“返工”。

4.5 阶段IV：测试知识库（图4-4）

尽管在阶段Ⅲ中已有许多通常称作测试的活动，然而，阶段Ⅳ在目的上不同于阶段Ⅲ。阶段Ⅲ的目的是产生在逻辑上完备且正确的，能表示你的经验知识的规则库；而阶段Ⅳ的目的则是产生有实际应用价值的专家系统。不论是原型还是最终的专家系统，它们都应能针对用户提出的实际问题，向用户提供正确而实用的咨询建议。阶段Ⅳ的主要步骤包括：

- 准备一套测试案例
- 评价知识库
- 改善人机对话
- “开门”评议

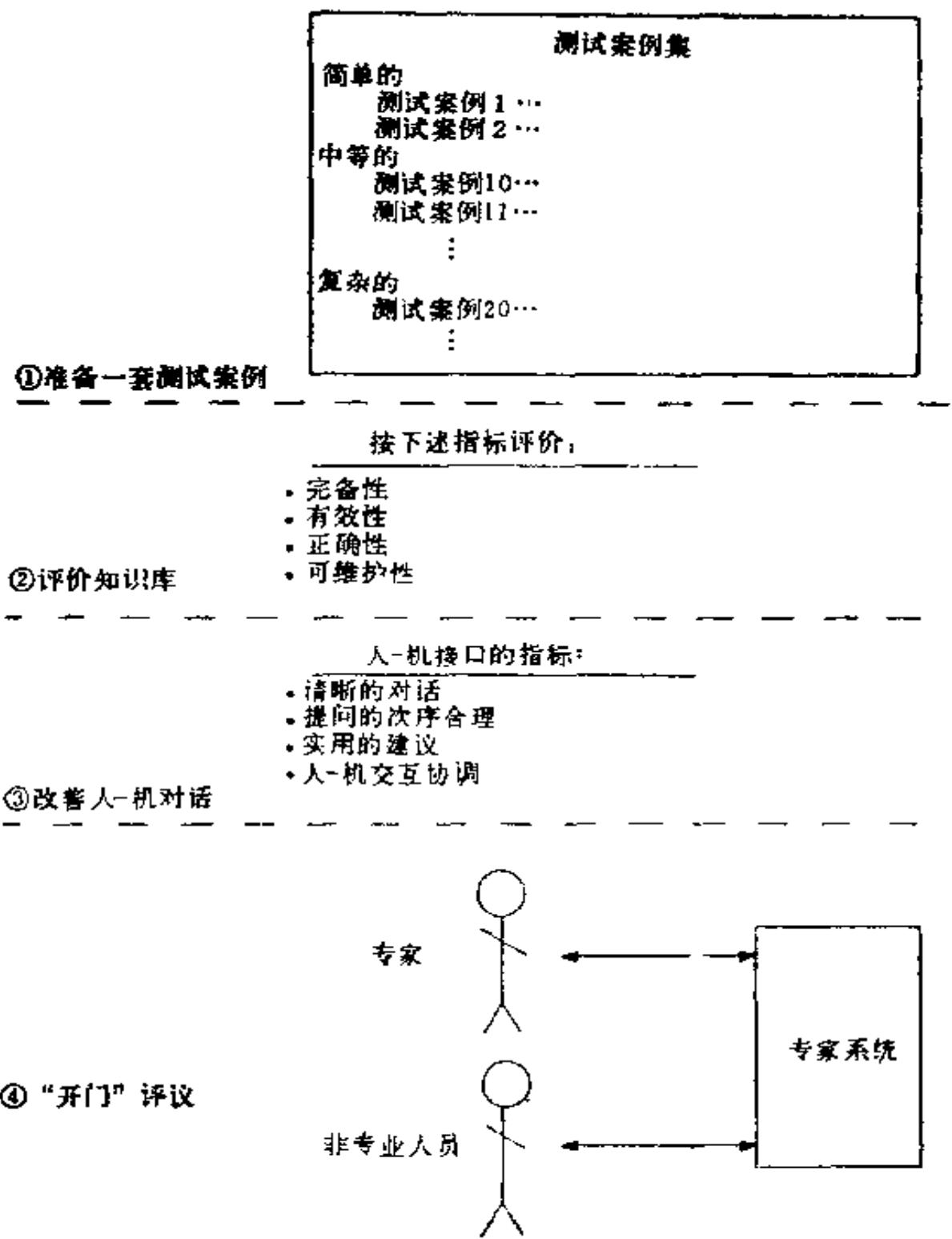


图 4-4 阶段Ⅳ：测试知识库

4.5.1 准备一套测试案例

你所选择的这套测试案例应包含足够多的情况，以便能够覆盖你的专家系统预计应该求解的问题。测试案例应以先简后繁的顺序排列。简单的实例也许只能测试知识库的一个小的局部，而复杂的实例也许一次就能够测试知识库的几个部分。

4.5.2 评价知识库

按照事先确定的系统性能指标，执行对每个测试案例的咨询，你可以分别对知识库的可用性和专家系统自身进行评价。使用追踪逻辑流程的命令可以发现错误。当发现知识库中存在错误时，应立即加以纠正并重新执行测试案例。

4.5.3 改善人机对话

除了保证你的知识库在逻辑上正确之外，还应该尽可能地使系统与用户之间的对话自然而且有效。在执行测试实例时，要注意发现是否有不合适的提问、解释以及咨询等，并思考有否改善人机对话的可能。

4.5.4 “开门”评议

为了使专家系统更实用、更可靠，应该邀请一些志愿者来充当顾客或用户，并请他们通过实际使用专家系统提出中肯的评价意见。通过这种方式，可以获得许多有用的反馈信息。

第五章 阶段 I：确定专家系统的应用领域

阶段 I 的主要目的是为后续三个专家系统开发阶段奠定基础。它的主要任务是建立表示专家系统应用领域的问题及其子问题的结构。阶段 I 的主要步骤包括：

- 分割应用领域
- 确定推理流程
- 合成人机对话
- 确定性能指标

其中前三步分别是应用领域的三种不同的分析方法。分割使你容易理解和看清问题的结构；确定推理流程使你能够清晰地了解贯穿整个领域的逻辑流向；而合成人机对话则使你看到专家在得出结论时，必须与顾客交换的信息。最后一步，应该由你为自己的专家系统确定一个性能指标。

5.1 示范实例

为了避免抽象地论述，我们将借助一个具体的实例来讲解阶段 I 及其后续三个阶段。该实例对于任何一位有普通公路交通知识的人来说都不会感到陌生。这个用于示例的领域与汽车交通事故有关，其专家系统的目的是判断汽车司机是否应对发

生的交通事故负责。这种专家系统也许对公路监理部门或保险公司有用。

为叙述方便起见，我们只对原型进行讨论。交通事故原型中的大多数规则取材于一本机动车驾驶员手册，任何申请驾驶执照的人都得通过一种与该手册介绍的内容类似的考试。今后，我们将主要讨论手册中有关交通事故责任推断的内容。

在进一步阅读之前，你最好能找一本适用于你所住地区的司机手册来阅读。然后，再跟我们继续学习四个专家系统开发阶段中的每一步骤。

5.2 分割应用领域

从司机手册中可知，问题领域似乎应该按照交通事故发生的地点来分解，并可用图5-1中的上下文树表示。

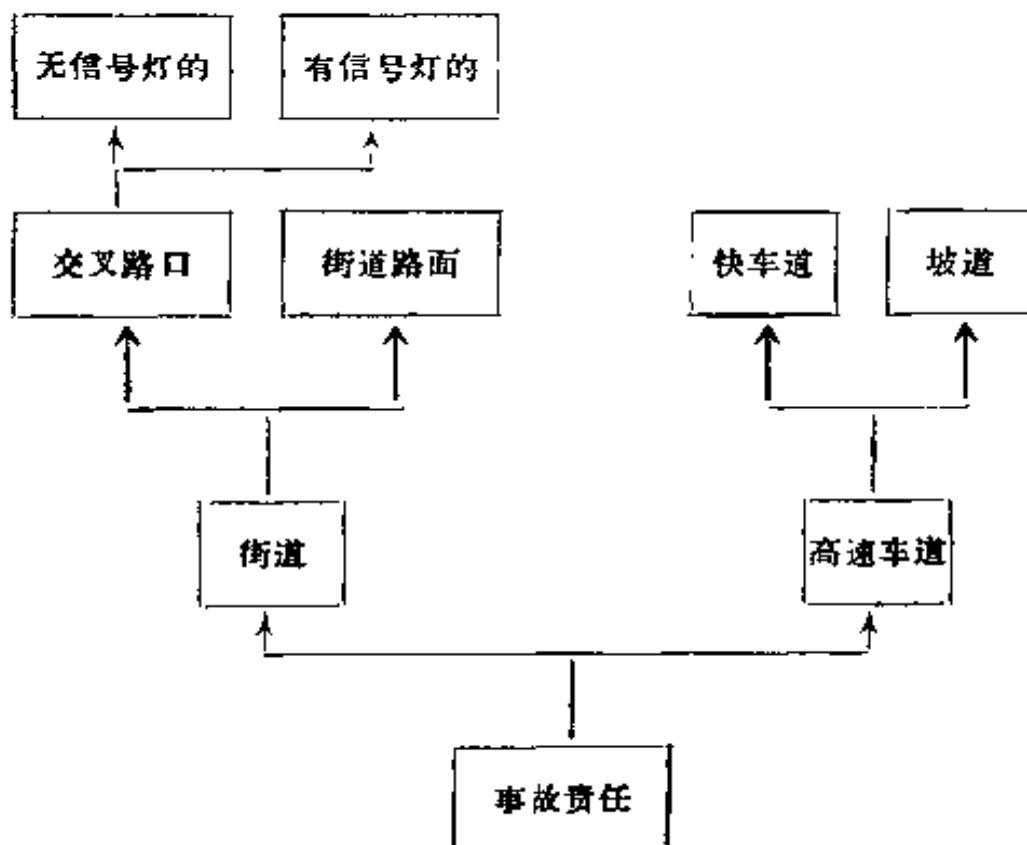


图 5-1 交通事故原型的上下文树（I）

在图 5-1 中，事故责任是根上下文。由此根上下文引出两个说明主要事故发生地点的上下文：“街道”和“高速车道”。其中街道又按照地点进一步分解为“交叉路口”和“街道路面”两个上下文。而“交叉路口”又进一步分解成“有信号灯的”和“没有信号灯的”。在“高速车道”这一枝，可以自然地按照类似的方法，把它分解为“快车道”和“坡道”。

5.3 确定推理流程

图 5-2 中给出了将系统的目标、可推演事实和可观察事实联系起来的推理流程图。其中目标是判断司机是否应对交通事故承担“责任”。“责任”依赖于三个可推演属性：“判断能力”、“车速”和“驾车动作正确”。而这些属性又进一步依赖于其它属性，其中有些是可推演的，另一些则是可观察的。所有这些依赖路径都最终结束于可观察属性。

影响司机“判断能力”的属性有“有警觉性”、“酒后开车”、“滥用兴奋剂”和“情绪异常”。

“车速”表示在特定道路上合适的行车速度，它依赖于司机正在行驶的“车速地段”以及在事故发生时他的“实际车速”，它还依赖于条件“安全的”，而“安全的”又依赖于汽车的“机械安全”以及“道路安全”两方面。

“驾车动作正确”也有多种情况，按实施驾车动作的地点可分为：在快车道上的驾车动作正确(记作“R-PROPER”)、在交叉路口的驾车动作正确(“I-PROPER”)、在街道上的驾车动作正确(“S-PROPER”)以及在停车场的驾车动作正确(“P-PROPER”)。所有这些属性的成立与否又由下述因素决定：司机实际实施的“驾车动作”、“车速”、“车道数目”、“道路标志”、“车流情况”以及“信号”等。

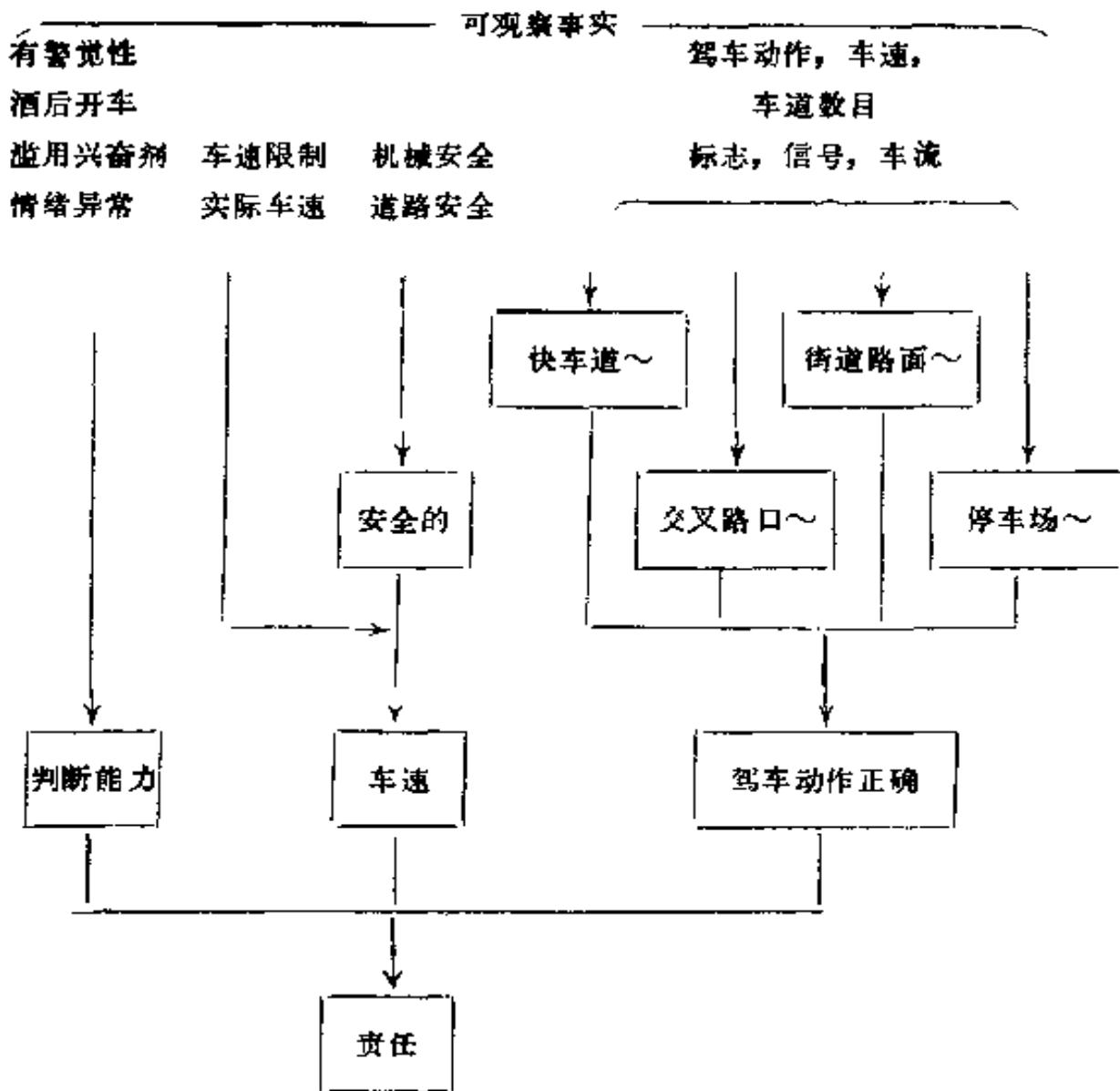


图 5-2 交通事故原型的推理流程图 (I)

5.4 合成入机对话

工作进展到这里，你也许会和我们一样，脑子里不断地转动着各种各样的想法，但又无法将它们联系起来。如果是这样的话，你不妨暂时将这些形形色色的想法“冷却”一下，而把精力集中到合成几个典型的事故情节的对话上去。

为了使叙述形象化，我们画出了如图 5-3 所示的事故情节

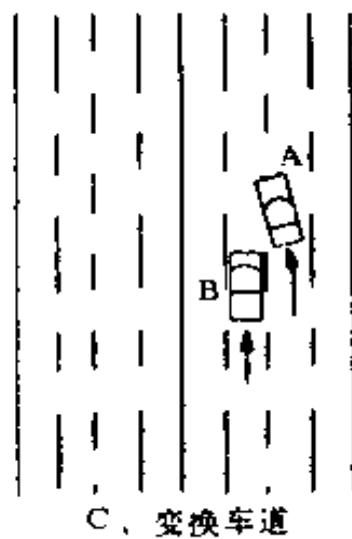
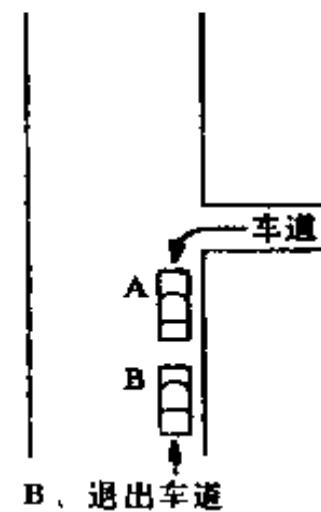
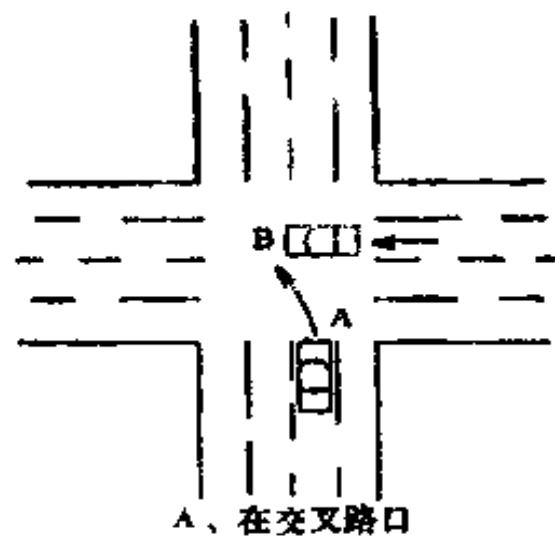


图 5-3 交通事故图例

图。其中图A说明发生在交叉路口的事故，当汽车A做左转弯时撞击了正在车道上行驶的汽车B；图B说明当汽车A倒车退出它的行驶路线时，撞击了正在行驶的汽车B，因而引起交通事故；图C则说明汽车A在超车时酿成了交通事故。

图5-4记录的是一段有关左转弯事故的对话。其中所提的问题是用来确定事故情节的。而给出的具体答案则说明了这个案例。提问应一直进行下去，直到获得了足够多的信息并能得出结论为止。

专家系统	用户
1. 事故发生在街道上还是高速车道上？	高速车道
2. 道路名称是什么？	一号公路
3. 事故发生在交叉路口还是道路中央？	交叉路口
4. 交叉路口的名称是什么？	法兰特
5. 交叉路口是否有交通信号？	没有
6. 司机的判断能力是否健全？	健全
7. 发生事故时，司机的驾车动作是什么？	左转弯
8. 进入交叉路口时，汽车的速度是多少 （公里/小时）？	10
9. 事故发生在什么地段？	住宅区
10. 当时的交通是否拥挤？	不
11. 司机做驾车动作（指7）之前在什么车道上？	右车道
12. 当时是否有突如其来的车辆？	没有
13. 是否有安全障？	没有
14. 是否有别的汽车与你同时或在你的前后进入交 叉路口？	有，同时
15. 司机应对事故负责。因为他的判断能力太差。 他转弯时，在他右边有汽车通过。	

图 5-4 有关左转弯事故的对话

通过对话专家系统得出司机应对交通事故负责的结论，因

为他在驾驶过程中判断能力太差，以致转弯时撞上了正在他的右边车道上行驶的车辆。

对于其它事故情节，也可以合成类似的人机对话。

5.5 重新组织或构造

在开发专家系统过程中，如果你想做某方面的扩充，那么就必须重新组织或构造以前建立起来的各种图表，找出上下文树与推理流程图之间的关系，并画出综合两者的上下文树-推理流程图。

5.5.1 重建上下文树

图 5-5 中是一个经过修改的上下文树。此时，系统关心的不再是事故“责任”的归属，而是“司机”本人是否应对事故承担责任。在这里，“司机”是根上下文。“司机”的两个子辈是事故可能发生的两类位置：“道路”和以前没有考虑过的“停车场”。

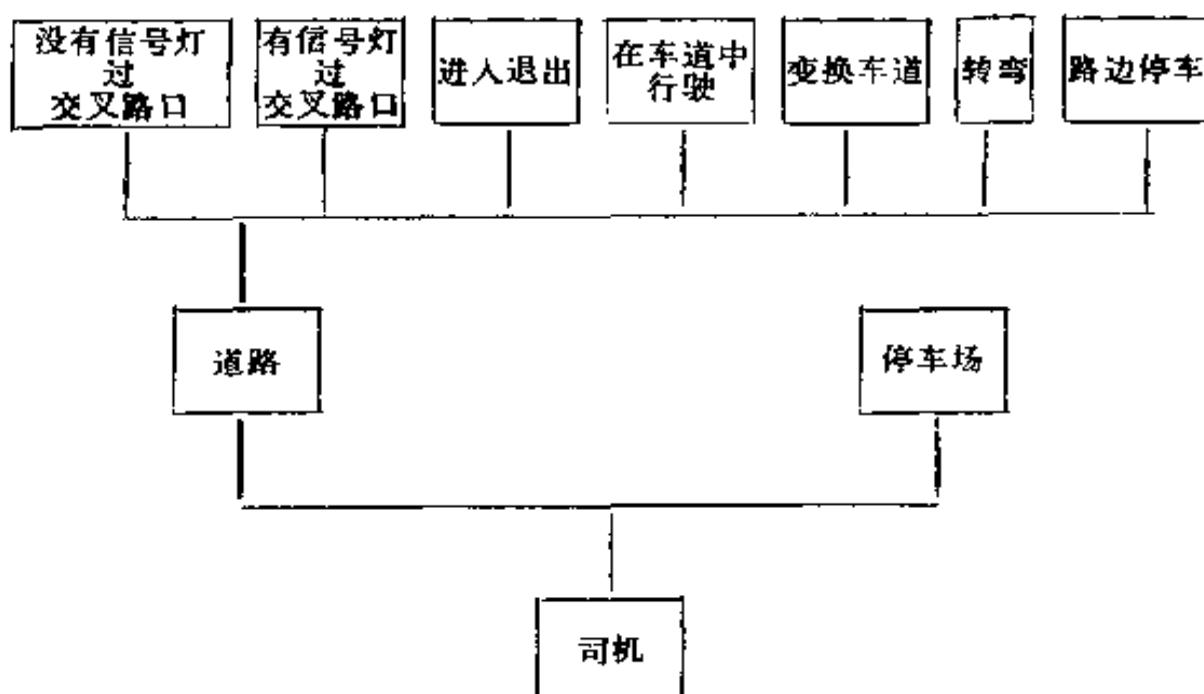


图 5-5 交通事故原型的上下文树 (1)

在“道路”上下文中并不立刻指出具体的出事位置，而是再按使司机遇到麻烦的驾车动作作进一步的分解。因此，交叉路口不再作为一个单独的上下文，而应该用“有信号灯的”和“无信号灯的”两个上下文合起来表示。为了更明白起见，我们最好将它们的名称改成“有信号灯过交叉路口”和“没有信号灯过交叉路口”。其余的驾车动作是：“在车道中行驶”、“变换车道”、“转弯”、“进入退出”和“路边停车”。上述每个上下文都是互相独立的，这是因为当交通事故在其中任何一个上下文中成立时，就绝不会再在其余的上下文中成立了。

5.5.2 重绘推理流程图

由于上下文树改变了，因此，推理流程图在很大程度上也得随之做相应的修改，至少应该使它们互相协调。在图5-6中，就是这样一个经过适当修改的事故原型的推理流程图。

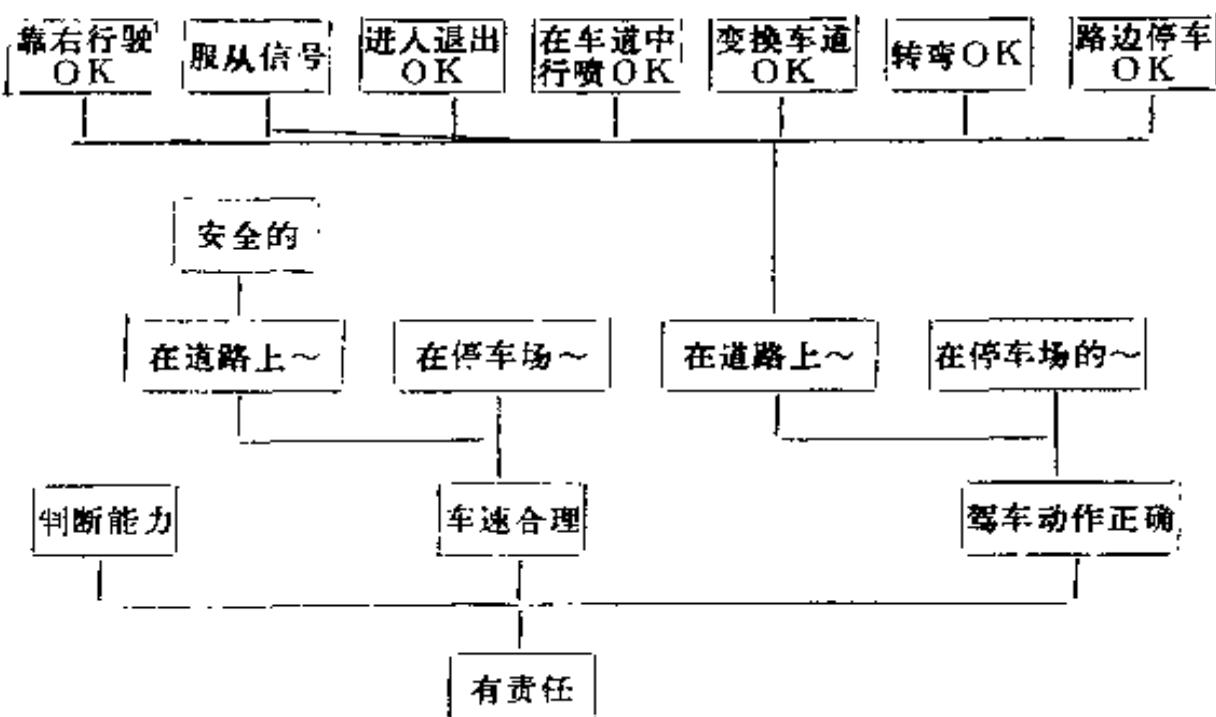


图 5-6 交通事故原型的推理流程图（I）

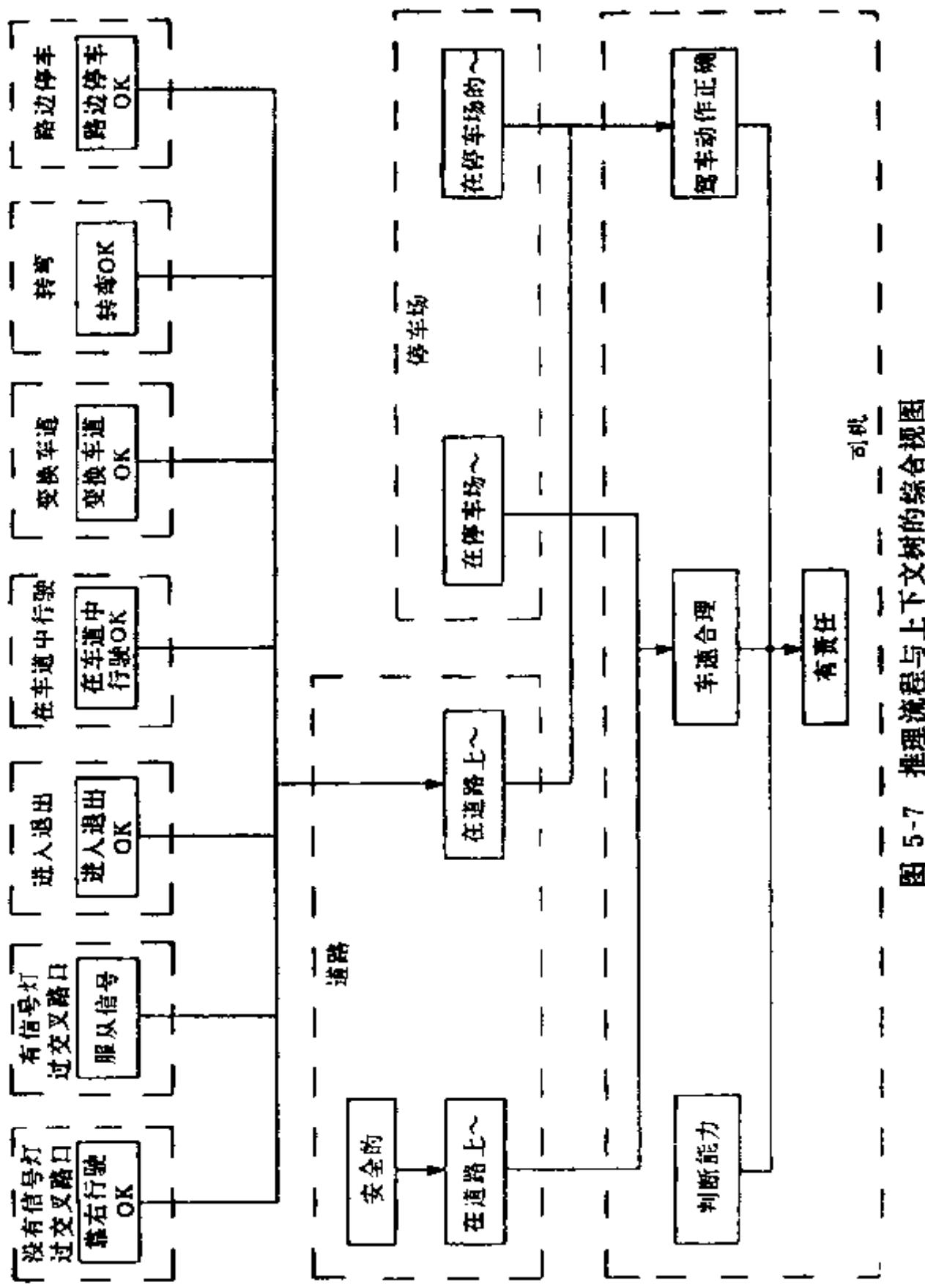


图 5-7 推理流程与上下文树的综合视图

首先，我们将“车速”改成“车速合理”，以强调“实际车速”是我们唯一加以考虑的因素。“车速合理”分别依赖于“道路”上下文中的“在道路上车速合理”和“停车场”上下文中的“在停车场车速合理”。

同样，如图所示，“驾车动作正确”分别依赖于“道路”上下文中的“在道路上驾车动作正确”和“停车场”上下文中的“在停车场的驾车动作正确”两个属性。而“在道路上驾车动作正确”又进一步依赖于“道路”的各子辈上下文中的有关属性。

图5-7是上下文树与推理流程图的综合图示。

应该不断地修改、完善这些图表以及对话，既要避免因为怕麻烦而不愿做必要改动的现象，也要避免因过份地苛求“完善”而举步不前的倾向。因为在阶段Ⅰ中隐藏下来的一些逻辑缺陷只有在后续的开发阶段中才能充分地暴露出来，因此，也只有到那时才能更正这些遗漏或错误。

5.6 确定性能指标

上述步骤都与确定专家系统能够做什么有关，它们确定专家系统领域的范围、所能覆盖的主要子区域，所考虑的关系与概念，以及最终的系统将怎样工作等。作为阶段Ⅰ的最后一步，本步骤的目的是确定你所设想的专家系统能够做到什么程度，也即系统的性能指标。

确定性能指标在很大程度上依赖于你让专家系统完成什么工作，在什么情况下使用它以及由谁来使用它等因素。除此之外，还需要考虑以下问题：由专家系统求解的问题在整个问题中所占的比重有多大？进行一次咨询需要多少时间？咨询建议的可靠性如何？将由谁来使用专家系统？是工作人员？还是顾

客？专家系统将用在什么环境中？是生产？销售？还是维护？等等。

图 5-8 介绍了在汽车交通事故领域中，专家系统一些可能的性能指标。

- 完备性 — 能求解95%的问题
- 有效性 — 在十分钟内完成大多数咨询
- 正确性 — 咨询正确率为90%
- 可维护性 — 在90%的情况下，增加新的规则不会影响已有的规则
- 可用性 — 通过电话就能得到咨询
- 友善性 — 电气工程人员都能够理解和掌握

图 5-8 汽车事故专家系统的性能指标

有一种习惯认为，应该把考虑性能指标的工作推迟到测试知识库的时候，即阶段Ⅳ，这实际上是一种误解。由于性能指标的确定有助于领域的确定，因此这项工作应该放在阶段Ⅰ来完成。当然，这并不是要求你在阶段Ⅰ中确定的性能指标没有任何问题，而是，这些不完善、不合理将在后续的开发阶段中，随着你对知识库有关内容的了解和经验的增加，逐步得到补充和改进。当工程进展到阶段Ⅳ时，你就会得到一个较为理想、较为实际的专家系统的性能指标。

第六章 阶段II: 开发表示 知识的词汇

在阶段I中，你已经使用了许多词汇来描述上下文、推理以及对话。现在，你应该仔细地定义你的知识库中的每一个词汇，特别是每个属性及其合法取值。然后，你才可以编写检验这些词汇的规则。

在阶段II中，主要包括下述步骤：

- 确定属性
- 定义属性的值
- 编写检验词汇的规则

6.1 确定属性

选择和确定属性是一项极讲究方法的工作。在阶段I中，细致地做好这项工作将为后续各开发阶段提供莫大的便利。而采用属性表的方法对这项工作的进行有着明显的帮助。为了填写好属性表，应该尽量使用好的属性。因此，先让我们来看看所谓好的属性究竟应该满足什么样的要求。

6.1.1 属性的要求

属性的选择、命名以及确定必须做到能覆盖所有可观察事实和可推演事实。它们还必须隶属于某个唯一的上下文。具体地讲，就是每个属性都应该具备：

- 有表现力
- 肯定的描述
- 没有冗余
- 隶属于合适的上下文

有表现力 首先，属性的名称应该反映它所表示的意思。这就是为什么选择（司机）“有责任”、“车速合理”和“车速地段”要比选择“事故责任”、“车速”和“车速标志”更合适道理。

肯定的描述 在可能的情况下，应该尽量以肯定的语气描述属性。例如，“车速合理”、“驾车动作正确”、“在道路上驾车动作正确”和“服从信号指挥”就比“车速不合理”、“在道路上驾车动作不正确”和“不服从信号指挥”要好。属性统一都以肯定的语气描述，既便于理解，也使知识库调试变得容易。

没有冗余 往往容易发生这样的现象，即有个属性虽然其名称与前面输入的某个属性不一样，但它们所代表的意义却十分的相近。譬如，你也许会发现，你先定义并输入了一个属性“信号灯”，而后又输入了一个属性“交通信号”，先是“车速限制”、“左转弯”，然后又是“车速地段”和“向左转”。对于这种属性冗余，应该保持高度的警惕。因为这可能引起描述上的混乱。

隶属于合适的上下文 将属性隶属于合适的上下文是至关

重要的，尽管合适的上下文常常不是显而易见的。下面是一些帮助你判断属性归属的原则：如果一个属性在上下文 A 及其子辈上下文 B 中都出现过，则该属性应该放在父辈上下文 A 中，而不应该放在其任一子辈上下文中；如果两个子辈上下文都需要同一个属性，那么该属性应该隶属于这两个上下文共同的父辈上下文。例如，属性“左转弯”、“右转弯”和“掉头”似乎应该同时隶属于下列上下文：“有信号灯过交叉路口”、“没有信号灯过交叉路口”和“转弯”。在这里，最好的解决办法就是把这些属性统统归入上述上下文共同的父辈上下文，即“道路”。

另一种表述同一概念的方法是：在两个独立的上下文中分别使用不相同的属性名称。假如你在上下文“道路”中使用了“实际车速”和“安全的”，则不要再到上下文“停车场”中使用同样的名称，而不妨使用诸如“在停车场的实际车速”和“停车场的安全性”这样的名称。

6.1.2 属性表

图6-1中是一张属性表，它显示了属性及其所属上下文，属性是可观察的还是可推演的，以及它们的自然语言翻译等内容。

可以借助推理流程图的帮助来完善该属性表。为了方便、准确起见，我们画一张如图6-2所示的草图，并参照它来选择合适的属性名。然后，将每个属性归入合适的上下文，确定每个属性是可观察的还是可推演的；最后，还有一个需要说明的专家系统处理细节，这就是给每个属性一个一般用户能够理解的自然语言翻译。

上下文	属性	可观察 /可推演	翻译
司机 在道路 上	有责任	推	司机应承担事故的责任
	判断能力	推	司机的判断能力
	车速合理	推	司机以合理车速行驶
	有警觉性	观	司机具有驾驶机动车所必需的警觉性
	喝醉酒的	观	司机已经喝醉了酒
	烦躁的	观	司机情绪烦躁影响驾驶
	滥用兴奋剂	观	兴奋剂影响司机的驾驶
	在道路上驾车 动作正确	推	司机在道路上的驾驶动作正确
	在道路上 车速合理	推	司机在这种路况下的车速合理
	安全的	推	行车条件是安全的
路 况 及 规 则 上	机械安全	观	汽车的机械性能好能安全行驶
	道路安全	观	路况良好能安全行车
	实际车速	观	汽车的实际车速
	车距	观	汽车与前面车辆的距离
	车速限制	观	允许的车速
	左转弯	观	司机做了左转弯动作
	右转弯	观	司机做了右转弯动作
	掉头	观	司机做了“掉头”动作
	横穿	观	司机横穿交叉路口
	反向车流	观	有与司机行进方向相反的车流
没有信号 灯过交叉 路口	信号正确	观	司机发出的信号正确
	服从标志	观	司机服从交通标志
	靠右行驶OK	推	司机遵守靠右行驶的交通规则

按按钮 进入	让车	观	司机为横向车流让车
	让车标志 进入的时间	观 观	在交叉路口有让车标志 相对其它车辆司机进入交叉路口的时间
有信号灯 过交叉路口	服从信号	推	司机服从交通指挥信号
	交通拥挤 交通信号	观 观	交叉路口的交通非常拥挤 交通信号的颜色
进入退出	进入退出OK	推	司机能正确地进入或退出道路
	上路 退出 车道开放	观 观 观	司机进入道路 司机退出道路 有允许通过的车道
在车道中 行驶	在车道中行驶 OK	推	司机在其车道中合法行驶
	车身距离 双右车道 拖挂 前车拖挂	观 观 观 观	以车身长度为单位，司机与前面车辆的距离 司机在两条右车道中的一条行驶 司机的车拖挂了另一辆车 前面的车拖挂了一辆车
变换车道	变换车道OK	推	司机正确地变换车道或超车
	超车 变换车道 迂回行进	观 观 观	司机超越前面的车辆 司机变换车道 司机在车道之间迂回行进
转弯	转弯OK	推	司机转弯动作正确

路边停车	路边停车OK	推	司机能在路边正确地停车
	路边空地 进入空地	观 观	在路边有供停车的空地 司机驾车驶入路边的空地
停车场	在停车场车速合理	推	司机在停车场上的车速合理
	在停车场的驾车动作正确	推	司机在停车场上的驾车动作正确
停车场	在停车场的驾车动作	观	司机在停车场上实施的驾车动作
	在停车场的实际车速	观	司机在停车场的实际车速
	停车安全	观	停车场的行车条件是安全的
	方向正确	观	司机行车的方向正确

图 6-1 交通事故原型的属性表

6.2 定义属性的值

选定各个属性后，就可以借助属性-值表来确定它们各自的取值类型和范围了。在进行此项工作之前，有必要先了解一下属性值的一般要求。

6.2.1 值的要求

对属性值的主要要求可以从属性-值表中得出，在图6-3中，给出了一张属性-值表。从表中我们可以看出有以下基本要求：

预定义值

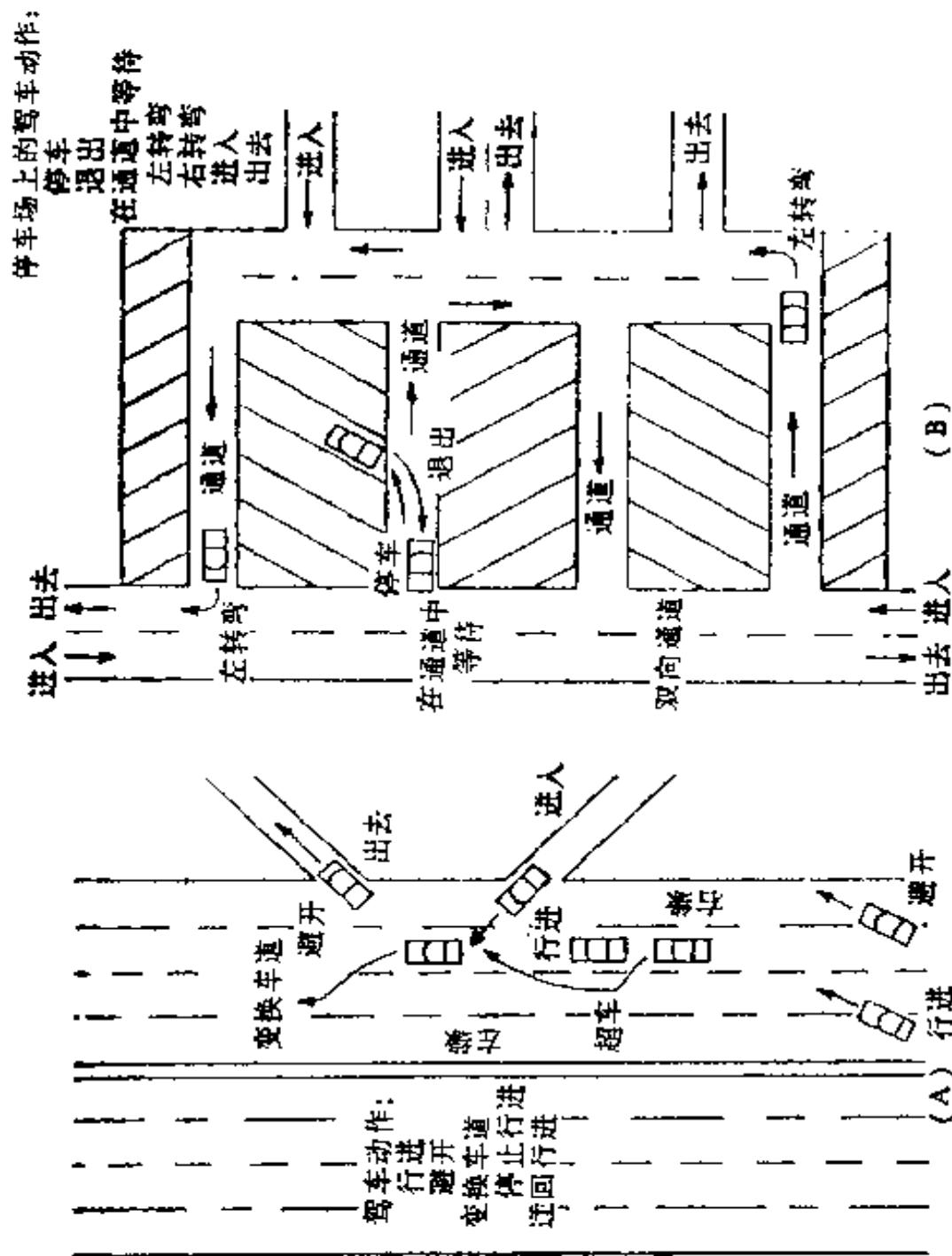


图 6-2 词汇的图示

- 使用是非型值
- 定义有值属性的值表
- 定义数值属性的单位和范围

预定义值 预定义就是为属性事先定义尽可能多的值。这种预定义使规则的书写，修改和调试变得容易，也使得逻辑清晰。

使用是非型值 取值为“是”(YES)或“不是”(NOT)的属性是最容易被人理解的。这种属性也为专家系统和用户提供了最简单的接口。在交通事故原型领域中，“有责任”、“车速合理”和“驾车动作正确”都是是非型属性。

定义有值属性的值 有时，你所喜欢使用的是非型属性会产生丢失信息的现象。例如，可以把“判断能力”设计成是非型属性，即司机有判断能力或没有判断能力。然而，为了使之能恰如其分地表示现实，最好的办法是为它设置一组可能的值，如“很好”、“一般”和“很差”。

“判断能力”被称作单值属性，因为在任何时刻属性“判断能力”只能取上述三个值中的一个。如果一个人的“判断能力”是“很好”，那就表示他的“判断能力”既不是“一般”，也不是“很差”。与此相反，多值属性却可以同时有多个值。

现举一个交通事故原型以外的例子，在风险投资知识库中有一个属性是“风险投资类型”。当专家系统向用户询问他感兴趣的“风险投资类型”是什么时，用户就可以从下述值中选择若干个值来回答：“债券”、“股票”、“存款凭证”、“金融基金”。多值属性的另一个例子是“学历”，在同一时刻它可以取若干个值，如“小学”、“初中”、“高中”、“大学本科”和“研究生院”等。

最后，一个属性应该设计成是非型属性还是让它取一组值，这需要视具体情况而定。问题并不在于你可以设想多少个值，而在于描述你的知识时，为了编写出清晰明了的规则，究竟需要多少个值。

定义数值属性的单位和范围 有时，你无法事先确定属性的取值，然而，你可以通过说明属性的合法取值范围来把握问题。你可以说明合法的值是数值型、文字型还是文字数值型（既可取文字型值也可取数值型值）。

如果属性是数值型的，你就应该说明其取值单位，例如是千米、米、厘米还是摄氏度等，以及允许的取值范围。例如，虽然“车速”是数值型属性，但是如果让“车速”大到500(公里/小时)就不甚合理了。

6.2.2 属性-值表

属性-值表

上下文：司机	预定义				非预定义				
	属 性	是非型	单值	多值	可能取值	文字	数值	单位	范围
有责任		√							
车速合理		√							
判断能力			√		很好 一般 很差				
驾车动作正确		√							
有警觉性		√							
滥用兴奋剂		√							
喝醉酒的		√							
烦躁的		√							

图 6-3 交通事故原型的属性-值表

图 6-3 中的属性-值表列举了上下文“司机”中的全部属性。用类似的表格可以表述交通事故原型的每个上下文中的属性。

6.3 编写检验词汇的规则

检验用上述方法开发出来的词汇是否有效的一个最好办法就是，直接用你的这些属性来编写规则。在编写规则的过程中，你会逐步了解这些属性是否具有表现力，如果没有，可以改变相应属性的名称、取值或翻译。你也许会发现冗余的属性，这可以通过改变其中一些属性的名称或意义来解决。你还可能发现属性间存在遗漏，增加一些属性可弥补这些逻辑漏洞。

有些属性的表述从字面上看似乎是肯定的，而在应用上却不然。例如，在图 6-4 A 中有一个由属性“有警觉性”、“喝醉酒的”、“滥用兴奋剂”和“烦躁的”表示有“判断能力”的规则，请注意在这条规则的如果部分中，除“有警觉性”一个属性是肯定形式的以外，其余的属性都是否定形式的。假若所有的属性相对于结论有“判断能力”来说都是肯定形式的，那不是在逻辑上更直观、更容易被理解吗？为此，我们可以将

有警觉性
非 喝醉酒的
非 滥用兴奋剂
非 烦躁的
判断能力 = 很好

A. 含否定形式的属性

有警觉性
清醒的
未服兴奋剂
镇静的
判断能力 = 很好

B. 全部是肯定形式的属性

图 6-4 用于检验词汇的规则

“喝醉酒的”改为“清醒的”，“滥用兴奋剂”改成“未服兴奋剂”、“烦躁的”改成“镇静的”，这样一来就可以简化规则的表示，正如图6-4 B所示的那样。

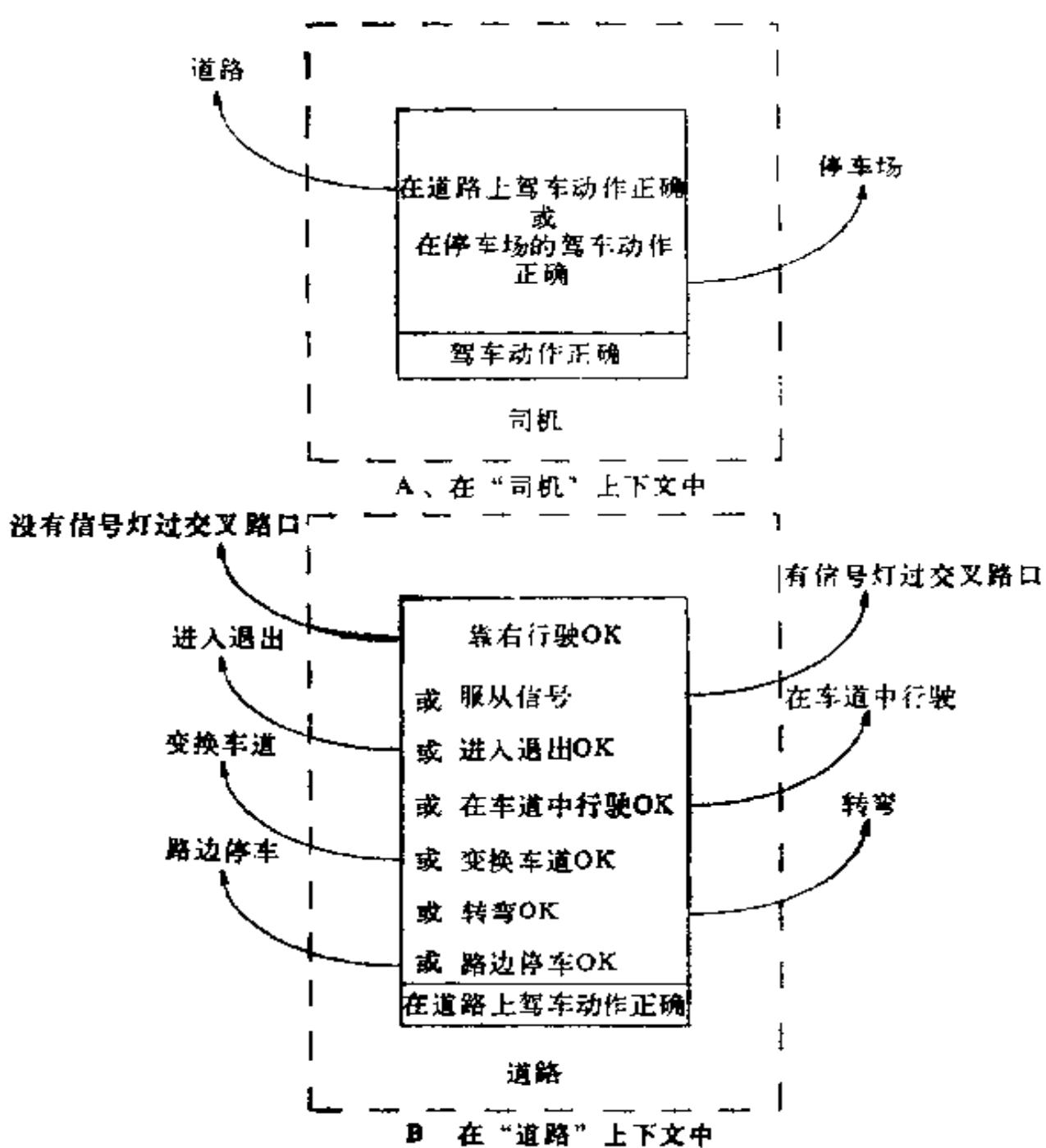


图 6-5 检验上下文干扰的规则

有时，有些规则会将一个上下文中的属性与另一个上下文中的属性联系起来。在图 6-5 中给出的正是这样两条规则。其中第一条规则属于上下文“司机”，它表示如果“在道路上驾车动作正确”为真或“在停车场的驾车动作正确”为真，则“驾车动作正确”为真。换句话说就是，如果司机在道路上或在停车场上的驾车动作合理，则他的驾驶动作合理。

尽管此规则处在上下文“司机”中，但它的每个如果子句都参照了另一个上下文：“在道路上驾车动作正确”参照了“道路”上下文，而“在停车场的驾车动作正确”参照了“停车场”上下文。然而，这并不意味着上述两个属性应如同图 6-1 所示的那样分别隶属于上下文“道路”和“停车场”。如前所述，一个被两个相关的上下文都引用了的属性应该归入它们的父辈上下文中。因此，在属性表中上述两个属性都将归入“司机”上下文中。

属性的隶属关系不正确的另一个例子如图 6-5B 所示，显然，该规则应该属于“道路”上下文。然而，它的每个如果子句参照了另一个上下文。同样，也不能因此而将这些属性归入“道路”的各子辈上下文。这些如果子句中的属性都应该归入上下文“道路”。

修改你的属性表以反映上述变化。

第七章 专家系统的推理过程

从本章开始，我们打算用两章的篇幅暂时中断一下对开发交通事故原型专家系统的叙述。本章介绍由计算机执行的推理过程，下一章则介绍“电脑顾问”这种专家系统开发工具的一些具体特性。之所以要先介绍上述两项内容，这是因为开发专家系统是一项需要你与专家系统开发工具相互配合的智力工程，对于专家系统所执行的每一个推理步骤，你都应该有深入透彻的理解。而这种理解对于取得开发专家系统的阶段Ⅲ和阶段Ⅳ的成功是必不可少的。

机器推理的过程就是计算机从一个规则到另一个规则跟踪属性的过程。如前所述，沿着从可观察事实到目标的方向跟踪规则的过程被称作正向推理，而从目标到可观察事实的过程被称作反向推理。归纳推理通常是用正向推理方式对输入的事例进行概括的。在这一章里，我们将详细讨论正向推理和反向推理。

此外，事实以及由规则推出的结论的不确定性也会对推理过程产生影响。

7.1 逻辑的模拟

规则库模拟从事实到目标或从目标到事实的推理逻辑。由于规则库中可能包含大量的规则，因此，需要一种能将规则按不同特征分组，并能把这些规则组互相联系起来的方法。在这

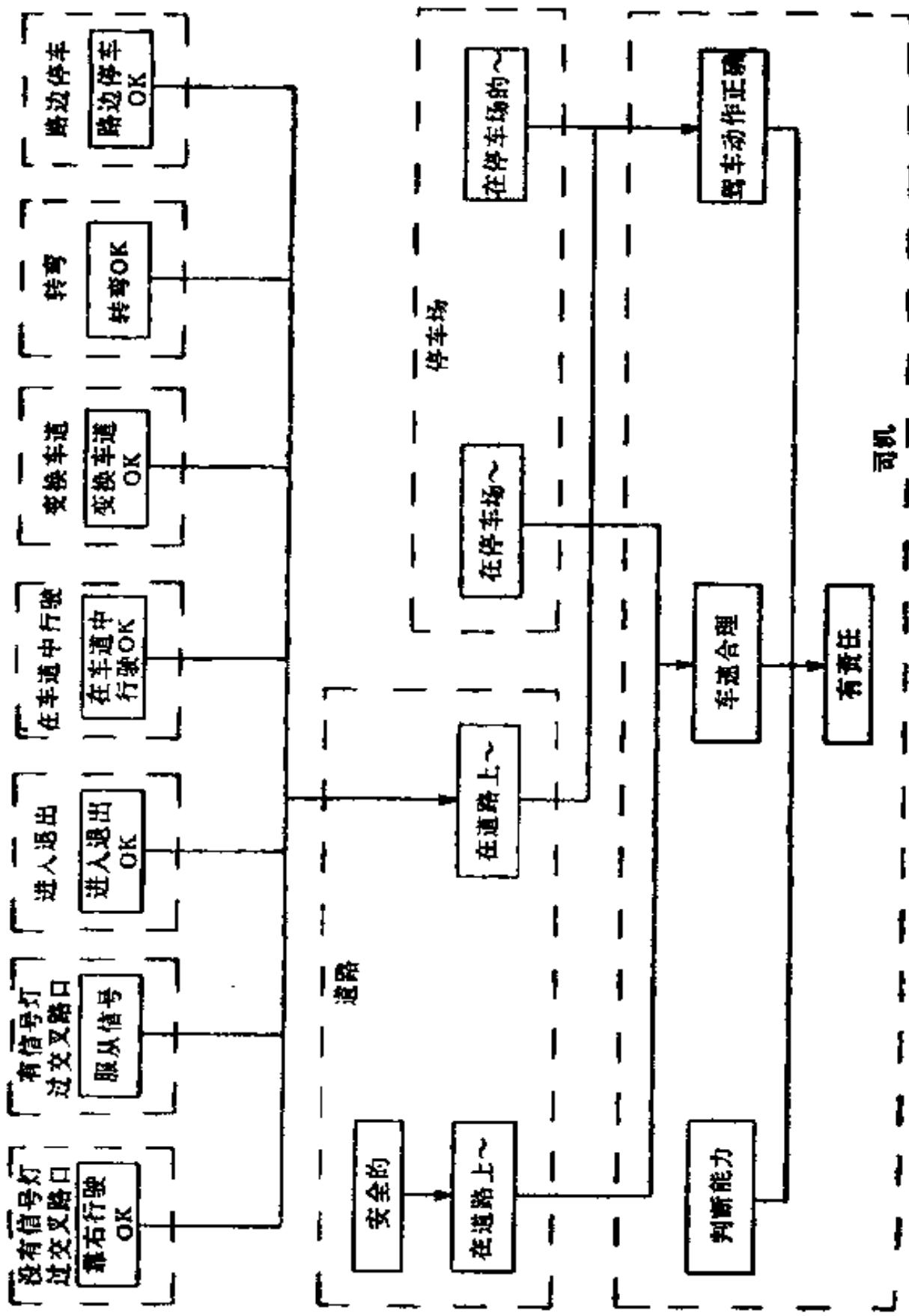


图 7-1 规则库图

里，我们给这种规则组以一个专门的术语，称作规则簇，使规则簇互相联系的结构是规则库图。

7.1.1 规则库图

在第五章的图5-7中，将推理流程与上下文树结合而形成的图示就是规则库图，因此我们在图7-1中重新给出了这幅图。如果你把规则的则部分中的属性名称作为相应的规则簇的名称的话，那么，图7-1反映的就正好是交通事故原型规则库中所有规则的一个概略图解。

7.1.2 规则簇

直接推导一特定属性结论的所有规则组成一个规则簇。在交通事故原型中，所有直接推导关于属性“有责任”的结论的规则都属于名称为“有责任”的规则簇（见图7-2）。

属性“有责任”依赖于另外三个属性：“判断能力”、“车速合理”和“驾车动作正确”。所有能够判断司机是否应对交通事故负“有责任”的条件或假设都被包含在规则R1, R2和R3的如果部分中。R1欲判断“司机”对交通事故没有责任，它要求下列所有条件都成立：“判断能力” = “很好”，“车速合理”且“驾车动作正确”。R2表示如果下列条件中的任何一个成立：“判断能力” = “很差”，非“车速合理”或非“驾车动作正确”，则“司机”应对事故负完全(100%)责任。R3则表示如果“判断能力” = “一般”，则司机对事故负有责任，但这个结论的可信因子CF只有80%。

你自己还可以给出影响属性“有责任”取值的其它条件组合，这只需对现有规则进行修改或加入新规则到规则簇中即可。

为了便于表示规则簇之间的相互联系，我们把影响一个规则簇的可推演属性写在规则簇图顶部的矩形框中。如在图7-2

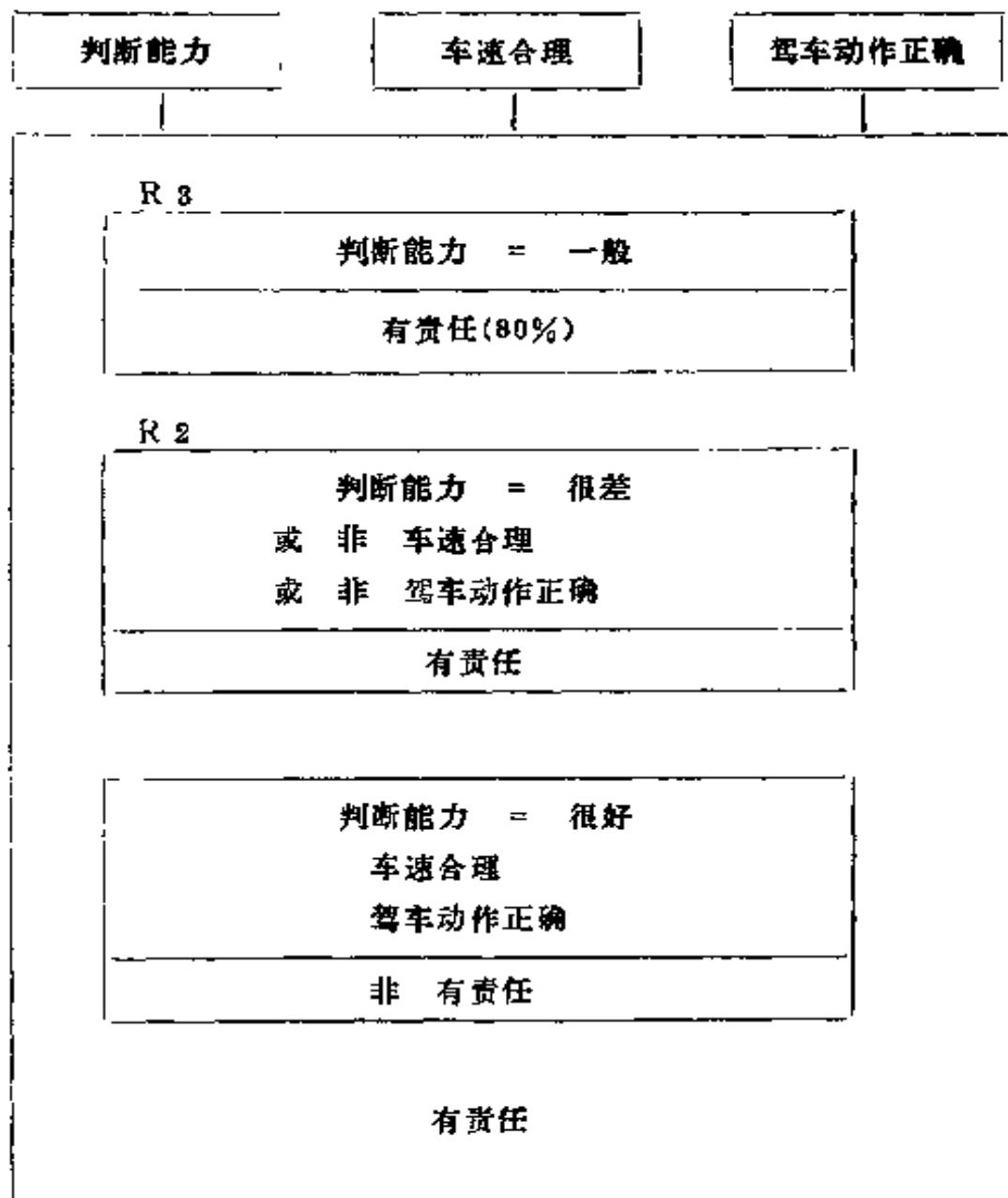


图 7-2 “有责任”规则簇

中，这些可推演属性分别是“判断能力”、“车速合理”和“驾车动作正确”。它们各自又都代替不同的规则簇。

图7-3所示的是规则簇“判断能力”，它包含R4、R5、R6和R7四条规则，并由这些规则决定在什么样的条件下属性“判断能力”分别取值“很好”、“一般”或“很差”。

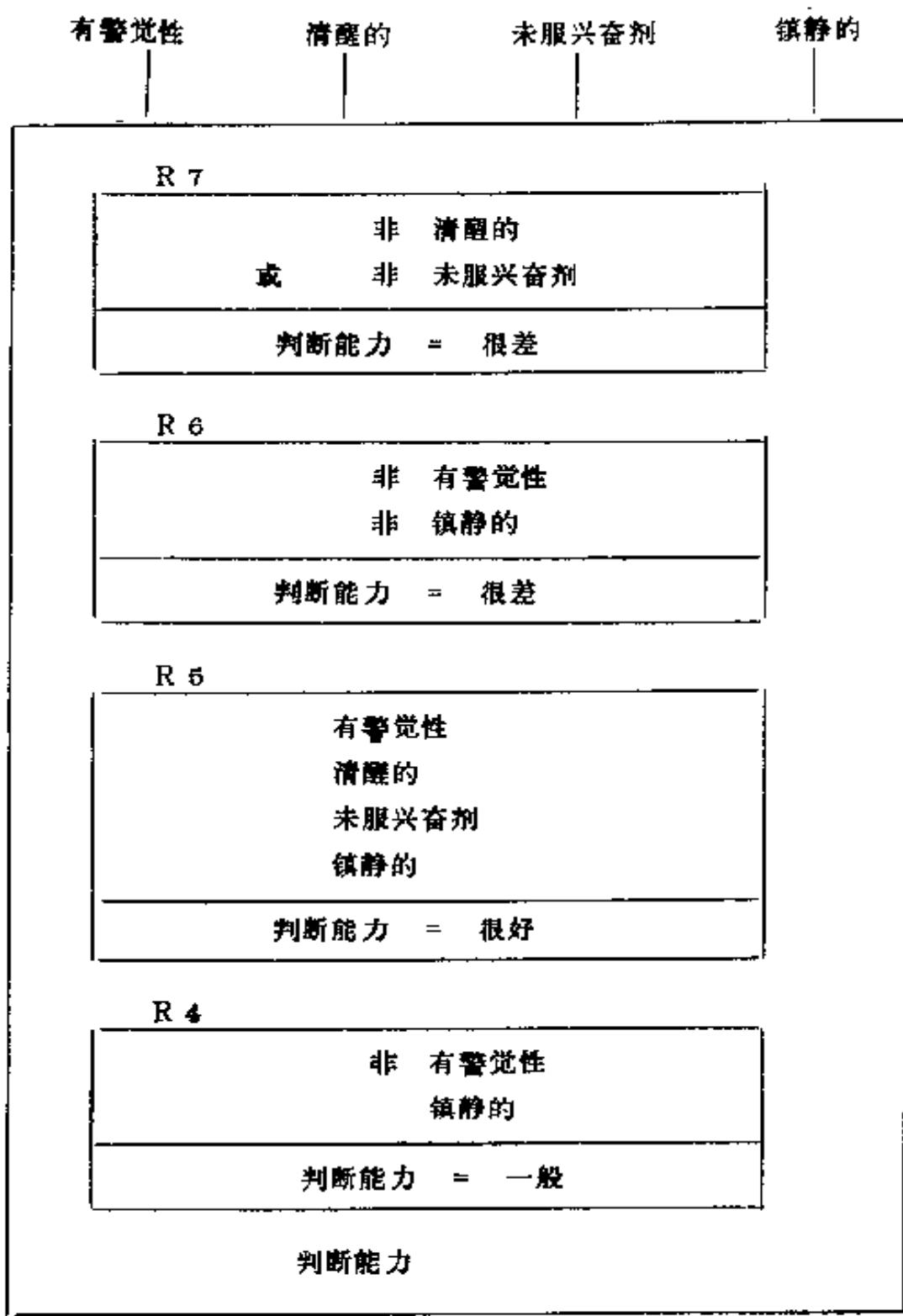


图 7-3 “判断能力”规则簇

规则簇“判断能力”中的规则依赖于四个属性：“有警觉性”、“清醒的”、“未服兴奋剂”和“镇静的”。由于这些属性都被看作是可观察的，因此，在“判断能力”规则簇的顶部不再需要用于与其它规则簇联系的矩形图案。

7.2 正向推理

正向推理从可观察事实出发，经过一层或多层规则的推演路径，最后到达关于目标的结论。正向推理能够选择实在的或概念的构件，并把它们结合起来以形成产品、组织或系统。对于需要综合的问题，正向推理尤其适用。事实上，美国的数据设备公司（DEC）用来配置 VAX 计算机系统的著名专家系统 XCON/R1就主要使用正向推理技术。

7.2.1 正向推理过程

正向推理的完整过程大致是这样的：将已知事实与规则的如果部分中的子句进行比较，从中找出匹配。如果找到一个与某规则的每个如果子句都匹配的情况，则该规则被激活，从而导致该规则的则部分被执行。对其它事实可以同样重复此过程，并由被激活的规则产生新的可推演事实。上述过程如此往复地不断进行下去，直到产生欲达到的结论或所有匹配条件均已用完时为止。

7.2.2 正向推理的例子

图7-4给出了一个正向推理的例子，其中事件的发生顺序用圆圈中的数字标记。一开始，有四个事实被依次输入，它们是：“有警觉性”①、“清醒的”②、“未服兴奋剂”③和“镇静的”④。这些事实正好与规则 R5 的各如果子句形成匹配，因此，规则 R5 被激活并产生可推演事实“判断能力” = “很好”⑤，这个新产生的事实又可作用于 R1，但由于 R1 的其它

条件没有相应的数据，因此，无法激活它。

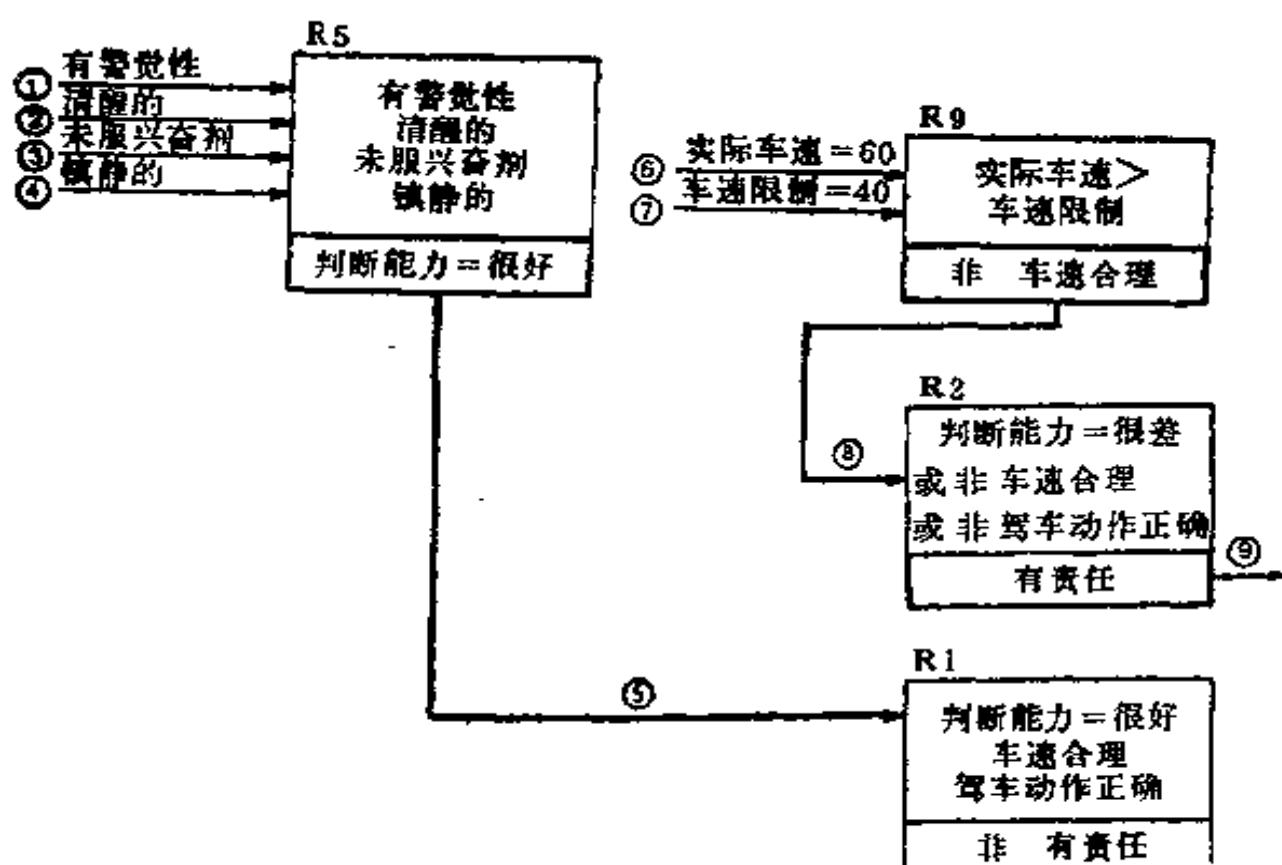


图 7-4 正向推理的例子

此后，事实⑥和⑦被输入：“实际车速” = 60 和 “车速限制” = 40，并发现它们与 R9 的如果部分匹配，因此规则 R9 被激活并产生结论——非“车速合理”。当这个新事实作用于 R2 ⑧时，形成匹配，因此激活规则 R2 并产生结论——司机对交通事故“有责任” ⑨。

7.3 反向推理

反向推理则从目标出发，沿着推理路径追溯到事实。它从一般性开始，然后逐步涉及细节，即它是通过求解较小的子辈问题达到求解较大问题的目标的。反向推理通过收集越来越详

细的证据以求证实一种情况或假设。反向推理特别适合作分析，这可能就是为什么 MYCIN 选用它来进行医疗诊断的原因。

7.3.1 反向推理过程

反向推理与正向推理相似，都是先匹配后执行，所不同的是方向相反。在反向推理过程中，一开始是用目标与规则的则子句进行匹配，而不是一开始就用已知事实与规则的如果子句匹配。当找到一条通向某条规则的路径，而此规则的所有如果子句由于均能与可观察事实匹配而成立时，则激活该规则，并沿着搜索经过的道路向回（即以正向）报告成功，直至达到目标。

7.3.2 反向推理的例子

我们借一个例子来说明在反向推理过程中各事件发生的先后次序（见图7-5）。在此例中，所示的规则来自三个不同的规则簇，即“有责任”、“判断能力”和“车速合理”。

首先，系统从目标出发，考查“有责任”规则簇中的第一条规则 R1 的如果子句。由于其中第一个属性是“判断能力”，因此，系统反向跟踪到“判断能力”规则簇，并考查其中第一条规则 R4 的第一个如果子句①。由于该如果子句是关于可观察事实的，因此，通过提问得到有关事实：非“有警觉性”② 和“镇静的”③。此时 R4 的条件与事实匹配，因此产生结论：“判断能力” = “一般”。

接下来，将新事实“判断能力” = “一般”回送到 R1④。由于它不能与 R1 匹配，所以系统转而尝试簇中的下一条规则，即 R2。

由于 R2 的如果部分的第一个条件不成立，因此系统接着考查下一个条件。为了获得该条件中的属性“车速合理”的取

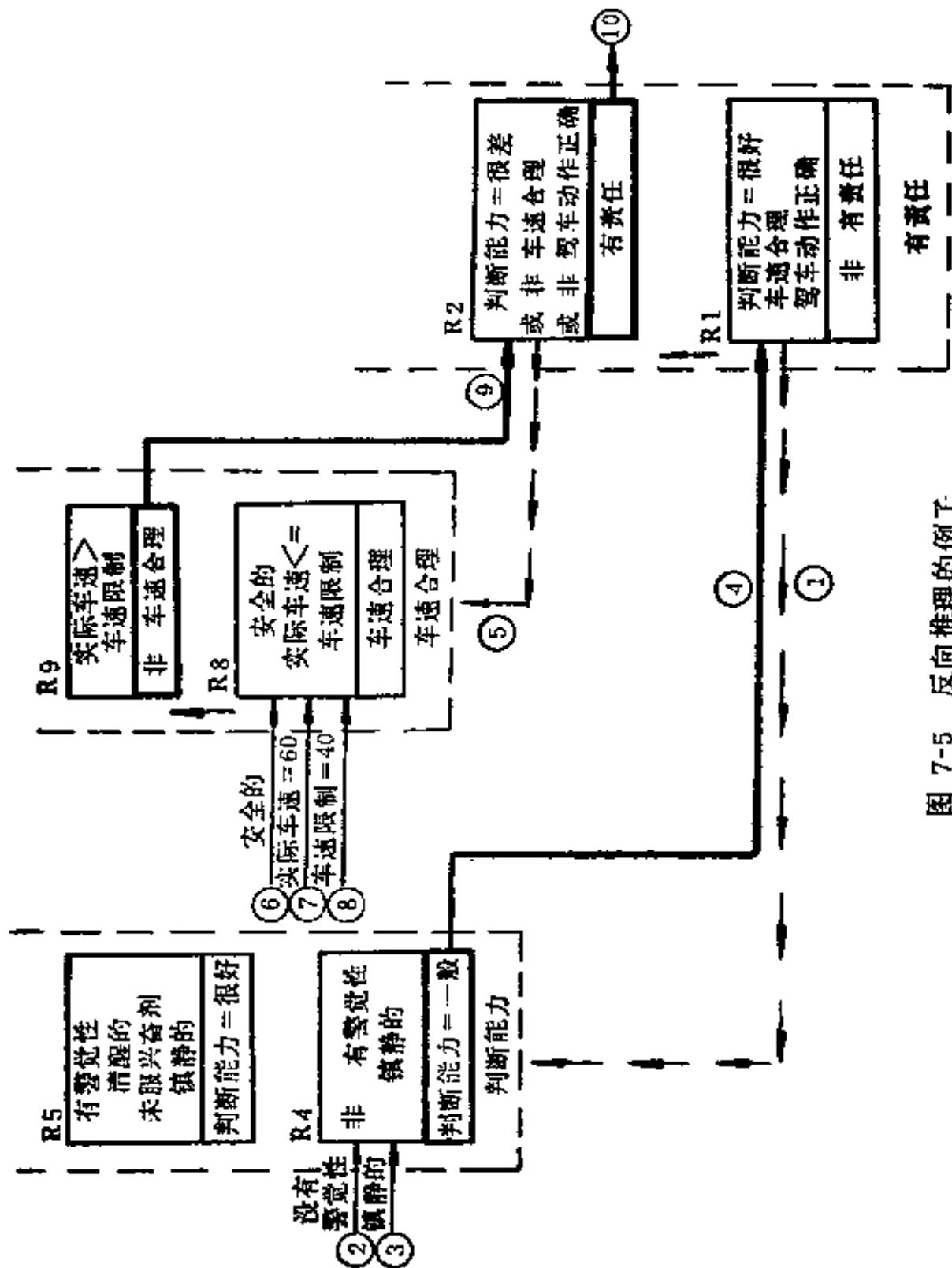


图 7-5 反向推理的例子

值，系统反向进入“车速合理”规则簇⑤。为了检查该规则簇中的第一条规则，即 R 8，是否存在匹配，系统需要确认以下事实：“安全的”⑥，“实际车速” = 60⑦和“车速限制” = 40⑧。然而，十分遗憾，匹配不存在。

由于匹配不存在，所以系统继续考查该规则簇中的下一条规则，即 R 9。由于 R 9 能与现有的数据匹配，因此，该规则被激活并且产生结论：非“车速合理”。

当结论非“车速合理”回送到 R 2 时⑨，R 2 被激活，因此，产生结论——司机对交通事故“有责任”⑩。

7.4 前件规则与后件规则

规则的如果部分中的子句也常称作“前件”，相应地，则部分中的子句常称作“后件”。正向推理也称作“前件驱动”的推理；而反向推理则称作“后件驱动”的推理。

也有把规则称为“前件规则”和“后件规则”的。因此，在正向推理中使用的规则被称为“前件规则”，而那些在反向推理中用到的规则被称作“后件规则”。

在有些反向推理系统中，也提供了一些规则进行正向推理，其中“电脑顾问”就属此列。在这类系统中，用于正向推理的规则被特别标记为前件规则。

7.5 元规则

有这样一类事物，它们是关于学习的学习，有关思维的思维，甚至是规则的规则。例如，有这样的规则，它说明在什么情况下应该选用这组规则或另一组规则，这种规则，我们称之为“元规则”。

元规则支持对领域中的问题进行深层推理。元规则提供了

使用规则的策略知识，后者是表示领域问题求解技术的，为了以示区别我们称它为“应用规则”。元规则还可用来确定子问题的求解次序或分解问题的方式。

在小型专家系统中，也许不需要元规则。但在大型专家系统中，元规则的使用可以使系统在推理过程中必须匹配的规则数目显著地减少。

图 7-6 给出了一个元规则的例子。该规则告诉系统如果车祸比较严重，而且在规则库中既有其如果部分中包含属性“实际车速”的规则，也有其如果部分中不出现该属性的规则，那么系统应该先执行前一种规则，然后执行后一种规则。

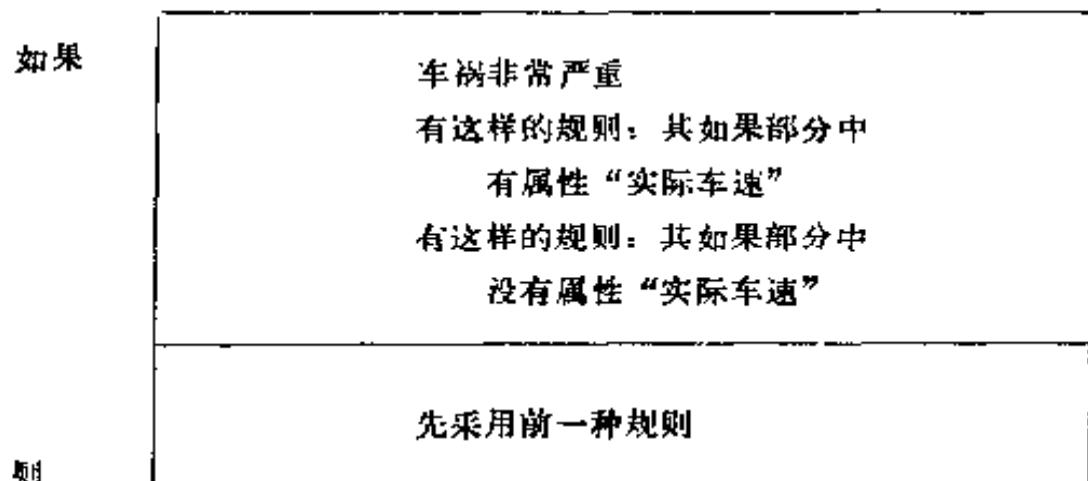


图 7-6 一条元规则

7.6 不确定性

在许多实际应用中，有些事实无法以确定的方式表述，对于规则也有类似的情况。当推理路径中出现若干个这样的不确定事实或规则时，最终获得的结论也许会比此推理中引用的任何规则的确定性都小。

7.6.1 不确定事实

可信因子 (CF) 以百分比的形式来说明事实不确定的程度。

度，其值可正可负。因此，当你对一事实的成立只有六成相信或把握时，就可以给它一个 +60% 的 CF；相应地，当你对事实的否定有六成的相信时，你可以给该事实一个 -60% 的 CF.

7.6.2 不确定规则

CF 也可以用 来说明与规则有关的不确定性。通常是通过给规则的则子句附加一个 CF 来实现这种说明的。如果在规则中存在多个则子句，则可以给每个则子句附加一个 CF 值。

7.6.3 合成事实与规则的不确定性

一条规则的如果部分和则部分的 CF 决定了该规则输出结论的 CF。这个 CF 的获得大致上是经过了如下两个步骤：

- 计算如果部分的 CF
- 合成如果部分和则部分的 CF

计算如果部分的 CF 如果部分 CF 的有效值是由各如果子句的 CF 值以及联结它们的逻辑联结词综合决定的。假设各如果子句由“与”联结，则各如果子句 CF 值的最小值就是该如果部分 CF 的有效值；而当如果子句是由“或”联结时，各如果子句 CF 的最大值就是该如果部分 CF 的有效值。

在图7-7中，R1的如果子句是由“与”联系的，因此，其如果部分 CF 的有效值为各如果子句 CF 的最小值——60%。而在 R2 中，如果子句是由“或”联结的，因此，其如果部分 CF 的有效值为各如果子句 CF 的最大值——80%。

合成如果部分和则部分的 CF 运用简单的算术乘法可以将如果部分的 CF 和则部分的 CF 合成为一个数值，这个数值就是规则结论的 CF 值。因此，R1输出的 CF 是 60% 乘以 100%，即 60%；而 R2 则输出 80% 乘以 90%，即 72%；类似地，R3 输出的

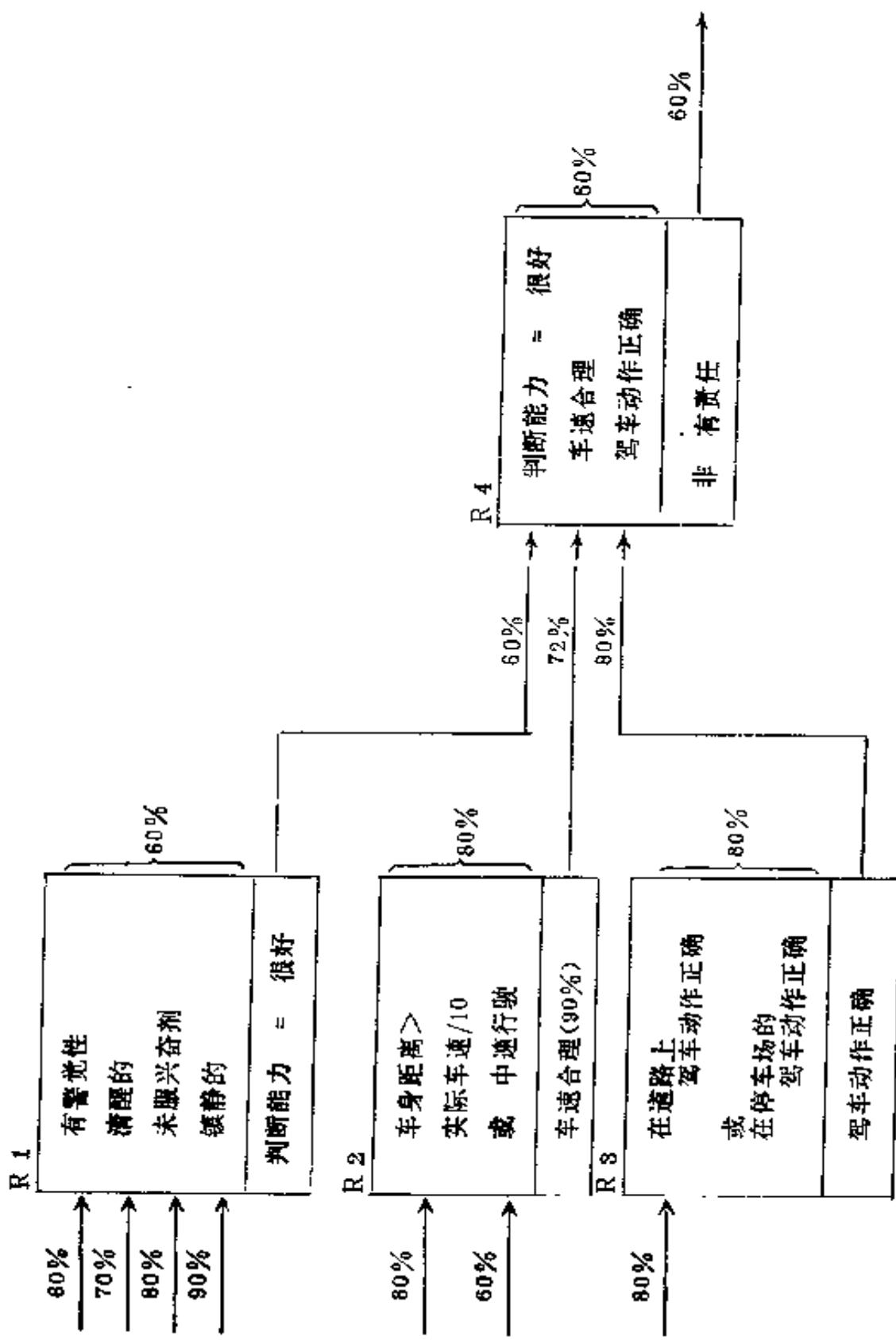


图 7-7 合成可信因子

CF是80%.

上述各规则输出的CF都成了R4的输入 CF，在该规则中，还将以同样的方式继续合成CF。在这里，由于规则的如果子句是用“与”联结起来的，因此，60%——最小的CF——就是如果部分的CF。又由于则部分的CF是100%，因而规则输出结论的CF就是60%。该值的意思是只有六成的把握确信司机对交通事故没“有责任”。

以上是包括“电脑顾问”在内的专家系统合成CF的方法。当然，还有许多其它的方法，由于篇幅的关系，我们不在此赘述。

第八章 专家系统开发工具—— 电脑顾问

目前，在软件市场上已经出现了若干种软件产品，它们能够实现或部分实现我们在上一章中所介绍的推理技术。可以理解，各种系统虽然完成的任务相同，但采用的方法可能不尽一样。然而，从宏观上看，这些细节上的具体差异是无关紧要的，而且，过份地强调这些细节只会陷入思想上的混乱。为了避免这种情况，我们特选择“电脑顾问”作为这类专家系统开发工具的代表，并加以叙述。

在这一章里，我们并不打算介绍“电脑顾问”的所有语言特征以及怎样使用其命令等具体细节，这些内容在软件的用户手册中有详尽的介绍。我们认为应该强调，并需要着重介绍的内容是建立和检验知识库的概念、方法和技术。

8.1 概 述

“电脑顾问”是一种在EMYCIN基础上发展起来的专家系统开发工具，其中后者实际上是 MYCIN 的“骨架”，MYCIN 是一个用于细菌感染诊断的专家系统，它是由参加过斯坦福大学“斯坦福启发式程序开发项目”的科学家布鲁斯·巴克兰和爱德华·肖特利夫等人研制的原型系统。

“电脑顾问”主要由三部分组成（见图 8-1）：开发机和推理机，这两者组成了专家系统开发工具，还有一些独立的执

行时实用程序（Run-Time Utilities）。开发机帮助专家建立知识库并对其进行调试。执行时实用程序的作用是将完整的知识库与一个推理机副本结合起来，并把它们一同复制到另一张磁盘上去，从而产生一个可以直接使用的完整的专家系统。

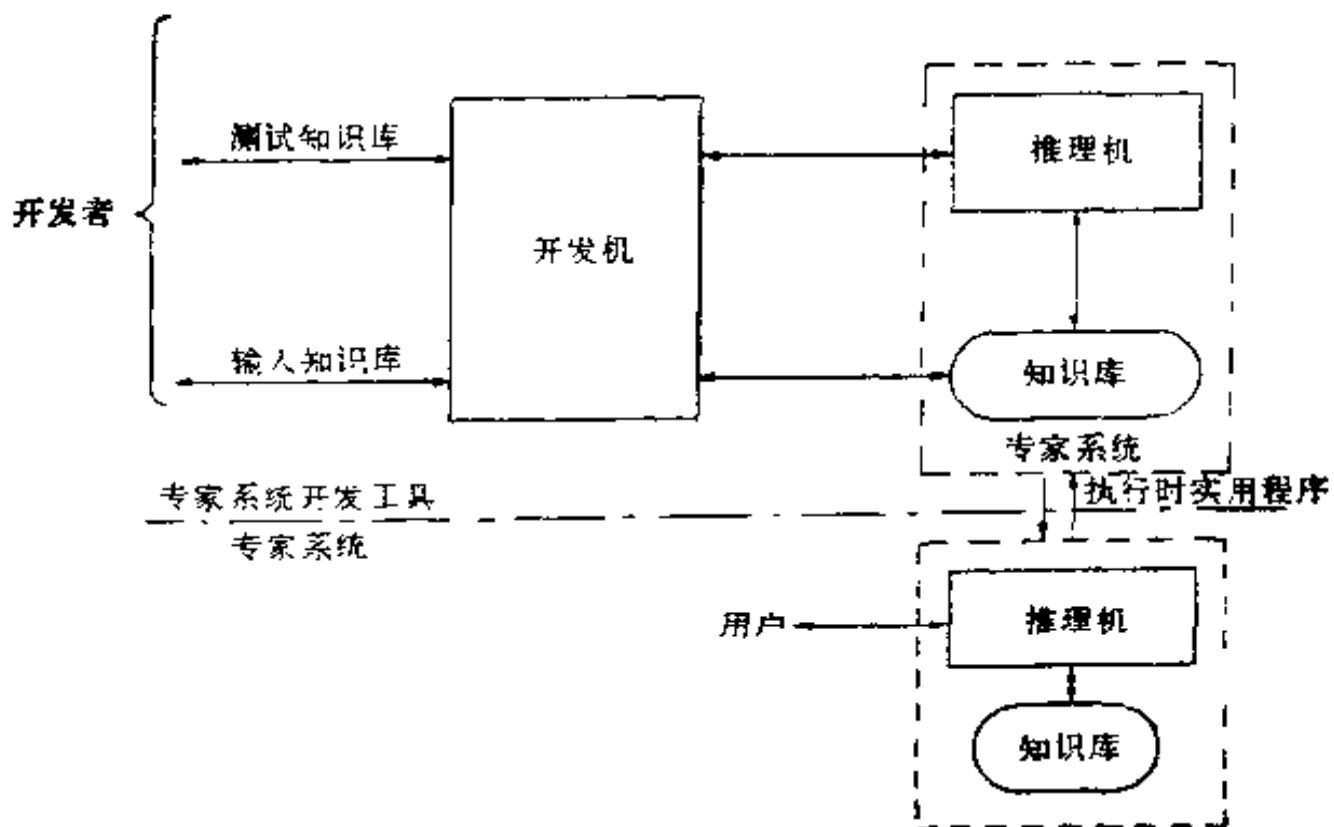


图 8-1 “电脑顾问”结构简图

“电脑顾问”对IBM PC及其兼容机的基本硬件要求是：640K字节的内存空间和一个温彻斯特硬磁盘驱动器。如果内存再大些，效果会更好。它同时还支持彩色显示器和数学协处理器，但这些硬件并不是必需的。

在附录A和B中，我们将对“电脑顾问”的特征和能力作简要的介绍。

8.1.1 开发机

开发机帮助专家建立、调试和精炼知识库，专家的知识正

是用这个知识库进行存贮的。当输入知识库元素时，开发机会提示专家怎样定义知识库元素，其中专家系统中与人机对话有关的内容也是这种定义的一个组成部分。开发机也能执行调试命令，因此，专家系统的开发者能够查找知识库中的错误。

8.1.2 推理机

推理机顺序地浏览知识库中的规则，向用户提出适当的问题，然后根据规则和用户对提问的回答提供咨询建议。

推理机是以反向推理方式进行工作的，但它也可以处理前件规则。

在“电脑顾问”中，规则的如果部分被称作“前提”，则部分称作“动作”，而属性则称作“参数”。这些术语在本章的后续部分还将用到。为了不引起混淆，今后在使用这些术语时，我们都将把以前用过的对应名称放在这些术语后面的括号里。

8.1.3 执行时实用程序

推理机和完整的、经过彻底调试的知识库一起组成最终的专家系统。然而，形成最终的专家系统这一工作是由执行时实用程序来完成的，它能将专家系统的上述两部分分别复制到一个有特殊复制保护的软磁盘上。这种软磁盘是特制的，需要单独购进。利用执行时实用程序可以不受数量限制地产生专家系统盘片。

8.2 语 法

在介绍怎样输入知识库元素之前，让我们来复习一下“电脑顾问”所采用的语法形式。“电脑顾问”所采用的模版格式与前面介绍的类似，如图8-2。其中参数（属性）、值和CF需由专家定义、谓词、联结词以及算术运算符（在模版中未列出）

则由系统自动提供。

	参数			
	函数	(属性)	谓词	值
前提 (如果)		子句 1		
		子句 2		
动作 (则)		子句 1		
		子句 2		

函数 } 系统提供
 谓词

参数 (属性) }
 值 } 开发者定义
 C F

图 8-2 “电脑顾问”的语法模版

$$\frac{3A(B+C)(D-F)}{G(H+1)(J-1)}$$

A. 算术表达式

$$(3 * A * (B + C) * (D - F)) /\\(G * (H + 1) * (J - 1))$$

B. 简写式形式

图 8-3 算术表达式

8.2.1 算术运算符

专家系统的开发者可以利用算术运算符编写变元为参数（属性）的方程。算术运算符包括+（加）、-（减）、*（乘）和/（除）。在图8-3中给出了算术表达式的例子，其中括号用来说明运算的先后次序。

8.2.2 谓词

谓词常用来构成子句，主要有三类：

- 数值谓词
- 值谓词
- 变量（属性）谓词

数值谓词 数值谓词用来表示参数（属性）与其取值之间的数量关系。它既可以用在规则的如果部分中，也可以用在规则的则部分中。下面是“电脑顾问”中的几个数值谓词：

<	小于
>	大于
>=	大于或等于
<=	小于或等于

利用数值谓词的子句例子如下：

“实际车速”>“车速限制”

值谓词 值谓词的作用是使一个值与其所属参数（属性）发生联系。当它用于动作（则部分）中的子句时，该子句的确

定性由其对应的 CF 决定。下面是一个在动作（则部分）中使用值谓词的例子：

(“判断能力” = “很差” (100%))

(“判断能力” != “很差” (80%))

在该例子中，= 和 != 都是值谓词，! 表示“非”或“不是”。

若是用在如果部分中，值谓词有多种形式且含义也不尽相同。图 8-4 给出了值谓词的三种形式及其含义。子句“判断能

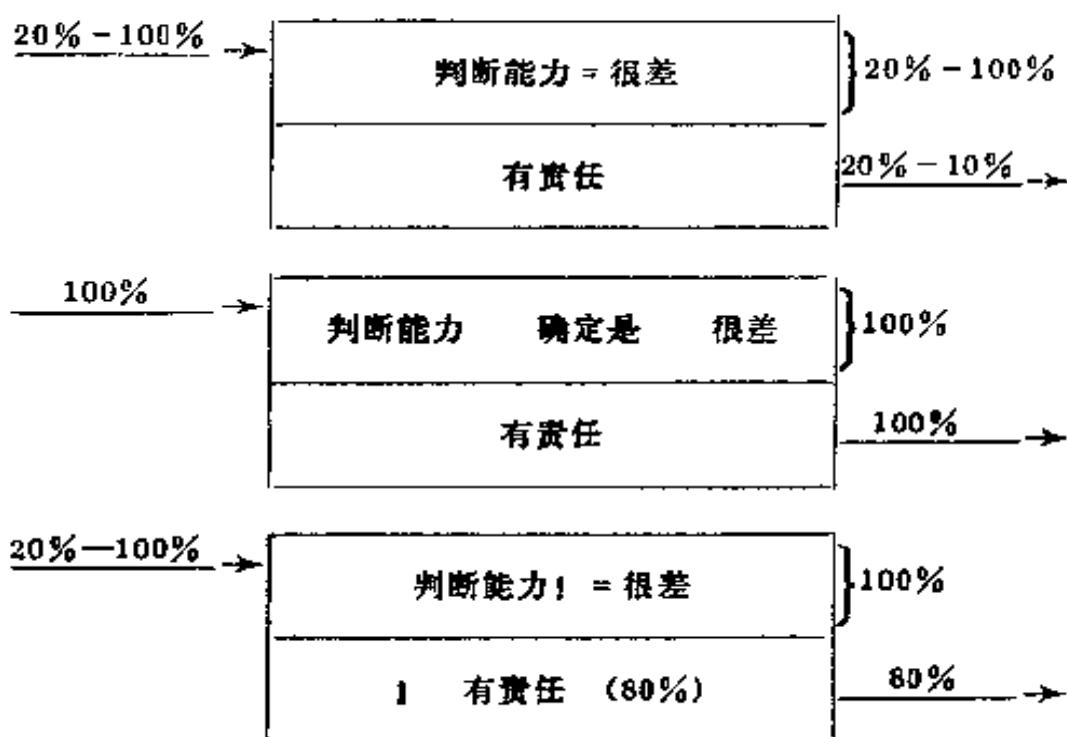


图 8-4 值谓词的例子

力” = “很差”的意思是如果“判断能力”“很差”的确定程度在 20% 以上，则激活该规则。除此之外，还有一层意思，这就是系统在计算如果部分 CF 的有效值时，必须考虑这个子句的 CF 值。因此，若已知该子句的确定性程度是 40%，由于它

是如果部分中的唯一子句，故如果部分CF的有效值就是40%。

另一种值谓词是“确定是”(IS DEFIS)。“确定是”的意思是只有当前提(如果部分)中该子句的CF为100%时，才激活规则。此时，若该子句是如果部分仅有的子句，则如果部分CF的有效值为100%。

与肯定的情况类似，如果子句“判断能力”! = “很差”的意思是如果“判断能力”不是“很差”的确定程度在20%以上，则激活该规则。与肯定的情况不同的是，在计算如果部分CF的有效值时，并不需要考虑! = 相应的CF，不论该子句的CF是20%到100%之间的任何值，系统都将其当作100%来处理。

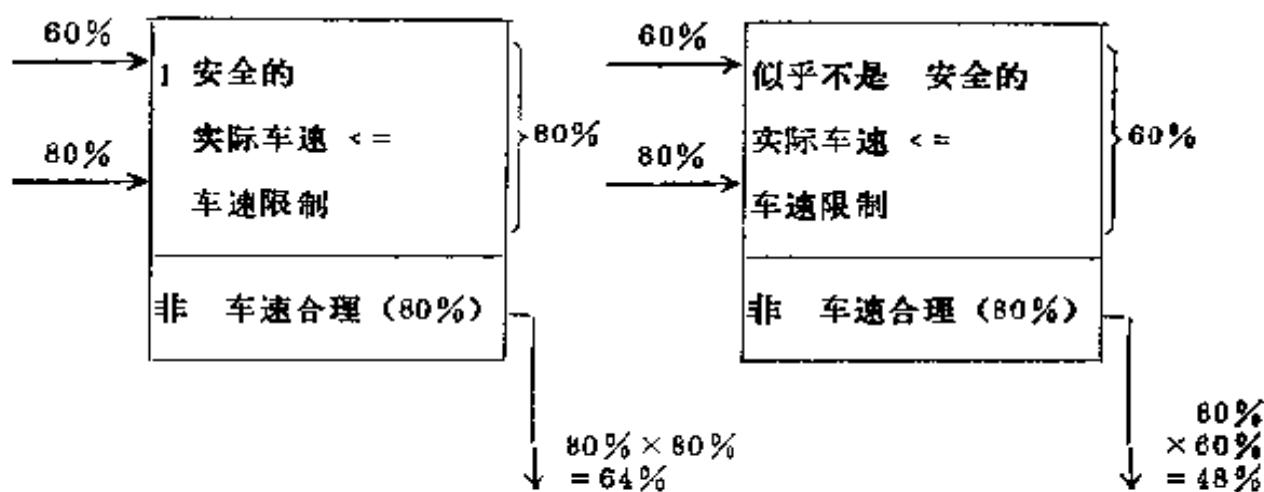
表示否定型如果子句的另一种方法是采用谓词“似乎不是”(IS THOUGHTNOT)。采用该谓词输入的CF能够通过规则。

在图8-5中，分别对用“与”(图A)和“或”(图B)联结如果子句时，!(“非”)与“似乎不是”在用法上的区别进行了比较。在图A中，输入给子句：“安全的”的CF是60%，由于!的作用它被忽略了，而由系统假设为100%。故取80%作为该如果部分CF的有效值。最终结论的CF，于是为80%乘以80%，即64%。

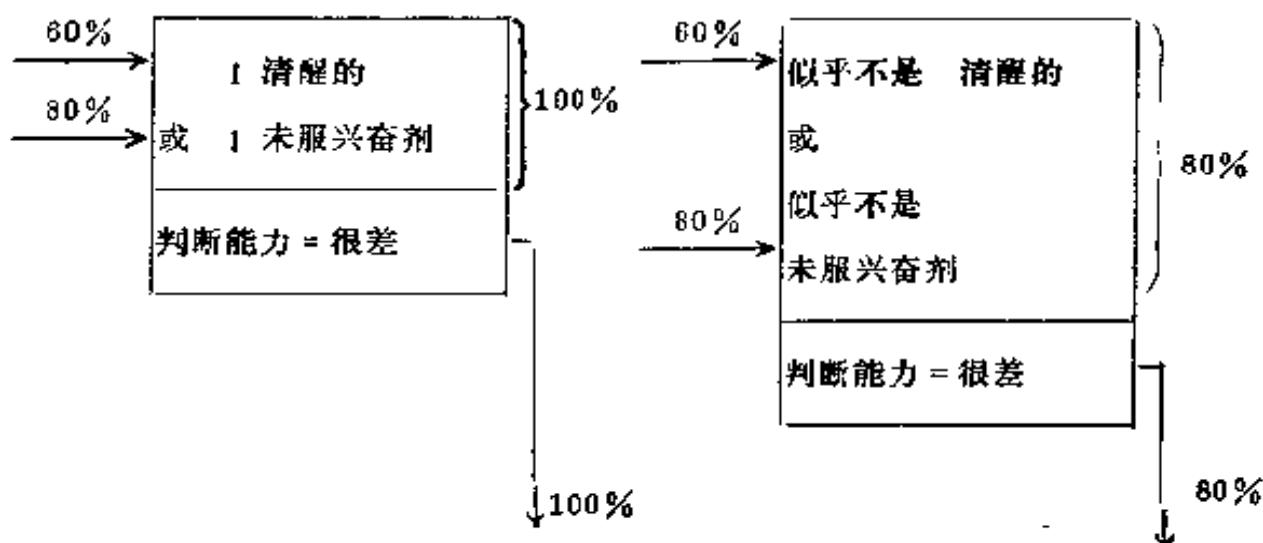
在使用“似乎不是”时，取各如果子句中较小的CF——60%——作为如果部分CF的有效值。此值再与则部分的CF——80%——合成，所得到的结果就是结论的CF值，只有48%。

在图8-5B中，当使用!时，系统不允许任何CF通过，即均被假设为100%。因此，规则输出结论所具有的CF为100%。然而，在使用“似乎不是”谓词时，各如果子句中输入CF的最大值——80%——变成了如果部分CF的有效值。由于规则

中则部分的CF是100%，因此，规则输出CF值也是80%。



A. 使用“与”的情况



B. 使用“或”的情况

图 8-5 “与”与“似乎不是”之间的比较

关于其它值谓词，由于在附录A中均有叙述，因此，在这里不再赘述。

参数（属性）谓词 参数（属性）谓词只能用在前提（如果部分）中。它只说明参数（属性）取值的确定性程度，而不管

参数（属性）究竟取什么具体值。

“是确定的”(IS DEFINITE)和“已知”(IS KNOWN)是两个参数(属性)谓词。在图8-6中，我们可以发现两者在意义上的差别。如果在前提(如果部分)中有“可见程度”“是确定的”，则不管“可见程度”的取值是“很好”、“一般”还是“很差”，只要它取值的确定性程度是100%，动作A就会被启动。如果在前提(如果部分)中有“可见程度”“已知”，同样，不管“可见程度”的值是“很好”、“一般”还是“很差”，只要“可见程度”取值的确定性程度大于20%，就启动动作B。

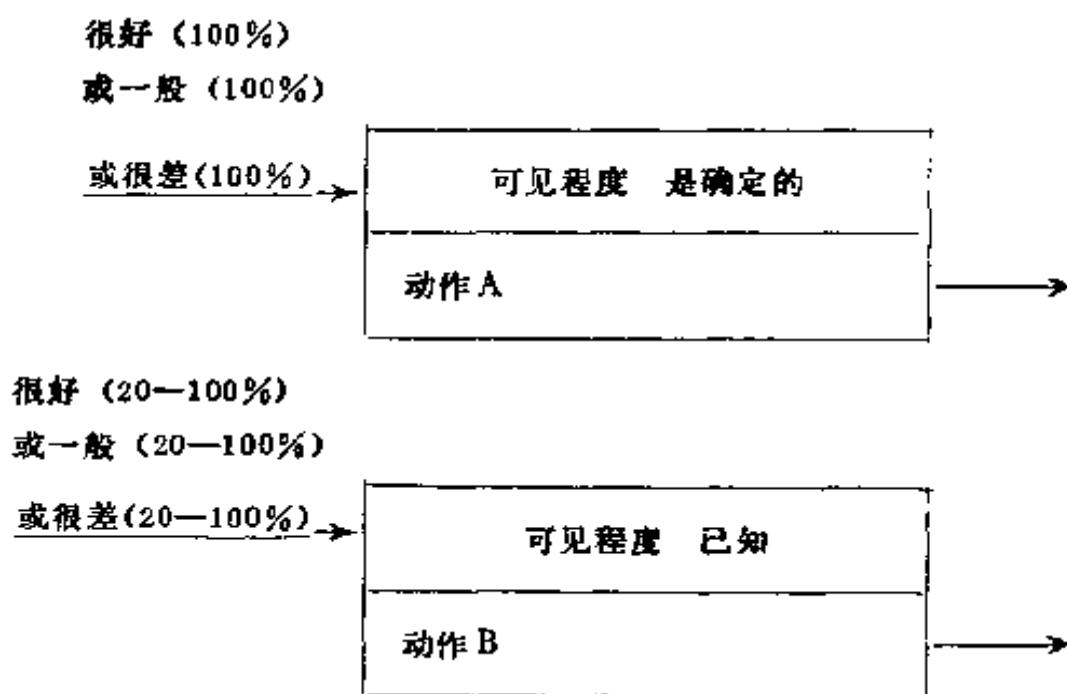


图 8-6 参数(属性)谓词的例子

8.2.3 函数

在“电脑顾问”中，谓词也可看作函数。然而，函数这个术语在此处有着特定的含义，它不能作为子句中的成分出现，

而是用来联结子句的。函数也分好几种，其中最重要的有：

- 逻辑函数
- 文本函数
- 用户定义函数

逻辑函数 逻辑函数包括“与”(AND)和“或”(OR)函数。“与”既可用在前提(如果部分)中，也可用在动作(则部分)中。用“与”联结如果子句的意思是在规则被激活之前，所有子句都必须为真；而用“与”联结则子句的意思是所有子句都会得到执行。

函数“或”只能用在如果部分中。用“或”联结子句的意思是只要其中一个子句成立就足以激活规则。

在如果部分中，子句既可用“与”联结，也可用“或”联结。

文本函数 文本函数只能用在规则的则部分中。有一个文本函数称“打印”(SPRINTT)，你可以用它来输出用引号括起来的任何文本信息，如图8-7。

前 提
(如果部分)：(↑ 清醒的 或 ↓ 未服兴奋剂)

动 作
(则部分)：(判断能力=很差 与 SPRINTT “开车前酗酒
和服用兴奋剂都是十分危险的。”)

图 8-7 使用文本函数的规则

用户定义函数 大多数专家系统只需利用“电脑顾问”现

有的函数就能开发成功。但是，有时也会出现这种情况，在开发专家系统时，你发现的确需要某个特殊的，而“电脑顾问”又没有提供的数学函数或查表功能。编写这种用户定义函数必须具备一定的LISP编程知识和能力。

如果你不会编LISP程序，这不要紧，你可以请具有这方面技能的人来为你编写相应的函数。以后，你可以在所有规则中引用新的函数。详情见附录B。

8.3 输入知识库元素

在“电脑顾问”中，运算符、谓词和函数都是由系统预先提供的，作为开发者，你必须完成的工作是定义上下文、参数（属性）和规则。在“电脑顾问”的引导下，你可以十分方便地将上述知识库元素输入到知识库中。

8.3.1 输入上下文

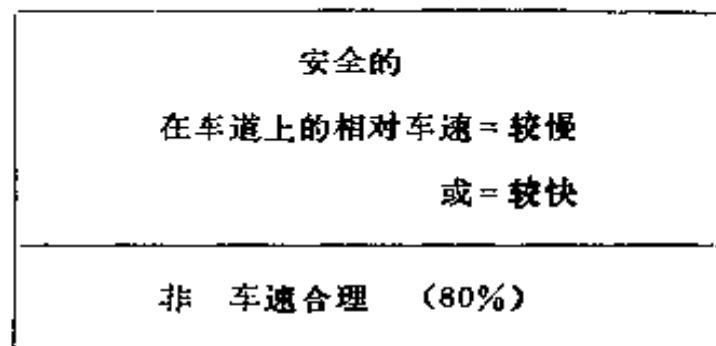
首先输入的是上下文，每输入一个上下文，系统都会询问上下文的各种特性的定义，某些参数（属性），如作为目标的参数（属性）也包括在这种定义中。

8.3.2 输入规则

图8-8说明一条规则是怎样以简写式规则语言（ARL）输入系统的，这种简写式规则语言最初为专家系统开发工具EMYCIN所采用。在这里，如果部分中的子句是逐个输入的，子句与子句用“与”或“或”联结，则部分的输入类似。不管如果部分还是则部分都必须用括号括起来。图8-8C给出了一条规则的自然语言解释。

在输入规则的动作（则部分）时，你可以在-1000至+1000之间为CF选择一个值。在这里，-1000表示100%的不确定，而+1000则表示100%的确定。如果CF值缺省，则系统默

认CF为100%.



A. 图式规则

前 提

(如果)：(安全的 与 在车道上的相对车速 = 较慢 或
在车道上的相对车速 = 较快)

动 作

(则)：(↑ 车速合理 (80))

B. 简写式规则

规则 016

如果 1) 情况对于驾车来说是安全的，并且
2) 1) 相对于同一车道上的车流，
 汽车在车道上的车速较慢，或
2) 相对于同一车道上的车流，
 汽车在车道上的车速较快，

则 有较强的证据显示(80%)：
 司机在道路上的车速合理不成立。

C. 自然语言式规则

图 8-8 输入规则到“电脑顾问”中

规则输入完成后，系统接着提示你输入这一规则的隶属关系，规则与它所属的上下文之间正是通过这种关系联系起来

的。因此，对于属于上下文“道路”的规则，可以说明它有隶属关系“道路规则”(ROADRULES)。

8.3.3 输入参数(属性)

在输入上下文和规则时，你要用到参数(属性)。此时，开发机会提示你定义这些属性。系统还会提示你输入每个参数(属性)的自然语言翻译，以及对用户的提示或提问信息等。系统也将提示你定义每个属性的取值范围。

8.4 人机对话

人机对话的定义是在开发机的协助下进行的。在这项工作中，你可以安排提问的顺序，定义对话的内容，编写解释以及你认为必要的任何信息。

8.4.1 提问次序的安排

当推理机从目标出发，追溯到可观察事实，然后，又回过头来沿着指向目标的路径逐个激活相应规则的时候，推理机同时也沿着此路径以向用户提问的方式收集必要的数据(详细过程见图8-9)。系统提问的时间安排是由你(专家系统的开发者)在定义下述对象时确定的：

- 上下文
- 可推演事实
- 可观察事实

定义上下文时 在根上下文中，你可以设置一段关于专家系统名称、功能、研制者等内容的“标题”信息，这段“标题”信息会在专家系统启动时显示出来。同样，在其它任何上下文中都可以设置一段引言式的显示信息。

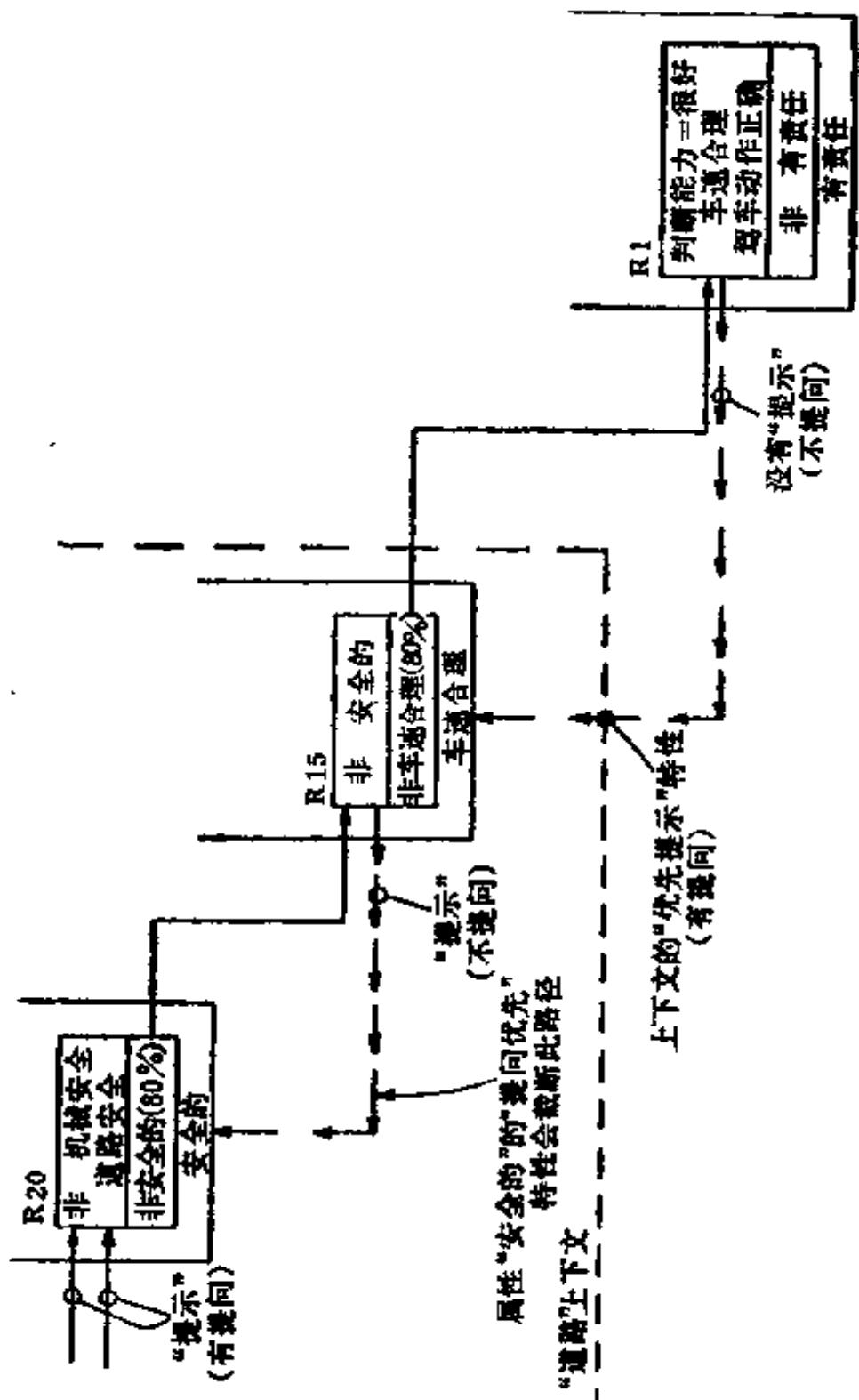


图 8-9 人机对话中专家系统提问的次序

利用“电脑顾问”提供的“优先提示”(PROMPT1ST)特性,你可以设置一个用于打开上下文的提问。例如,在“道路”上下文中,你可以设置这样的“优先提示”,“这个交通事故是否发生在公路上?”,当用户用“是”回答该提问时,系统进入上下文“道路”,并进行推理。如果你没有设置“优先提示”,则系统进入上下文前,不会有任何提问而是按正常的逻辑运行。

定义可推演事实时 可推演事实,即那些依赖其它属性的属性。如果你不为可推演事实设置“提示”(PROMPT)特性,则系统在推理中,当进行到对它的求解时不会产生任何提问。因此,在图8-9中,没有对R1中的“车速合理”的提问信息。

如果你为可推演属性设置了“提示”特性,那么系统是否通过提问方式求解该可推演属性,这取决于你有没有规定相应的属性是“提问优先”(ASKFIRST)属性。在图8-9的示例中,系统从R15追踪到R20时,并没有对属性“安全的”进行提问。当我们定义属性“安全的”具有“提问优先”特性时,那么系统在进到R20之前,会在R15对属性“安全的”进行提问式求解。

如果在规则簇“安全的”中,系统无法确定属性“安全的”的取值,则它在从R20返回到R15之前,会用提问的方法从用户那儿获得属性“安全的”的值。

定义可观察事实时 如果你给相应的参数(属性)设置“提示”特性,届时,系统在跟踪到该属性时就会向用户提问,否则,系统不会产生提问信息。例如,规则R20中的两个输入属性有“提示”特性,因此,当系统对它们进行求解时,产生了关于“机械安全”和“道路安全”属性的提问。

8.4.2 对 话

为了方便开发者设计对话，“电脑顾问”允许开发者事先说明提问的内容及其可能的答案。提问的内容（即问句）是由属性的“提示”特性定义的，可能的答案则由“期望值”(EXPECT) 特性定义。

“提示” 在“提示”特性中，你要告诉专家系统构造问句的方式。你可以直接给出一个完整的问句，也可以仅给出一个值 T。如果是后者，系统将根据属性的“翻译”(TRANS) 特性的内容自动地构造问句，“翻译”特性的内容是你为属性定义的自然语言解释。例如，属性“道路安全”的“翻译”特性是“道路状况对于驾驶来说是安全的”。

“期望值” “期望值”特性要求你定义属性的取值类型，如数字、正数、文本等。如果属性是单值的或多值的，你还得列举出用户可能选择的全部值。

8.4.3 解释推理过程

“电脑顾问”向用户提供了三种类型的解释推理功能。其中每一种功能都与一个可由用户随时按动的功能键对应，你可以根据自己的需要控制相应的推理解释功能。表示这三种推理解释活动的功能键是：

- “求助” (HELP)
- “为什么” (WHY)
- “怎样推” (HOW)

“求助” 如果用户不理解专家系统向他提出的问题，他就按下“求助”键，并可望获得解答。这是由于作为专家系统开发者的你已经把对提问的进一步解释等信息事先输入到问

题属性的“再提示”(REPROMPT)特性中了，因此，用户看了这个解释以后，就可能对原来的提问比较理解。

你可以使用预先准备好的句子告诉用户怎样回答问题，特定的词汇的含义是什么，以及回答问题时应采用的单位或任何你认为有用的内容。

“为什么” 如果用户不明白为什么专家系统要提问某个

提示：汽车是否能安全行驶？

为什么？

汽车是否能安全行驶可用来判断驾驶条件是否安全

规则 020

如果 1) 汽车不能安全行驶，并且

2) 道路状况对于驾驶来说是安全的。

则 有较强的证据显示(80%)：驾驶条件是安全的不成立。
立。

为什么？

驾驶条件是否安全可用来判断司机在当时的道路状况下
是否车速合理

规则 015

如果 驾驶条件是安全的不成立。

则 有较强的证据显示(80%)：司机在当时的道路状况
下车速合理不成立。

图 8-10 “为什么”命令

问题，他可以按下“为什么”键。此时，推理机将通过向用户显示当前正在使用的规则来进行解释。如果用户再次按下“为什么”键，他便会看到此规则将作用的那条规则（见图8-10）。

提示：当事故在交叉路口发生时，司机是否有驾车动作？
怎样推？

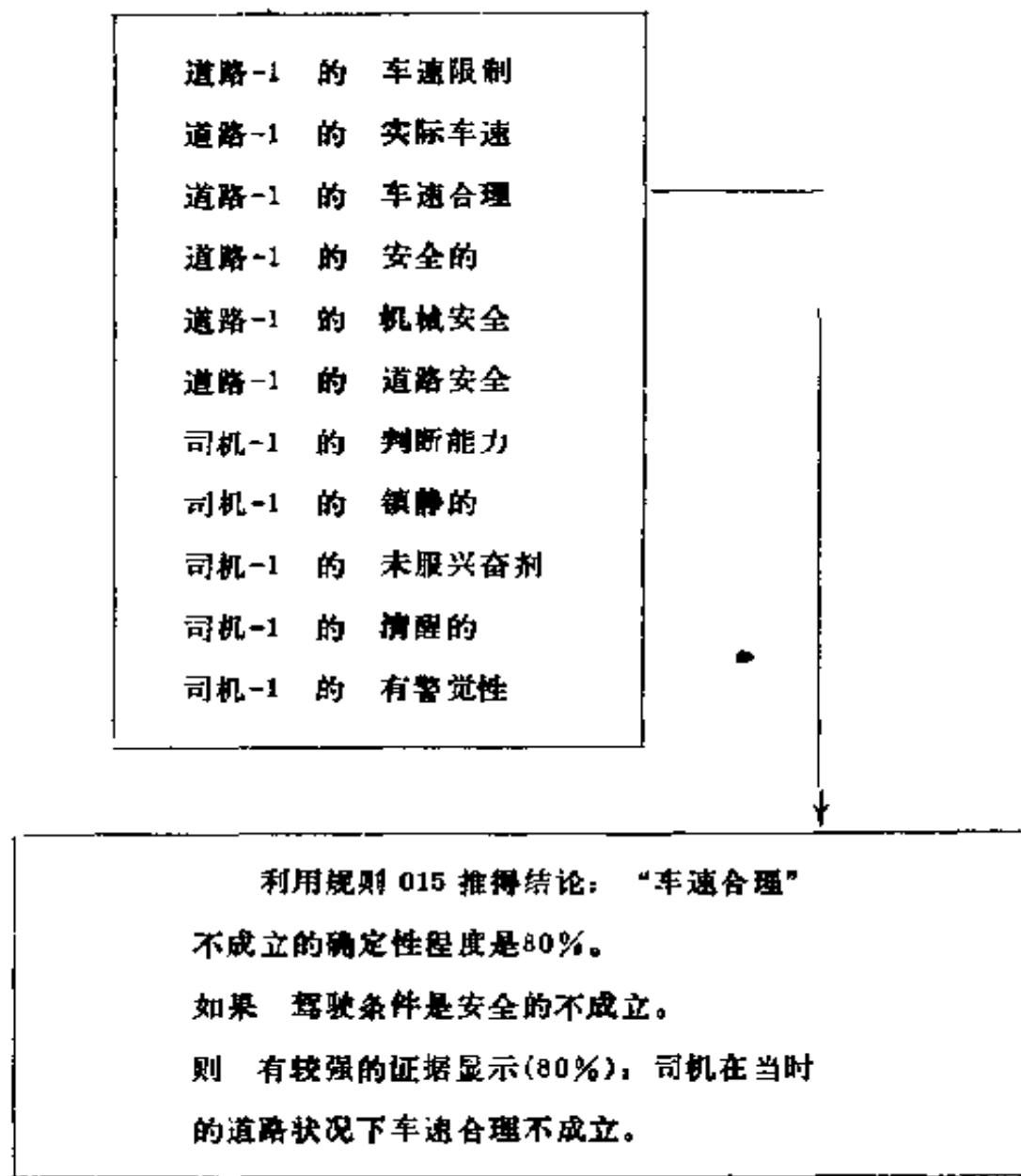


图 8-11 “怎样推”命令

在上述解释推理的过程中，描述规则的句子是否通顺，这取决于你给每个属性的“翻译”特性输入的内容。为属性编写的句子，不仅要能够提供正确的解释，而且应该尽量地做到当它用在规则中时，读起来仍然通顺流畅。

“怎样推”如果用户不明白系统是怎样推出结论的，他可以使用“怎样推”命令来询问专家系统。此时，系统将显示一系列属性，用户可以从中选出一个属性，然后，系统解释它是怎样推出关于该属性的结论的，其方法是向用户提供为推得该结论而引用过的属性和规则。在图8-11中，列出了一张这样的属性表，其中用户选中了“道路-1”中的属性“车速合理”（“道路-1”是上下文实例），并且显示了推得的结论以及用来推得此结论的规则 R15的描述。

同理，在此处显示的规则描述的可读性，取决于你给规则中各个属性的“翻译”特性设置的内容。

8.4.4 特殊信息

如果在某条规则被激活时，你想向用户提供一条信息，可以使用“打印”(SPRINTT) 函数，并把想显示的内容用引号括起来。

8.5 调 试

“电脑顾问”提供了几种辅助调试功能，以便帮助专家系统的开发者检查知识库的逻辑关系以及可用性。其中“定义”和“列清单”辅助功能用来帮助开发者输入知识库，“解释”和“追踪”辅助功能则在执行咨询时使用。

8.5.1 定 义

当你输入上下文、规则或属性时，系统的开发机会逐项提示你输入必须说明的各特性。它同时进行语法检查，当发现输

入的数据不正确时，它会立刻建议你给予更正。有时，它还会为你改正输入错误提供帮助。

8.5.2 列清单

当你往“电脑顾问”中输入了大量知识库元素之后，便可以使用“列清单”命令来显示所有的规则，也可以显示所有上下文或属性。所有这些知识库元素的清单信息既可以在荧光屏上显示，也可以输出到打印纸上。

“列清单”有多种用途。其中最直接的用途是通过比较计算机中存贮的内容与开发者手头的内容，从而判断输入是否正确。

通过检查清单信息可以确信每一个属性和规则是否隶属于正确的上下文。在每个属性后面显示的是相关属性组的名称，而该属性组又隶属于某个上下文。例如，在属性“判断能力”后面显示的是“司机的参数”(DRIVER PARMS)，这说明该属性隶属于上下文“司机”。在“判断能力”规则簇的规则后面跟着“司机的规则”(DRIVERRULES)，它表明该规则簇是隶属于上下文“司机”的。

使用“列清单”命令，也可以得到一给定规则簇的所有规则。对于某个属性，这些规则在副标“被更新”(UPDATED-BY)下列出，且按推理机查找的顺序显示(见附录D)。

8.5.3 解释

直接执行咨询过程是测试知识库的一种行之有效的方法。当推理机提问时，按下“为什么”键就可以知道现在专家系统正搜索到哪条规则了，而当专家系统得出结论时，按下“怎样推”键就可以知道系统在推得结论的过程中激活了哪些规则。

8.5.4 追踪逻辑流程

最有效的辅助调试功能是“追踪”(TRACE)命令。若你

使用“追踪”命令，在咨询过程中，与你回答系统提问的同时，系统会一步一步地追踪推理的逻辑过程，其中包括每一个上下文的打开，每个用户的输入，每个参数（属性）都会被追踪，每条规则都会被测试，每个动作都会被执行（包括结论）。

你可以将完整的追踪过程全部打印出来，以备闲暇时仔细研究。

图8-12说明了一个简单的追踪过程，以及与之对应的逻辑推理流程。这里事件的发生顺序仍然用圆圈中的数字表示，它们在追踪信息清单（图A）和推理图（图B）中代表相同的意义。

各事件的顺序如下：建立根上下文“司机-1”①。系统一旦进入上下文，就请求用户输入“司机姓名”，此时，用户输入了姓名“波尔”②。然后，系统决定追踪与此上下文相关的目标并按顺序搜索“有责任”规则簇中的第一条规则R1③。

系统开始追踪R1的第一个参数（属性）“判断能力”，因此，进入到规则簇“判断能力”中④。当系统遇到簇中第一条规则R4的第一个如果子句时，它通过提问向用户请求该参数（属性）的值。此时，用户输入了“没有警觉性”⑤。接着，系统向用户询问下一个参数（属性）的值，当接收到用户的回答是“镇静的”时，因为R4的如果部分为真，所以动作（则部分）也就为真⑦，即“判断能力”=“一般”，且其CF为1000，即100%。

当将此值回送到R1时，规则的前提（如果部分）不成立，因而系统进一步搜索到R2⑧。在这里系统对第二个属性“车速合理”进行询问（出于说明原理的目的，我们为它设置了“优先提问”特性），此时用户的回答是“不”⑨。到此有充分的理由执行规则R2的动作（则部分）了，因而得出结论：“司机

- ① 建立上下文
获得初始数据 : NIL :: 司机-1
用户输入 : (司机-1 司机姓名)
② 置参数
——参数追踪结束 : (司机-1 司机姓名) = ((波尔 1000))
追踪下列目标 : (司机-1 司机姓名)
: (有责任 判断能力 车速合理 驾车动作正确)
③ 追踪参数 : (司机-1 有责任)
尝试推导 (司机-1 有责任) 的规则: (规则001 规则002 规则003)
——测试规则的前提 : (司机-1 规则001)
追踪参数 : (司机-1 判断能力)
④ 尝试推导 (司机-1 判断能力) 的规则: (规则004 规则005 规则006
规则007 规则008)
——测试规则的前提 : (司机-1 规则004)
追踪参数 : (司机-1 有警觉性)
尝试推导 (司机-1 有警觉性) 的规则: NIL
没有关于该目标的规则 : (司机-1 有警觉性)
⑤ 用户输入 : (司机-1 有警觉性) = ((非 1000))
置参数 : (司机-1 有警觉性) = 是 CF - 1000
——参数追踪结束 : (司机-1 有警觉性)
追踪参数 : (司机-1 镇静的)
尝试推导 (司机-1 镇静的) 的规则: NIL
没有关于该目标的规则 : (司机-1 镇静的)
⑥ 用户输入 : (司机-1 镇静的) = ((是 1000))
置参数 : (司机-1 镇静的) = 是 CF 1000
——参数追踪结束 : (司机-1 镇静的)
执行规则的动作 : (司机-1 规则004)
⑦ 置参数 : (司机-1 判断能力) = 一般 CF 1000
没有关于该目标的规则 : (司机-1 判断能力)
——参数追踪结束 : (司机-1 判断能力)
⑧ 规则前提不成立
——测试规则的前提 : (司机-1 规则001)
——测试规则的前提 : (司机-1 规则002)

追踪参数	: (司机-1 车速合理)
⑨ 用户输入	: (司机-1 车速合理) = ((非 1000))
置参数	: (司机-1 车速合理) = 是 CF ~ 1000
——参数追踪结束	: (司机-1 车速合理)
执行规则的动作	: (司机-1 规则002)
⑩ 置参数	: (司机-1 有责任) = 是 CF 1000
没有关于该目标的规则	: (司机-1 有责任)
参数追踪结束	: (司机-1 有责任)

A、‘追踪’清单

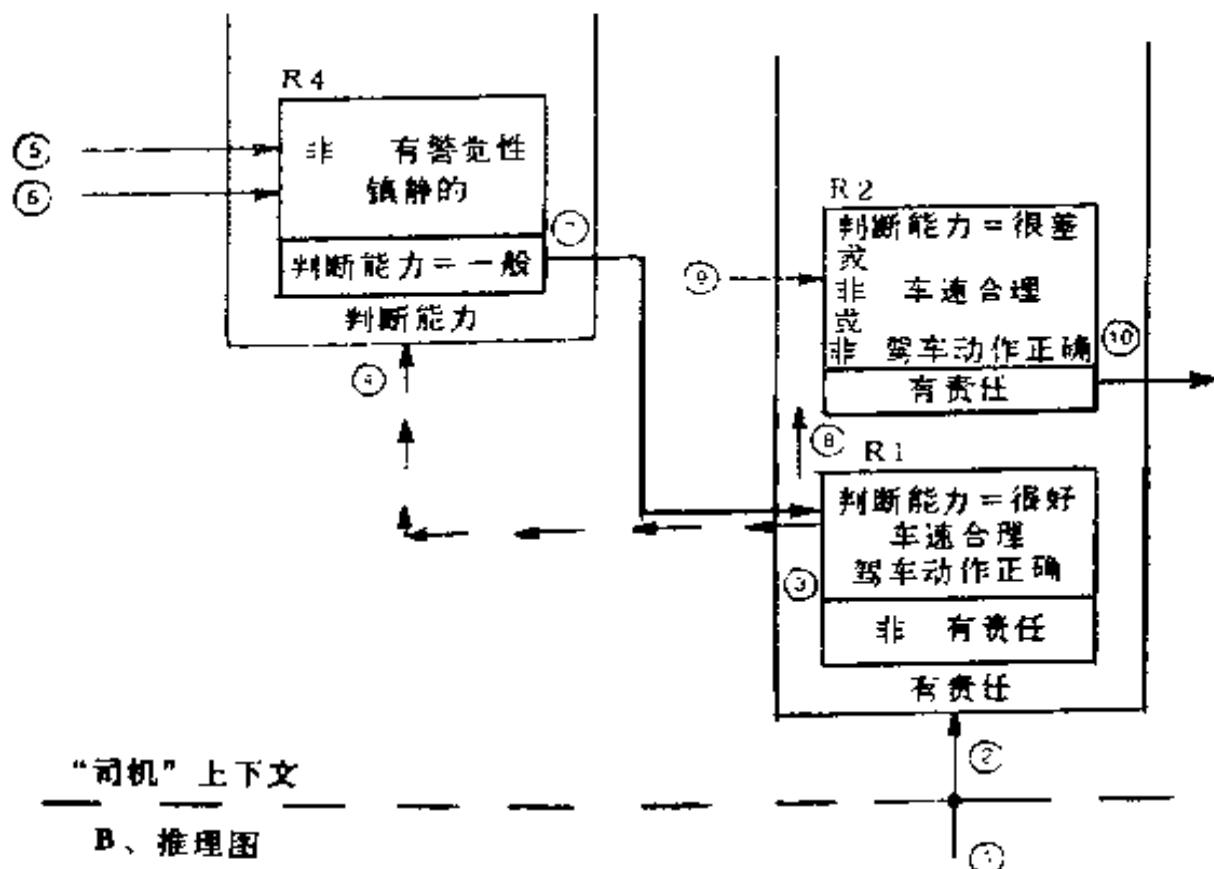


图 8-12 使用“追踪”命令

-1”对交通事故负“有责任”⑩。

为了节省时间，“电脑顾问”还提供了“记录(追踪过程)”(RECORD)命令，用它可以将完整的咨询过程记录到磁盘文

件中，利用“重现（追踪过程）”(PLAYBACK)命令，可以看到记录在追踪文件中的内容，即使现在知识库的内容已经改变也无妨。

第九章 阶段III：建立知识库

到目前为止，我们已经知道了“电脑顾问”是怎样推理的，因此，我们可以利用它来建立交通事故原型专家系统的知识库了。为了避免出现差错和便于将来测试知识库，请你先把知识库写在活页纸上。然后按照这本活页纸上的规则库图将所有知识库元素输入“电脑顾问”，并且对此知识库的推理逻辑和可信因子进行检查。

如前所述，本书所介绍的内容虽不象烹调手册中的内容那样，必须照样施行。但是，我们认为本章将介绍的一些方法和过程对你时十分有益的。由于它列举了一些你在建立自己的知识库时应该加以考虑的问题，因此，一旦你对这些因素熟悉了，我们相信你会形成自己特有的工作方法。

阶段Ⅲ的主要步骤是：

- 准备书面形式的规则库
- 输入知识库元素
- 检查规则的逻辑正确性
- 检查可信因子

9.1 准备书面形式的规则库

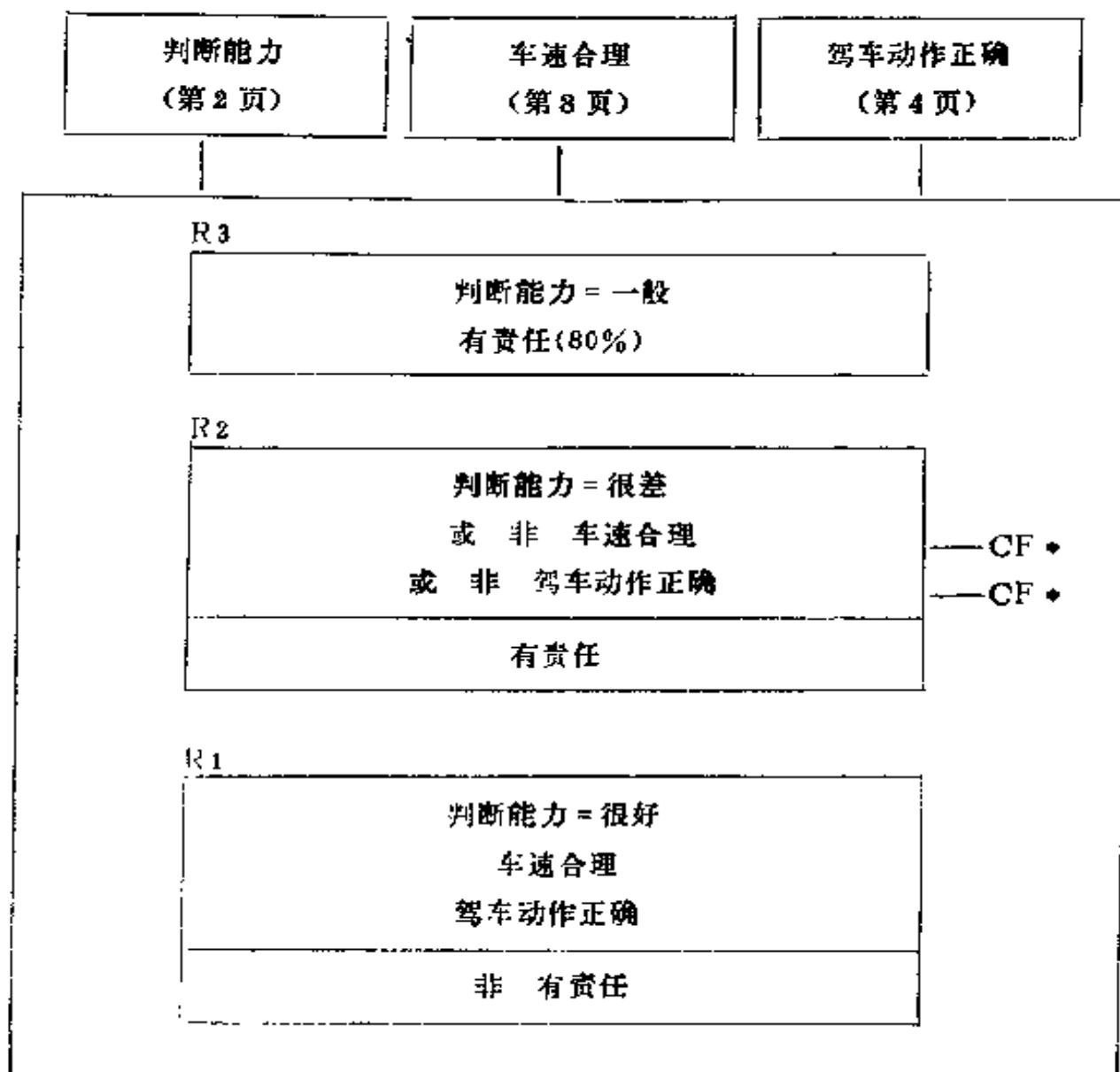
在输入知识库元素之前，准备工作应该做得既细致又充

分。尽管规则库的逻辑完备性是你关心的首要问题，然而，一旦你使用起软件来，你的注意力就会不由自主地转向诸如句法之类的软件细节中去。由于这个原因，为了保证你输入的知识库元素与规则库图一致，一个最好的办法是把你的全部规则都记录下来，而且做到每个规则簇用一张活页卡片记录。

9.1.1 记录规则簇的活页卡

上下文：司机

规则簇：有责任



• 此处值谓词是“似乎不是”而不是“!”（下同）

上下文：司机
规则簇：判断能力

R 7	<p>非 清醒的 或 非 未服兴奋剂</p> <p>判断能力 = 很差</p>
R 6	<p>非 有警觉性 非 镇静的</p> <p>判断能力 = 很差</p>
R 5	<p>有警觉性 清醒的 未服兴奋剂 镇静的</p> <p>判断能力 = 很好</p>
R 4	<p>非 有警觉性 镇静的</p> <p>判断能力 = 一般</p>

图 9-1 有关规则簇的两张活页卡

以图示方式将属于同一规则簇的规则记录到一张独立的方

形纸卡上，然后，将这些单个的纸卡收集到一个活页夹中形成一个活页本。无疑，这是一种十分简便易行的好方法，建议你不妨一试。以规则簇“有责任”和“判断能力”为例，这样做的结果是得到了如图9-1中所示的两张“活页”卡片。在该图中，处在顶端矩形中的页号表示该规则簇中引用的子辈规则簇所在的活页编号。

此外，请注意在规则R2的第二、第三个如果子句的右部，有一条标有CF的短横线，我们用它表示相应的CF能够“通过”该规则。用“电脑顾问”的术语就是，此如果子句所用的值谓词是“似乎不是”而不是“非”（即：）。

9.1.2 规则簇的完备性

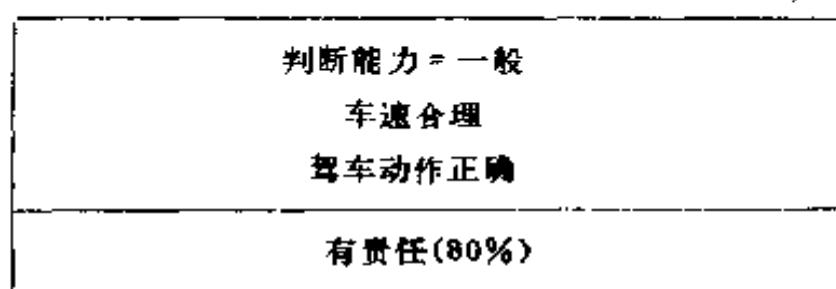
每当建立一个规则簇时，你都会面对这样一个不能回避的问题，即规则簇中的现有规则是否包括了所有逻辑条件？解决这个问题的一种行之有效的方法是采用选择表检查法。

运用具体例子也许最容易说明选择表的作用。在图9-2中，有一个关于规则簇“有责任”的选择表，在这个表的左上方是决定规则簇中结论的那些属性：“判断能力”、“车速合理”和“驾车动作正确”，在其下方列出了这些属性所有可能的取值组合。在表的右部则给出了适用于每种情况的规则。

对该表稍加分析便会发现，用图9-1中的R1、R2和R3三条规则并不能包括逻辑条件的所有可能性。在现在的表中，表的第一行定义了规则R1，各虚线框定义了R2，它规定不论其它属性取何种值，只要下述三个条件中有一个成立，“司机”就对交通事故负“有责任”。这三个条件是：司机的“判断能力”“很差”，他并没有保持“车速合理”或“驾车动作正确”。R3定义了表格中央第五行的情况：如果“判断能力”“一般”，则“司机”“有责任”的可信因子是80%。

判断能力	车速合理	驾车动作正确	有责任规则
很好	是	是 <input type="checkbox"/> 非 R2	R1 (非有责任)
		<input type="checkbox"/> 非 R2	R2 (有责任)
一般	是	是 <input type="checkbox"/> 非 R2	R3(有责任80%) 不完整
		<input type="checkbox"/> 非 R2	
		是 <input type="checkbox"/> 非 R2	
很差	R2	是 <input type="checkbox"/> 非 R2	R2 (有责任)
		<input type="checkbox"/> 非 R2	
		是 <input type="checkbox"/> 非 R2	
		<input type="checkbox"/> 非 R2	

修改后的R3



借助这个选择表，我们发现了R3中的隐含假设，即司机的

有警觉性	清醒的	未服兴奋剂	镇静的	判断能力规则
是	是	是	是 非	R5(很好) R6(一般) 一新 规则
			非 R7 是	
			非	
	非 R7 是	是	是	R7(很差)
			非	
			非 是 非	R4(一般)
非	是	是	是 非	R6(很差) } 需要 修改
			非 R7 是	
			非	
	非 R7 是	是	是	R7(很差)
			非 是 非	
R4	R6	R8		
非 有警觉性 清醒的 未服兴奋剂 镇静的	非 有警觉性 清醒的 未服兴奋剂 非 镇静的	有警觉性 清醒的 未服兴奋剂 非 镇静的		
判断能力 = 一般	判断能力 = 很差	判断能力 = 一般		

图 9-3 “判断能力”规则簇的选择表

“车速合理”和“驾车动作正确”，而这两个条件并没有明确地包括在R3的如果部分中。通过改写R3就可以补上这一逻辑“漏洞”。至此，所有影响属性“有责任”的逻辑条件就都被我们考虑到了。

图9-3是规则簇“判断能力”的选择表，其中虚线框定义了规则R7，它表示无论其它属性取怎样的值，只要“司机”是“非”“清醒的”或“非”“未服兴奋剂”的，则该“司机”的“判断能力”“很差”。另外，表中的第1行定义了规则R5，规则R4和R6分别由表中的第9和第10两行定义。但是，在图9-1中，我们看到规则R4和R6中并没有属性“清醒的”和“未服兴奋剂”存在。现在，我们借助选择表的帮助可以给R4和R6补上这些如果子句，这样，这两条规则就能反映这两行表示的情况了。

仔细观察这张表，我们还会发现，第二行所表示的情况被规则漏掉了。因此，需要增加一条新规则R8来弥补这个被遗忘的情况：司机“有警觉性”，并且是“清醒的”、“未服兴奋剂”和“非”“镇静的”，这时“司机”的“判断能力”是“一般”。增加了规则R8以后，规则簇“判断能力”在逻辑上就是完备的了。

9.2 输入知识库元素

以书面形式准备好规则库之后，你可以按上述两种次序输入知识库元素，这两种次序分别是输入上下文、属性和规则，或上下文和规则。若是用后一种次序，你必须在“电脑顾问”的提示下定义属性。

9.2.1 输入上下文

上下文是按一种称作深度优先的次序输入的，如图9-4中

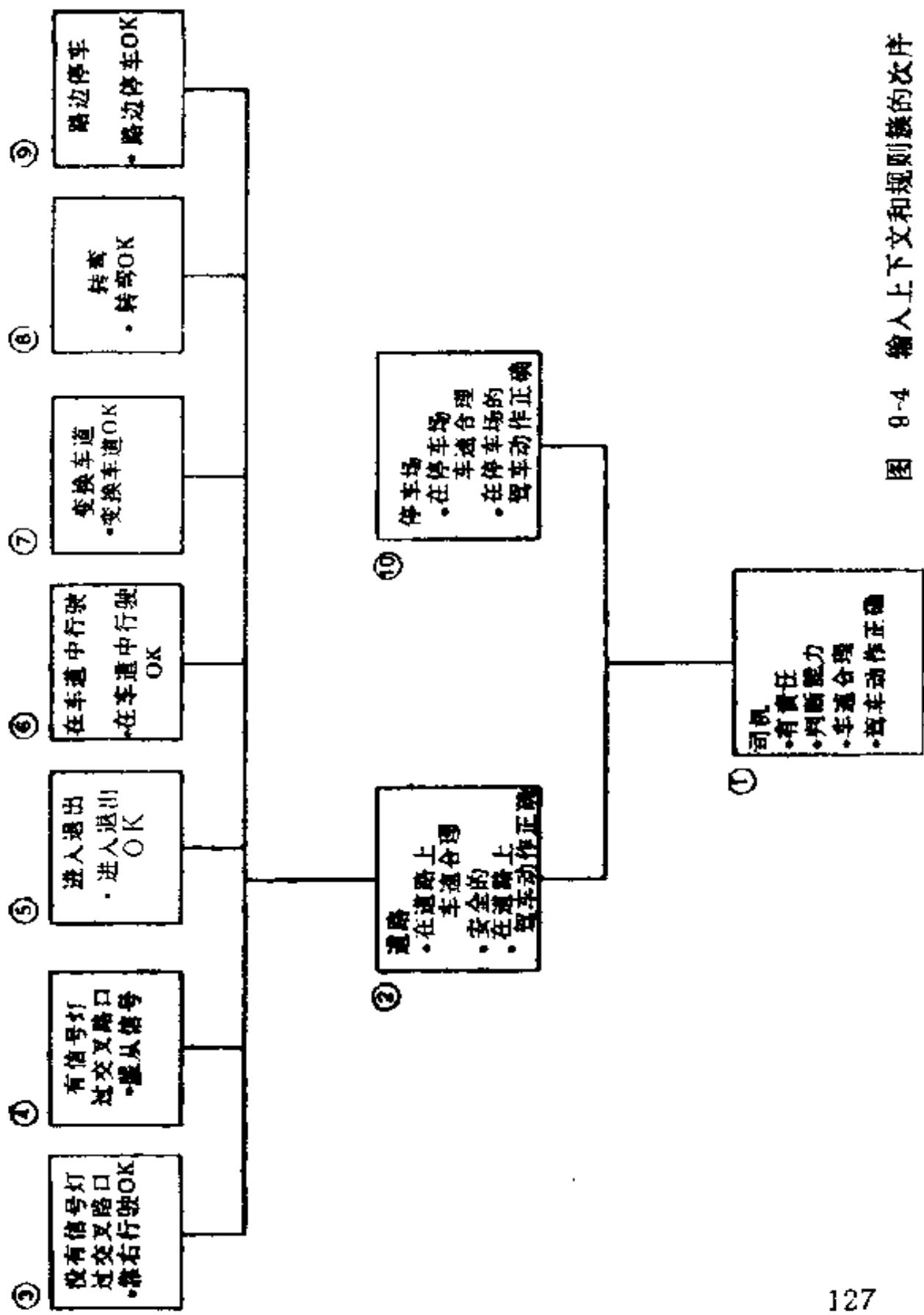


图 9-4 输入上下文和规则嵌套的次序

带圈的数字表示。即以从“司机”到“没有信号灯过交叉路口”的次序优先输入所有左枝，然后，顺序输入下一枝（右边的一枝），直到最右边的最低子辈（在此图中是顶部）。当再没有分枝可输入时，同样的规则就适用于父辈上下文。例如，整个交通事故原型上下文树的输入顺序是：“司机”，“道路”，“没有信号灯过交叉路口”，“有信号灯过交叉路口”，“进入退出”，“在车道中行驶”，“变换车道”，“转弯”，“路边停车”，“停车场”。

9.2.2 输入规则

规则的输入是按照图 9-4 所示的顺序，逐个上下文，逐个规则簇地进行的。

9.2.3 输入属性

可以首先输入属性，然后输入用到这些属性的规则；也可以在输入完所有规则后，再根据系统的提示定义每个新属性的各特性。借助在阶段Ⅰ中建立起来的各种属性表可以有条不紊地加速此过程的进行。

9.2.4 获得清单

当所有上下文、规则和属性都输入到“电脑顾问”之后，可以打印一份整个知识库的清单。将其中属性定义清单中的内容与你在阶段Ⅰ填入属性表和属性-值表的属性加以比较，并根据比较的结果作出合理的修改或调整。将打印出来的规则清单与你已有书面记录的规则进行比较，以便发现拼写错误，输入错误或其它明显的差错。

9.3 检查知识库的逻辑正确性

检查知识库的逻辑正确性，这一工作主要通过由开发者自己充当用户并反复地执行咨询来完成。每执行一次咨询都能检

查不同条件组合时知识库的逻辑结构。“电脑顾问”的解释和逻辑追踪功能为这种检查工作提供了有力的支持。

9.3.1 检查什么？

检查知识库的逻辑正确性应从以下两个方面入手：

- 属性的正确性
- 规则的逻辑正确性

属性的正确性 寻找属性定义的缺陷，应在检查规则库的规则簇时进行。属性的名称既不能含糊其辞，也不能引起歧义，并且应该尽量采用肯定的而不是否定的语气表述。当然，最重要的是没有遗漏必要的属性。

同时在父辈和子辈上下文中出现的属性应该归属于父辈上下文，而不是子辈上下文。同一属性不得分属上下文树中的不同分枝。

在检查规则簇中的每条规则时，还应该注意冗余的属性（几个属性具有相同的含义）和矛盾的属性（同一属性表示不同的含义）。

规则的逻辑正确性 检查规则逻辑的任务是：找出无法用规则定义的情况；相同的条件经由规则推得不同的结论；还有不同的规则产生相同的结果等。

9.3.2 检查方法

做逻辑检查的一般方法比起任何一项具体的实用技术来都显得更为重要。这个方法的基本思想是这样的，每次只检查一个规则簇，并且先从目标簇开始，然后进入到更深一层的规则簇，即“判断能力”、“车速合理”和“驾车动作正确”，再后则进入到“在道路上驾车动作正确”、“在停车场的驾车动

作正确”等。只要情况允许，在进入到另一个上下文中的规则簇之前应该尽量对当前上下文中的所有规则簇进行检查。在图9-6中，圆圈中的数字说明了一种可供参考的检查顺序。

但是怎样才能只检查一个规则簇，而又不致牵涉到与它联系的其它规则簇呢？尤其是当这些规则簇又属于其它上下文时。解决这个问题的方法是巧妙地利用“提问优先”特性。例如，对“有责任”规则簇所依赖的所有属性都设置“提问优先”特性。这样一来，“有责任”规则簇就可以成为一个隔离的“封闭盒子”。（见图9-5）

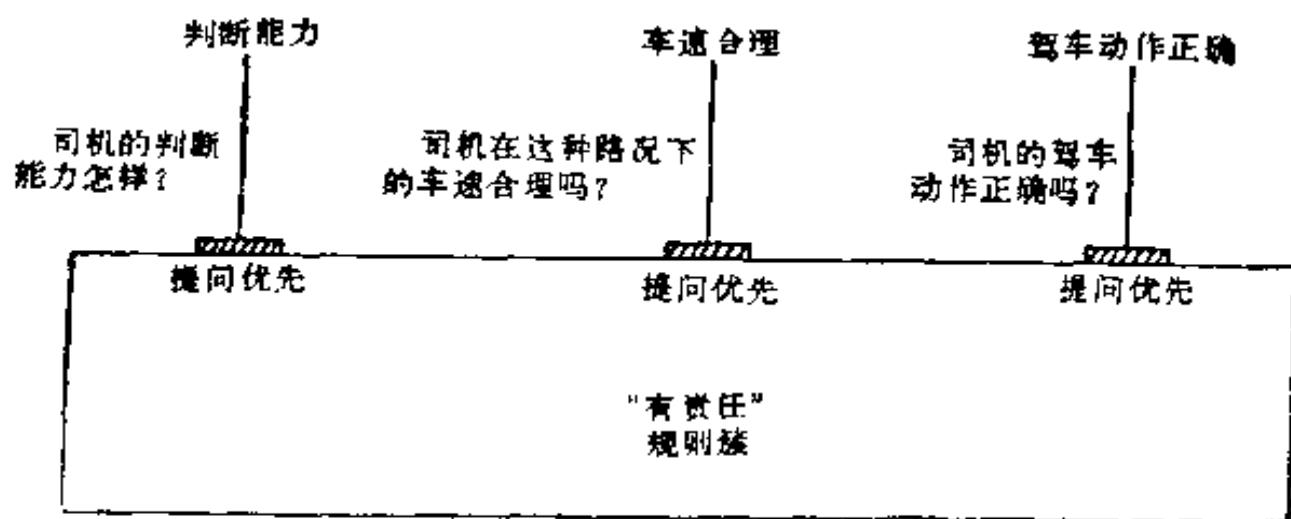


图 9-5 使“有责任”规则簇成为“封闭的盒子”

在执行咨询时，“电脑顾问”的推理机将对所有直接影响规则簇的条件属性进行求值。而“提问优先”特性可使系统产生三个向用户询问“判断能力”、“车速合理”和“驾车动作正确”取值情况的问句。通过给这些属性假设所有可能的取值组合，你便能够十分容易地检查该规则簇中的所有规则。

检查第一个规则簇也许还算容易，那么对后续的各深层规则簇应该如何检查呢？再具体一些就是，怎样使深层规则簇，

诸如“驾车动作正确”，也成为“封闭的盒子”呢？对于这个问题，让我们来看图9-6中是怎样做的。在该图中，首先给属性“在道路上驾车动作正确”和“在停车场的驾车动作正确”

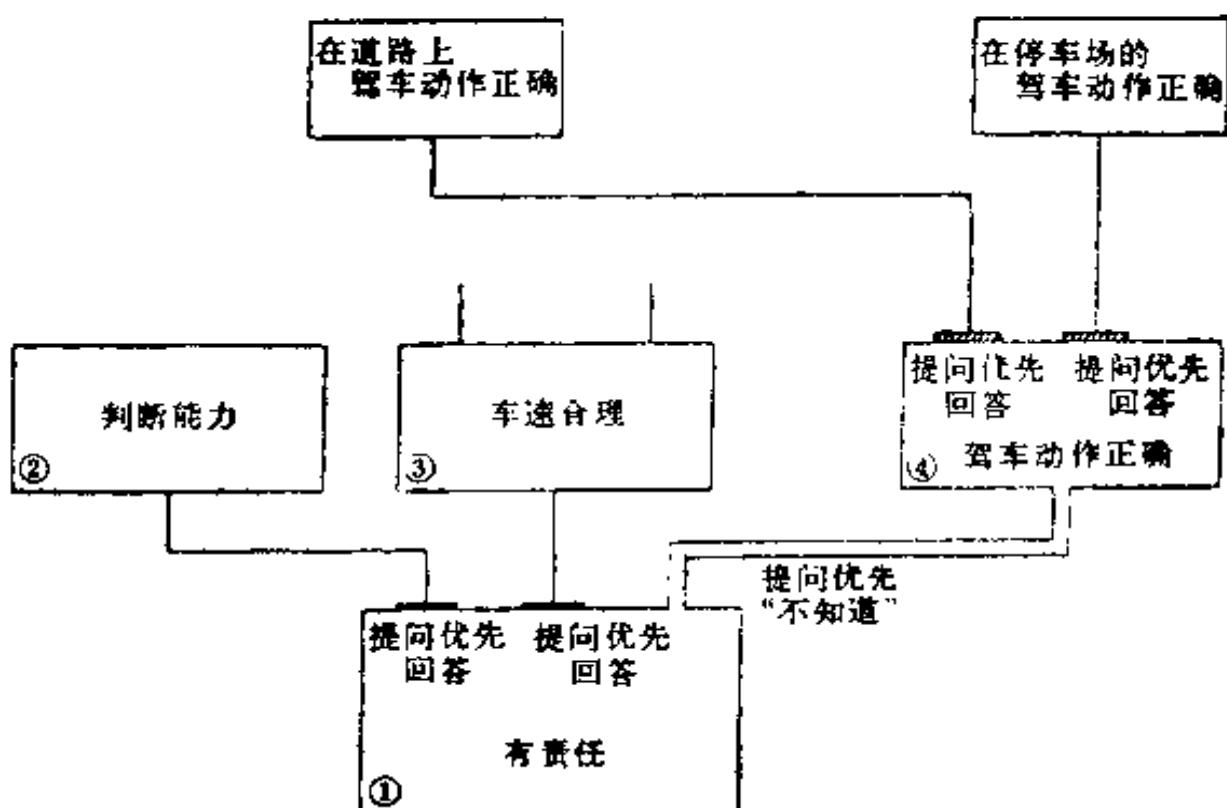


图 9-6 检查规则簇逻辑的顺序

设置“提问优先”特性，然后便开始执行咨询。这时，推理机如以前检查目标簇一样，再从同一类提问开始，即从与“有责任”簇中规则对应的提问开始。对系统关于属性“判断能力”和“车速合理”的提问，可以分别用确切的值“很好”和“是”来回答，而对关于属性“驾车动作正确”，也就是被检查规则簇的属性，的提问，则应该用“不知道”(UNKNOWN)来回答。回答“不知道”可以打开封闭规则簇“驾车动作正确”的“阀门”。现在，你已进入了“驾车动作正确”这个新“盒子”，下面你所要做的如同检查目标簇时在“有责任”

规则簇中所做的几乎没有什么差别，所不同的只是被检查的规则和需要回答的提问改变了。即通过给关于“在道路上驾车动作正确”和“在停车场的驾车动作正确”的提问以不同的回答，来检查规则簇中的各规则。

假若你还想进一步检查规则簇“在停车场的驾车动作正确”，首先，如上所述，进入规则簇“驾车动作正确”，然后，在回答关于“在停车场的驾车动作正确”的提问时，不用确定的值来回答，而是再次用“不知道”回答该提问。通过这个“不知道”回答，系统又一次打开了封闭规则簇“在停车场的驾车动作正确”的“阀门”，从而使我们得以进入该规则簇，并对其中的各规则进行检查。

通过恰到好处地利用“提问优先”特性和回答“不知道”你便可以有选择地进入任何需要检查的规则簇。待规则检查完成后，应该从相应的属性中取消不希望有的“提问优先”特性。

9.3.3 使用“为什么”和“怎样推”命令

找出系统逻辑缺陷的另一个好方法是，在检查规则簇的时候，系统地使用“为什么”和“怎样推”命令。当推理机向你提出一个问题时，键入“为什么”命令就可以知道当前使用的是哪条规则；然后，你就可以根据情况决定是检查当前这个规则，还是让系统继续检查规则簇中的下一条规则，并按照你的判断对提问提供相应的回答。如果从系统的提问或显示的信息来看，知道它已经离开了本规则簇，则可以使用“怎样推”命令来检查规则簇中的属性，系统此时会告诉你已推得的结论以及为了获得该结论实际用到了规则簇中的哪些规则。

图 9-7 中是一个使用“为什么”和“怎样推”命令的例子。表格中有四栏，其中跟在每个系统提问右边的是一系列与

专家系统的提问	为什么/ 怎样推	专家系统的响应	对提问的回答
①波尔与前面的车辆保持 了几个车身的距离?	为什么?	R 146	6
②发生车祸时波尔的实际车速是多少?	为什么?	R 57	40
③波尔是否撞上了车辆?	为什么?	R 59	是
④波尔是否处在双右车道中的一条之中?	为什么?		非
对于波尔是否对交通事故有责任, 我暂时无法得出任何结论...	怎样推?		

对提问的回答

6

40

是

非

对于波尔是否对交通事故有责任，我暂时无法得出任何结论...

图 9-7 使用“为什么”和“怎样推”命令查找逻辑错误

此相关的动作。每当系统给出一个提问，作为开发者的你，在正式回答该提问之前，应该先用“为什么”或“怎样推”进行反问，系统则会以规则来答复你的这个提问。尽管实际上“电脑顾问”是用自然语言形式显示规则的，但是，在此处为了更直观起见，我们以图形的方式给出规则。

在例子中，第一个提问是“波尔与前面的车辆保持了几个车身的距离？”，你键入“为什么”，系统给出规则R146作为回答，该规则的激活条件是，“车身距离” \leq “实际车速”/10，此时你对系统提问的正式回答是“5”①。下一个提问是“发生车祸时，波尔的实际车速是多少(公里/小时)？”此时，你回答“40”②，从而使规则R146的条件不成立。对关于属性“拖挂”的提问③反问“为什么”，可以知道系统正在使用规则R57。然后，再回答“是”，使系统搜索下一条规则，并再用“为什么”反问系统，在知道了这条规则是R58后，你回答“非”④。至此，系统不再提问，而是告诉你“对于波尔是否对交通事故负有责任，我暂时无法得出任何结论……”

为什么会出现这种情况呢？为了追究其原因，你可以使用“怎样推”命令，首先检查显示出来的属性表，你会发现属性“在车道中行驶OK”未列出，这表明系统没有得出关于是否“在车道中行驶OK”的结论。

出现这种现象一定是遗漏了某条规则。这个被遗漏的规则是R143，它说明如果司机的汽车后面拖挂了拖厢，并且他不在双右车道中行驶，那么他在车道中行驶合理这个结论就不成立。

9.3.4 使用“追踪”命令

在“追踪”状态下执行咨询时，每当你对一个提问给予回答，系统都会将相应的输入值、被追踪的参数（属性）、上下

建立上下文 : 道路 -1:1 在车道中行驶 -1
 追踪下列目标 : NIL
 测试规则的前提 : (在车道中行驶 -1 规则146)
 追踪参数 : (道路 -1 车身距离)
 尝试推导 (道路 -1 车身距离) 的规则: NIL
 没有关于该目标的规则: (道路 -1 车身距离)
 ① 用户输入 : (道路 -1 车身距离) = ((5 1000))
 置参数 : (道路 -1 车身距离) = 5 CF 1000
 参数追踪结束 : (道路 -1 车身距离)
 追踪参数 : (道路 -1 实际车速)
 尝试推导 (道路 -1 实际车速) 的规则: NIL
 没有关于该目标的规则: (道路 -1 实际车速)
 ② 用户输入 : (道路 -1 实际车速) = ((40 1000))
 置参数 : (道路 -1 实际车速) = 40 CF 1000
 参数追踪结束 : (道路 -1 实际车速)
 规则前提不成立 : (在车道中行驶 -1 规则146)
 测试规则的前提 : (在车道中行驶 -1 规则057)
 追踪参数 : (道路 -1 拖挂)
 尝试推导 (道路 -1 拖挂) 的规则: NIL
 没有关于该目标的规则: (道路 -1 拖挂)
 ③ 用户输入 : (道路 -1 拖挂) = ((是 1000))
 置参数 : (道路 -1 拖挂) = 是 CF 1000
 参数追踪结束 : (道路 -1 拖挂)
 规则前提不成立 : (在车道中行驶 -1 规则057)
 测试规则的前提 : (在车道中行驶 -1 规则058)
 追踪参数 : (在车道中行驶 -1 双右车道)
 尝试推导 (在车道中行驶 -1 双右车道) 的规则: NIL
 没有关于该目标的规则: (在车道中行驶 -1 双右车道)
 ④ 用户输入 : (在车道中行驶 -1 观右车道) = ((非 1000))
 置参数 : (在车道中行驶 -1 双右车道) = 非 CF -1000
 参数追踪结束 : (在车道中行驶 -1 双右车道)
 规则前提不成立 : (在车道中行驶 -1 规则058)

测试规则的前提	: (在车道中行驶 -1 规则144)
规则前提不成立	: (在车道中行驶 -1 规则144)
测试规则的前提	: (在车道中行驶 -1 规则145)
规则前提不成立	: (在车道中行驶 -1 规则145)
没有关于该目标的规则:	(道路 -1 在车道中行驶OK)
参数追踪结束	: (道路 -1 在车道中行驶OK)

图 9-8 利用“追踪”命令追踪逻辑流程

文的进入、被追踪的规则以及产生的判断等信息显示出来，你也可以将整个追踪过程打印出来，以备闲暇时研究。使用“追踪”可以使你发现系统逻辑“误入歧途”的症结所在。

图9-8是上一例的逻辑追踪过程，其中圆圈中的数字表示四个提问。追踪首先显示对规则R146的前提(如果部分)的测试。系统首先追踪第一个参数(属性)“车身距离”，并根据用户的回答，将它置为“5”①；然后，追踪第二个参数(属性)“实际车速”，并且按照用户的回答将它置成“40”②。由于这个输入数据使R146的前提(如果部分)不成立，因此，进一步测试R57的前提，此时，它追踪并获得用户对关于“拖挂”提问的输入“是”③，而这导致了R57的前提不成立。系统于是继续搜索下一条规则R58，并追踪属性“双右车道”，由于得到的用户输入是“非”④，因此，R58也不能激活。

图9-8的末尾说明，在规则簇“在车道中行驶OK”中再也没有规则可供追踪了。由于缺少了一条必要的规则，导致系统在该规则簇中得不到结论，而这正是使系统的整个咨询过程陷入迷途的症结所在。将图9-7中给出的规则加入规则簇“在车道中行驶OK”中，这个问题便迎刃而解了。

这里介绍的仅是一种极为简单的情况，“追踪”命令实际上可以用在更为复杂的情况下，因为它能准确地指出逻辑错误究竟在什么地方。

9.4 检查可信因子

有时，规则的逻辑也许没有毛病，然而，如果从可信因子的角度来看就不一定如此了。如同对规则逻辑一样，对于可信因子也必须按照一定的方法进行检查。必须对规则，特别是那些其则部分的CF小于100%的规则，进行检查，其中包括进行独立的和与其它规则联合的检查。

9.4.1 检查可信因子CF

如第七章所述，一条规则输出的CF是则部分的CF与如果部分CF有效值的乘积。若如果子句是以肯定方式陈述的，而如果子句是由“与”联结的，则如果部分的CF有效值是所有输入CF的最小值；若如果子句是由“或”联结的，则如果部分CF的有效值是所有输入CF的最大值。

对于以否定方式陈述的如果子句，CF有效值取决于如果子句中引用的值谓词是“非”(!) 还是“似乎不是”。如果是“非”，系统不传递输入的CF，而假设其为100%；如果是“似乎不是”，则让输入CF“通过”。

如果推理流程经过若干条确定性小于100%的规则，那么最终结论的CF就可能显著地下降。将结论确定性程度的这种下降与你的主观经验和判断进行比较，从而决定是否应该修改CF或更换谓词。

9.4.2 一般检查方法

检查CF的方法不同于检查规则逻辑时采用的方法。在检查规则逻辑时，总是从目标簇开始，然后向深层展开检查，直到可观察事实。然而，对于可信因子的检查则恰好相反，从可观察事实开始，到目标结束。

检查规则逻辑与检查可信因子的另一个重要差别是，在做

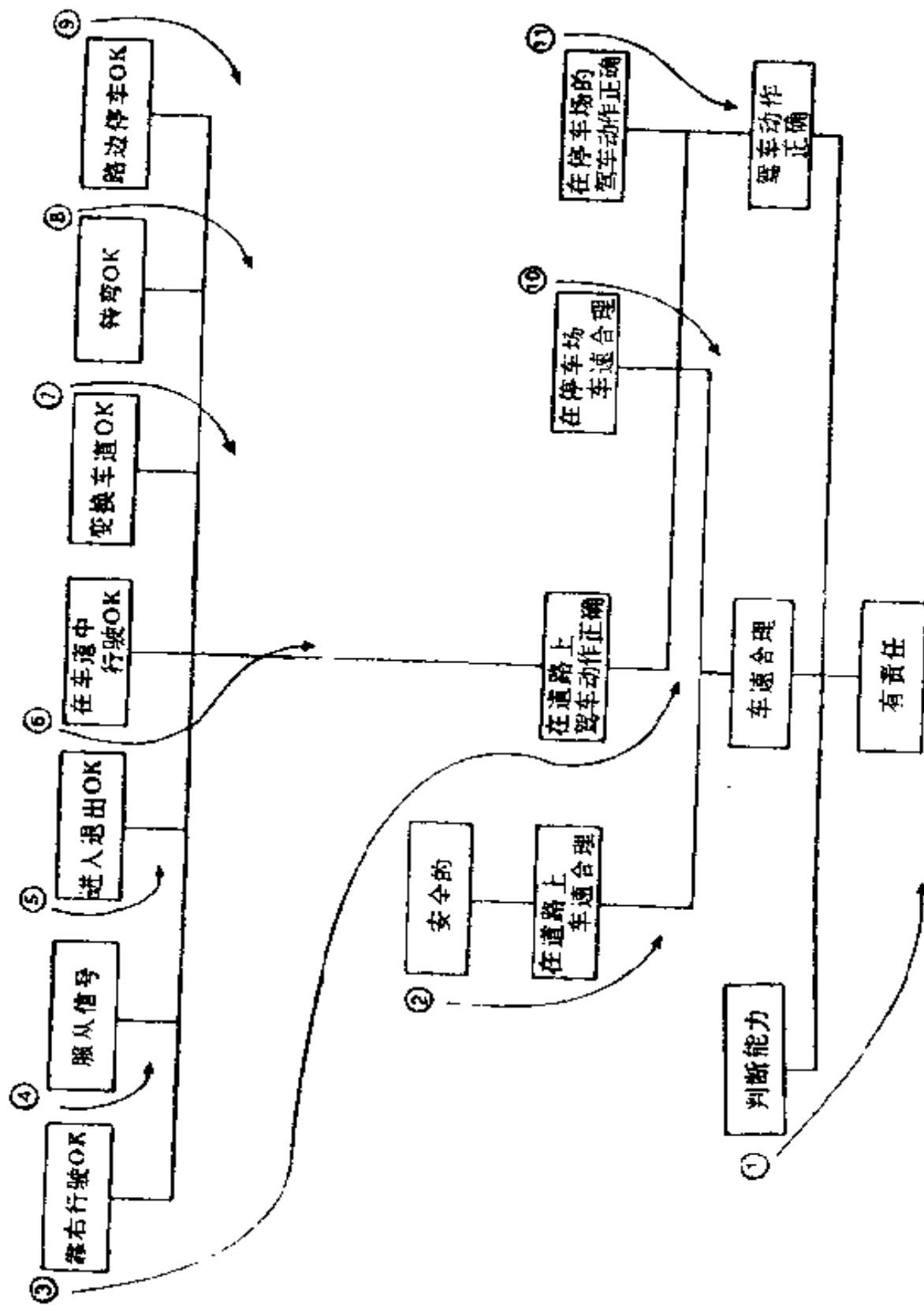


图 9-9 检查可信因子的一种顺序

后一项检查时，检查工作通常很难限制在一个规则簇中。这是因为当你给一个属性规定一个小于100%的CF值时，系统并不会因此而关闭“阀门”，而是继续搜索其它结论。

这也许是为什么要采用如图9-9所示的规则簇序列来检查CF较为容易的原因。为了分离出一个特定的规则簇序列，应该暂时去掉处在此序列中的每个属性的“提问优先”特性。例如，在检查序列③时，就应该去掉属性“靠右行驶OK”、“在道路上驾车动作正确”和“驾车动作正确”的“提问优先”特性。只要系统退出一个规则簇，你就可以使用“怎样推”命令查看相应结论的CF等情况。

按照这种方法检查CF，可获得的另一个好处是比较容易检查CF对规则逻辑的敏感性。

第十章 阶段IV：测试知识库

阶段IV的目标是产生实用的专家系统，使它能作出清晰、正确、容易理解和实用的咨询建议。到目前为止，你还只是从各个不同的侧面测试了知识库，因此，现在你必须把知识库作为一个整体来加以测试；此外，在阶段Ⅲ你只是从个人的角度测试了规则逻辑和CF两部分，现在必须换一个角度，即以一个用户的眼光来检查专家系统；最后，如果参加专家系统开发工程的还只限于你和你的同事们，那么，从现在起，你应该邀请一些有关人士对你的系统和工作提出客观的批评和建议。

评价整个知识库的全部特征的一个最好方法是，用知识库来实际解答一些应用案例。因此，阶段IV的主要步骤包括：

- 准备一套测试案例
- 评价知识库
- 改善人机接口
- “开门”评价

10.1 准备一套测试案例

准备一套测试案例，这些案例应该体现专家系统的设计能力。为了达到这个目的，你应该将注意力从专家系统的内部逻辑中解脱出来，转而认真思考那些用户可能遇到的问题。为了

使问题或专家系统将解决的问题概念化，你必须再次回到阶段I的主题上，即先准备一些实例的人机对话。然后扩充这些对话，直到准备好你的测试案例。

不同领域的专家所确定的测试案例不会相同，即使是同一领域的不同专家也不会得出相同的测试案例来，即测试案例的产生是因人而异的。由于这个原因，在这里我们不打算过多地论述应该如何定义测试案例，而只提供一些原则和可供参考的线索，以及一组用于交通事故原型的测试案例。

测试案例集应该包含足够的测试案例，以覆盖专家系统应该解决的问题。测试案例应按先简后繁的次序排列。简者只测试知识库中的一个部分，繁者一次就可测试知识库的几个部分，最复杂的案例则能揭示领域中多个主要成分间的相互作用。

在图10-1中，我们简要地列举了有关交通事故原型领域的测试案例集。与大多数实用领域不同，该原型仅是一个十分简单的、用于示教目的的领域，其中一个部分与另一个部分的相互作用很少。因此，这里列举的每个案例都是互相独立的。

1. 司机没有靠右行驶
2. 司机不遵守交通信号
3. 车祸发生在司机进入车道时
4. 车祸发生在司机退出车道时
5. 司机没有保持车距
6. 司机变换车道时太在意
7. 司机转弯动作不正确
8. 司机喝醉了酒
9. 司机在雾天或其它不良气候下驾车
10. 车速太快

图 10-1 关于交通事故原形的部分测试案例

10.2 评价知识库

在正式评价知识库之前，请首先仔细审查一下你在阶段 I 为专家系统确定的性能指标，并做必要的修改。

然后，按照这个经过审定的性能指标，运行测试案例。如果运行的结果与你的要求不符，请记录下整个咨询过程，修改知识库元素，然后再次运行测试案例，并注意对知识库的修改是否对运行结果有改进。利用在第八章中介绍的调试命令可以找到系统的错误或缺陷。顺序地对每一个测试案例重复上述过程，直至获得满意的结果。

判断你所确定的性能指标能否达到，这必须执行大量的咨询过程才行。由于这个原因，我们应该从考虑影响每种性能指标的准则入手。由于具体的准则都是因领域而异的，所以，下面我们只就每种知识库性能指标的一般性准则进行论述（见图 10-2）：

- 逻辑完备性
- 效率
- 逻辑正确性
- 可维护性

10.2.1 逻辑完备性

评价专家系统逻辑完备性的三个一般性准则是：

- 能否解决预定的问题？
- 是否使用了实际领域中的全部概念？
- 是否拥有适当层次的细节信息？

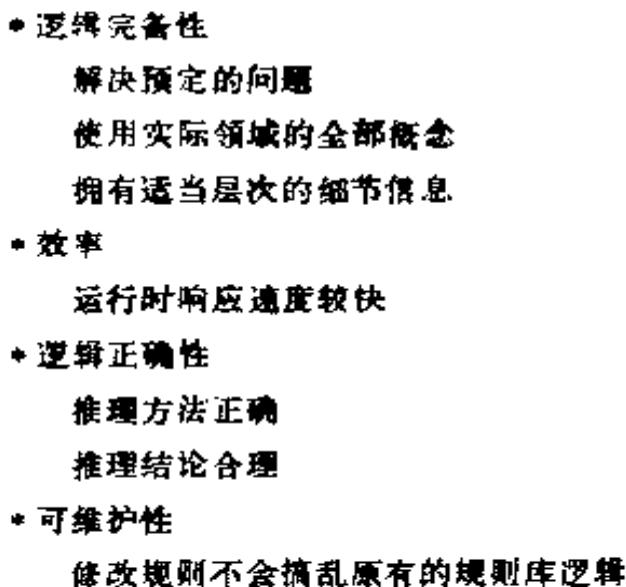


图 10-2 评价知识库的一般准则

能否解决预定的问题？ 专家系统能否解决你所设想的那些问题？是否包括了所有应该解决的问题方面？是否包含了多余的领域信息？能否解决实际问题？诸如此类的问题往往是由用户提出来，并要求予以解决的。总之，一句话，现在的这个专家系统是不是工程开始时你所设想并决心实现的那个专家系统？

虽然，本书介绍的交通事故原型主要是为示教目的建立的。但如果按交通法规的要求对它作进一步的补充、完善，并考虑到损伤评判和双方司机责任的分担等内容的话，它完全有可能用于保险公司的事故责任分析。也许还能评价两位司机（如果车祸发生在两辆机动车之间）各自的责任或司机与行人（如果事故发生在机动车与行人之间）各自的责任。在建造这样的专家系统时，把目标属性“有责任”设计成是非型属性显然是不行的，而应该给事故双方赋予相应的责任水平。

是否使用了实际领域中的全部概念？ 专家系统必须是完备的，这里的完备性不是理论意义上的，而是就实际领域中的概念而言的。尽管理论认识是经验知识的精髓，然而，在实际

应用中，完全没有必要将理论的词藻生搬硬套地凑到规则中去。当然，实际问题中依赖于理论的因素仍应该在规则中得到体现。

在交通事故专家系统中，“判断能力”可能受各种心理因素的影响。然而，这些不是专家系统所关心的。从解决实际问题的角度来看，必须判断的是事故责任的归属，而不必深究包含其中的理性原因是什么。

那么，怎样判断是否使用了实际领域的全部概念呢？为了能更客观地考查交通事故领域，我们对汽车事故案例作了实际统计，发现原型中有几处与统计事实不符。在原型中没有考虑司机开车越过车道中心线（进入车道左部）、未系安全背带以及碰撞行人等情况。因而，若要把原型扩充成一个实用专家系统的话，肯定需要补充处理上述情况的规则。

是否拥有适当层次的细节信息？适当层次的细节信息是什么？可观察事实的适当层次又是什么？例如，以这种层次出发观察一个人，能否足以判断他是“清醒的”？研究表明，影响一个人大脑清醒程度的三个因素是司机的体重、饮酒量以及饮酒后持续的时间。图10-3是一个体重在63公斤到70公斤的人受酒精影响的情况，类似的图示适用于其它体重级别的病人。

你能够按照此图设计属性“体重”、“饮酒量”和“酒后时间”吗？能否为每种情况编写相应的规

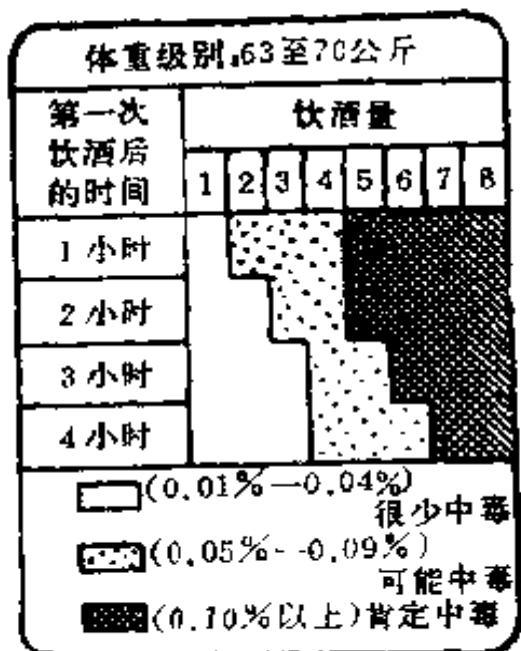


图 10-3 血液中酒精含量 (BAC) 与人的神志的关系

则呢？当然，这样做也许需要编写大量的规则。在本章的结尾，我们将介绍一种较为简便的方法。

10.2.2 效 率

我们所关心的效率主要是使用专家系统的效率。从专家系统开发者的角度来看，主要是考虑系统的响应时间，即对每个输入用户得等待多长时间？例如，有没有这种情况，有时系统好象自己“想心事”去了？若有，这种等待时间是否让人无法忍受？这种情况是否经常出现？

速度慢的原因是多方面的，有些是由于软件本身的性质引起的，但大多数主要是由于专家系统为处理某种情况而必须搜索的规则数目过于庞大造成的。若是这样，应利用“追踪”命令对知识库进行仔细的分析，以便通过改变规则库的结构来提高专家系统的响应速度。

10.2.3 逻辑正确性

逻辑正确性应从开发者和用户两个方面来看，确切地讲，就是两个准则：

- 推理方法是否正确？
- 推理结论在实际问题中是否合理？

推理方法是否正确？ 在这里，我们关心的一个重要问题是，建立知识库时采用的概括程度。即专家系统对知识的表示是太笼统，还是太具体？或是兼而有之？当浏览测试案例的执行过程时，你可能发现对于某些属性的提问不充分，而对另一些属性又过于细致。

例如，有关属性“有警觉性”、“清醒的”、“未服兴奋剂”以及“镇静的”的四个提问能否提供足够确切而具体的信

息，以便系统形成关于“司机”“判断能力”的结论。增加规则并考虑更为详细的属性也许是一个值得尝试的方法。例如，用新增属性“驾驶时表情严肃”来帮助“有警觉性”的判断；用属性“行为异常”判断“未服兴奋剂”。

以交通事故原型为例，它包含的细节还没有达到太详细的程度。然而，若我们想更细致一点的话，不妨设想一下，让规则簇“道路安全”依赖于规则簇“可见程度”，而“可见程度”又依赖于规则簇“气候”和“有障碍物”。如果把所有可能的气候条件和所有可能的障碍物类型都编进规则，那么仅这些规则的数目就可能轻而易举地突破数百。这是否值得呢？所有这些规则是给专家系统增辉，还是适得其反？

问题的本质也许还不是概括的程度。而是概括是否遗漏了某些必须考虑的特殊案例，相反，具体化是否把大问题弄得模糊不清。

推理结论在实际问题中是否合理？ 在阶段Ⅱ期间，经过从目标到可观察事实，又由可观察事实返回到目标的调试过程，你检查了知识库内部的逻辑完备性。然而，这只是对最初定义的推理结构进行检查。这种结构能否推出解决实际问题的结论呢？

在执行一个完整的测试案例时，结果是否集中到一个合理的结论上？知识库中的这一部分是否会产生与另一部分相对立的结论？不同部分产生的结论能否以逻辑的方式加以结合？

出现“鸡生蛋，蛋孵鸡”这样的逻辑循环是结合失败的例子。当然，在交通事故原型这种简单的系统中，这种逻辑错误不易出现。为了说明问题，在图10-4中，我们特地构造了一个这样的循环。假设另外增添三个规则簇：“道路安全”、“可见程度”和“有障碍物”。其中规则1、2、3分别作为每个

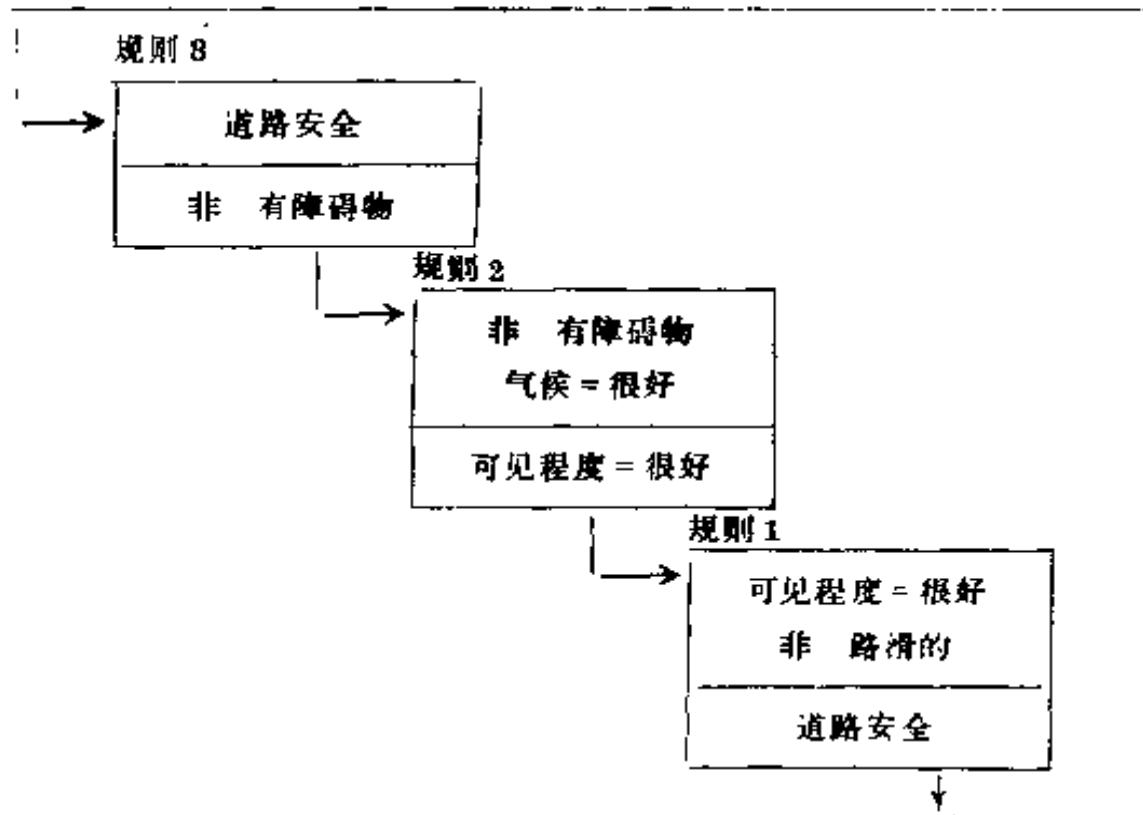


图 10-4 “鸡生蛋，蛋孵鸡”型逻辑循环

规则簇的代表。具体情况是这样的：“道路安全”这个条件的成立依赖于“可见程度”，“可见程度”又依赖于是否“有障碍物”。而按规则 3 的意思，“有障碍物”又依赖于“道路安全”，这就形成了循环。只要专家系统激活其中一条规则，它就会陷入永无止境的循环搜索之中。在这个例子中，问题的症结是非常明显的：规则 3 不应该再回过头去参照“道路安全”。在更为复杂的领域中，出现类似“鸡生蛋，蛋孵鸡”的逻辑循环可能没有这样明显。然而，通过仔细的追踪，这样的逻辑错误是不难排除的。

10.2.4 可维护性

你所建立的专家系统容易维护吗？系统的可维护性虽然与效率一样依赖于工具软件的特性，但构造规则库的方法对知识库的可维护性也有一定的影响。请考虑下面这些问题，当你向

知识库输入一条新规则时，它是否会搅乱规则库的已有逻辑？规则在规则簇中的次序对推理逻辑的影响怎样？等等。

上述问题对于事故原型来讲都不是十分重要，因为在该原型中，大多数情况都被有效地隔离了。然而，在一个复杂的领域中，许多似乎毫无联系的因素之间也许存在着微妙的关系，发现并揭示这些关系需要不懈的努力。

10.3 改善人机对话

在确认知识库达到性能指标之后，应该考虑的另一个重要方面是怎样使人机对话的性能达到标准。为了协调专家系统与终端用户之间的对话，可以从四个方面进行改善（见图10-5）。

- 阐明对话内容
- 使提问次序合理
- 使咨询建议实用
- 协调人机交互

图 10-5 人机对话的一般准则

10.3.1 阐明对话内容

在执行咨询时，你可能注意到对话显得比较呆板或词序不够正确。引起这些问题的原因主要与专家系统的“语言表述”机制有关，并不一定是逻辑错误，即其中大多数是由于给属性或上下文的“翻译”特性设置的文字序列不正确造成的。正如图10-6中所列举的那样，这里我们介绍其中的几个例子。

不同问题的表述相似 这种情况出现在你用相同的内容定义两个不同属性的时候，见图10-6A。在那里属性“在道路上车速合理”和“在停车场车速合理”的“翻译”特性是相同的，因此，对这两个属性的提示信息也必然相似。若发现这类问题，只要对相应属性的“翻译”特性作适当修改即可。

不自然的提问 在图10-6 B 中有一个用词不当的问句，“何时司机进入交叉路口是什么？”为了改变这种情况，你可以把属性“进入的时间”的“翻译”特性改成“司机进入交叉路口的时间”，如果此时你将该属性的“提示”特性设置为

属性	提问	改进
在道路上车速合理	司机的车速合理吗？	司机在这种路况下的车速合理吗？
在停车场车速合理	司机的车速合理吗？	司机在停车场的车速合理吗？
A. 不同问题的表述相似		
进入的时间	← 翻译 → 何时司机进入交叉路口是什么？	司机进入交叉路口的时间是什么？
B. 不自然的提问		
在车道中的相对车速：相对于同一车道上的车流，汽车在车道上的相对车速是什么？	增加“再提示”特性：	
较快	为了避免车祸，最好是与同一车道中的其它车辆保持相同的车速，你的可选答案如下：	
相同	较快 - 比其它车辆快10公里/小时	
较慢	相同 - 与其它车辆速度比，正负不超过10公里/小时	
	较慢 - 比其它车辆慢10公里/小时	
C. 含混不清的提问		

图 10-6 改进人机对话

“T”，意即由“电脑顾问”根据“翻译”特性自动生成问句：“司机进入交叉路口的时间是什么？”这虽然仍算不上规范的问句，却比修改以前容易理解多了。若要使专家系统给出的问句

自然通顺，可以在“提示”特性中直接给出一个完整的问句，例如，“司机什么时候进入与另一司机相会的交叉路口？”

含混不清的问题 对于系统给出的有些提问，由于用户不理解问句的意思或不知道可供他选择的项目是什么意思，因而在回答提问时会出现困难。例如，在图10-6C中，“较快”、“速度相同”和“较慢”是什么意思？即速度是相对于谁而言的？这时，如图所示可用“再提示”特性进一步说明，“再提示”中的内容在用户按下“求助”键后显示出来。

10.3.2 使提问的次序合理

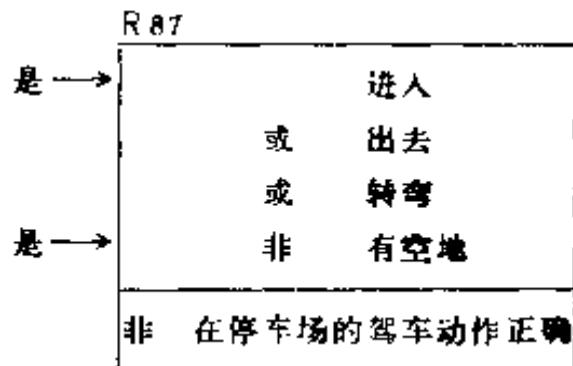
专家系统的提问次序应该让用户能够接受，同一问题不要重复提问，也不要提问多余的问题。所有问题都应该按照精心安排的计划，逐步给出。

下面是专家系统提问中的几个不合理现象。

重复提问 被要求反复回答一个相同或相似的询问是令人厌倦和反感的。你可能会遇到这样的情况，询问“波尔有警觉性是真的吗？”在整个咨询过程中重复出现，你不知道为什么会出现这种情况，于是按下了“求助”键，这时系统告诉你，它正在试图确定属性“有警觉性”、“清醒的”、“未服兴奋剂”和“镇静的”，以便求证属性“判断能力”。第一个询问显然是关于属性“有警觉性”的，那么第二个同样的询问是从哪里产生的呢？检查属性“清醒的”，我们发现它的“翻译”特性与“有警觉性”的相同。因此，必须对后一个属性的“翻译”特性进行修改。

提问顺序不合理 当系统以一种不合理的顺序进行询问时，也容易引起用户的困惑。在图10-7中，列举了专家系统进入上下文“停车场”后提出的一系列询问，以及用户的回答。在图的右方给出了使系统产生上述询问的相应规则。在这些询

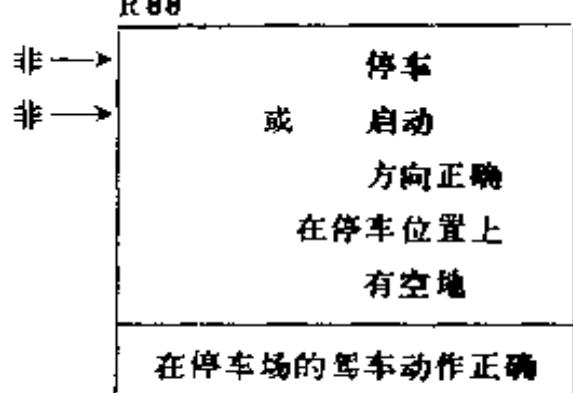
①波尔进入了停车场吗?



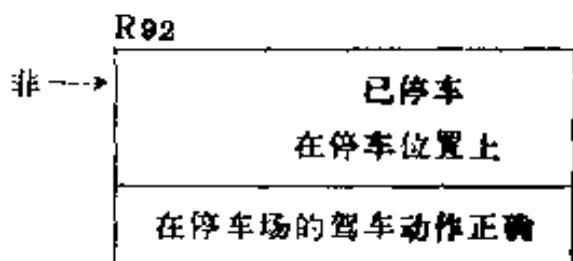
②有空地供汽车通过吗?

③波尔进入了停车位置吗?

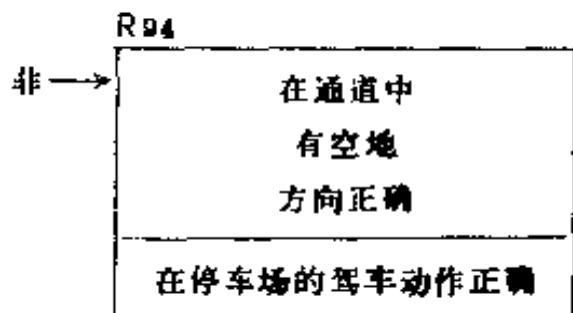
④波尔退出了停车位置吗?



⑤汽车已停车吗?

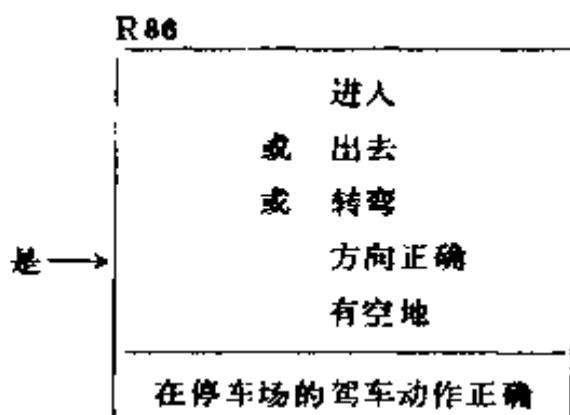


⑥汽车在通道中吗?



关。系统直到第 7 个询问才回到了主题, 即询问汽车是否沿允许行驶的。但随后的四个询问, 即询问 3、4、5 和 6 都与“进入”无关, 开始的两个关于属性“进入”和“有空地”的询问是合

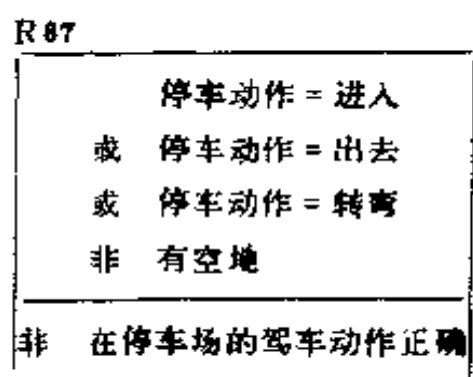
⑦汽车行驶的方向正确吗?



A. 问题

属性	单值	值
停车动作	∨	停车 启动 进入 出去 转弯 在通道中 已停车

B. 新属性：停车动作



C. 经过修改的规则

图 10-7 改善提问的不合理次序

的方向行驶。

如果你检查一下系统搜索过的规则，就会明白系统为什么提出这些不合理的询问。这是因为对于系统来说，各个属性在求值时是同等对待的，它不因为“进入”（停车场）为真，就自动地认为“已停车”为假。如果想让系统按较合理的思维方式推理的话，有一种方法，即定义一个新的单值属性“停车动作”，该属性在任一时刻仅取一个值，如“进入”、“出去”或“停车”等，见图10-7B。相应地，规则簇“在停车场的驾车动作正确”中的规则也要做必要的修改。图10-7C中正是这样一条经过修改的规则，即规则R87。

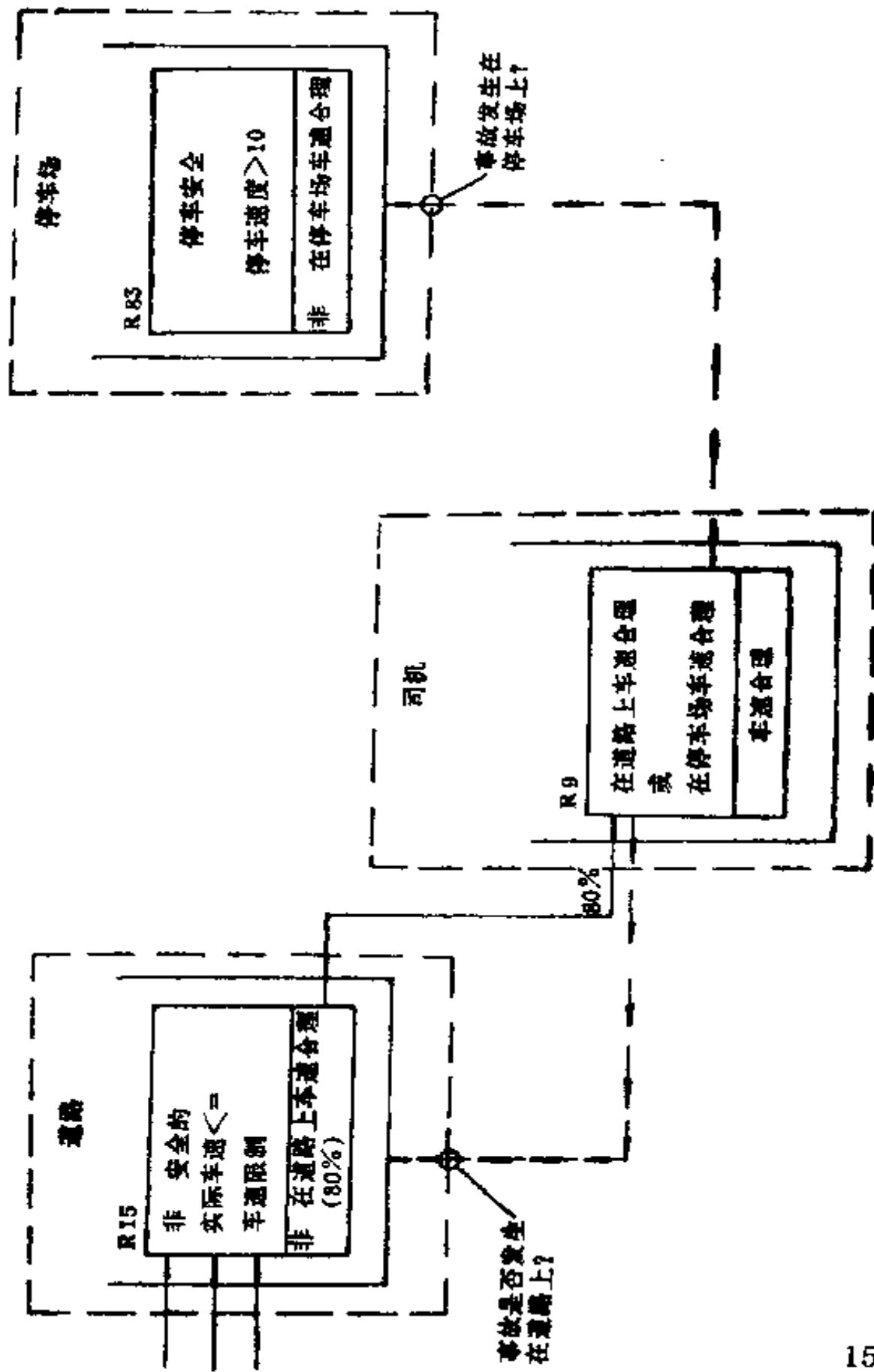


图 10-8 多余的提问

不必要的询问 若专家系统常提一些多余的询问，这显然也是令人厌倦的。请考虑这样的情况，当用户对提问“事故是否发生在道路上？”给予肯定的回答之后，系统接着询问了一些有关上下文“道路”的问题。但是，此后系统若突然冒出个“事故是否发生在停车场上？”的询问，用户会怎么想呢？

现实中，一起交通事故是不可能同时在道路上和停车场中发生的。在读图10-8之前，你也许不会明白为什么会出现这种怪现象。如图10-8所示，系统采用反向推理方法，从R9追踪到R15，但由于它必须先进入上下文“道路”，才能引用R15进行推理，因此系统在进入上下文“道路”之前，显示提示信息“事故是否发生在道路上？”当你回答“是”时，系统便进入上下文“道路”，并询问几个有关“在道路上车速合理”的问题。由于用户对上述询问的回答使系统激活了规则R15，并产生结论。然而，这个结论并不能使R9的第一个如果子句成立，于是系统继续追踪，它试图满足第二个如果子句。为了进入上下文“停车场”，因此，系统向用户显示提示“事故是否发生在停车场上？”

出现这种情况的原因是由于R9同时参照了两个相互排斥的上下文“道路”和“停车场”。为了确保系统一旦进入上下文“道路”和“停车场”中的任何一个，就不再试图进入另一个，我们可以在上下文“司机”中增加一些使用辅助属性的新规则，如图10-9所示。我们看到属性“车速合理”直接依赖于辅助属性DUMMY_AS的取值，而不是象以前那样，直接依赖于“在道路上车速合理”和“在停车场车速合理”。该辅助属性的取值又依赖于另一属性“位置”以及“在道路上车速合理”或“在停车场车速合理”。

在本例中，各事件的发生顺序是这样的：规则R9查找规则

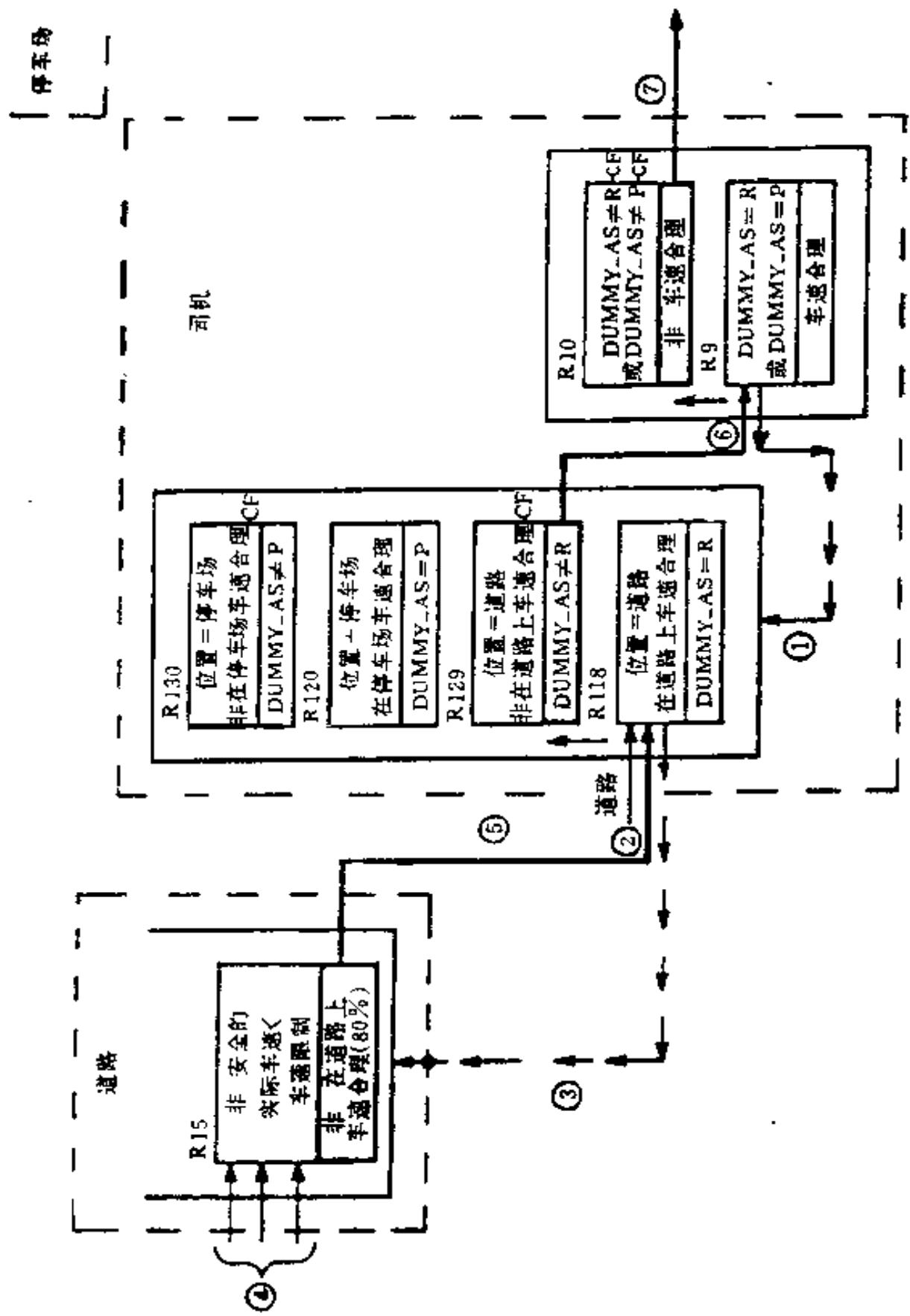


图 10-9 消除冗余提问的一种方法

簇DUMMY_AS①，并找到规则R118。系统求解该规则的第一个如果子句，并向用户询问“事故发生在什么地方？”，当用户回答“道路”②，系统继续追踪到规则簇“在道路上车速合理”③，找到R15并求证其如果部分④，激活R15⑤，此时虽然不能激活R118，却激活了它的下一条规则R129⑥。返回到规则R9，它的激活条件不成立，然而下一条规则R10却被激活了⑦。

在这里，关键的问题是，必须严格区别哪些上下文是互相排斥的，哪些上下文不互相排斥。图10-10是一个经过改进的交通事故原型的规则库图，从水平方向看，各上下文是互相排斥的，如进入“道路”就不能再进入“停车场”，进入“转弯”则不会再进入“路边停车”等。

10.3.3 使咨询变得实用

应该以最实用的形式向用户提供咨询。有许多技术可使咨询变得更为实用，例如将推理拟人化，对结论给出解释以及提供其它辅助信息等等。

在我们的示范原型中，可以将结论形式“司机-I”对事故负有责任，改成波尔对事故负有责任。为此，你必须定义一个属性“司机的姓名”，并通过“同义词”(SYN)特性将它与“司机”上下文联系起来（详细情况见附录A）。

另外，为了解释事故责任的归属，可能还得增加下述辅助结论：

- 波尔的判断能力很差
- 波尔的车速合理
- 波尔的驾车动作正确

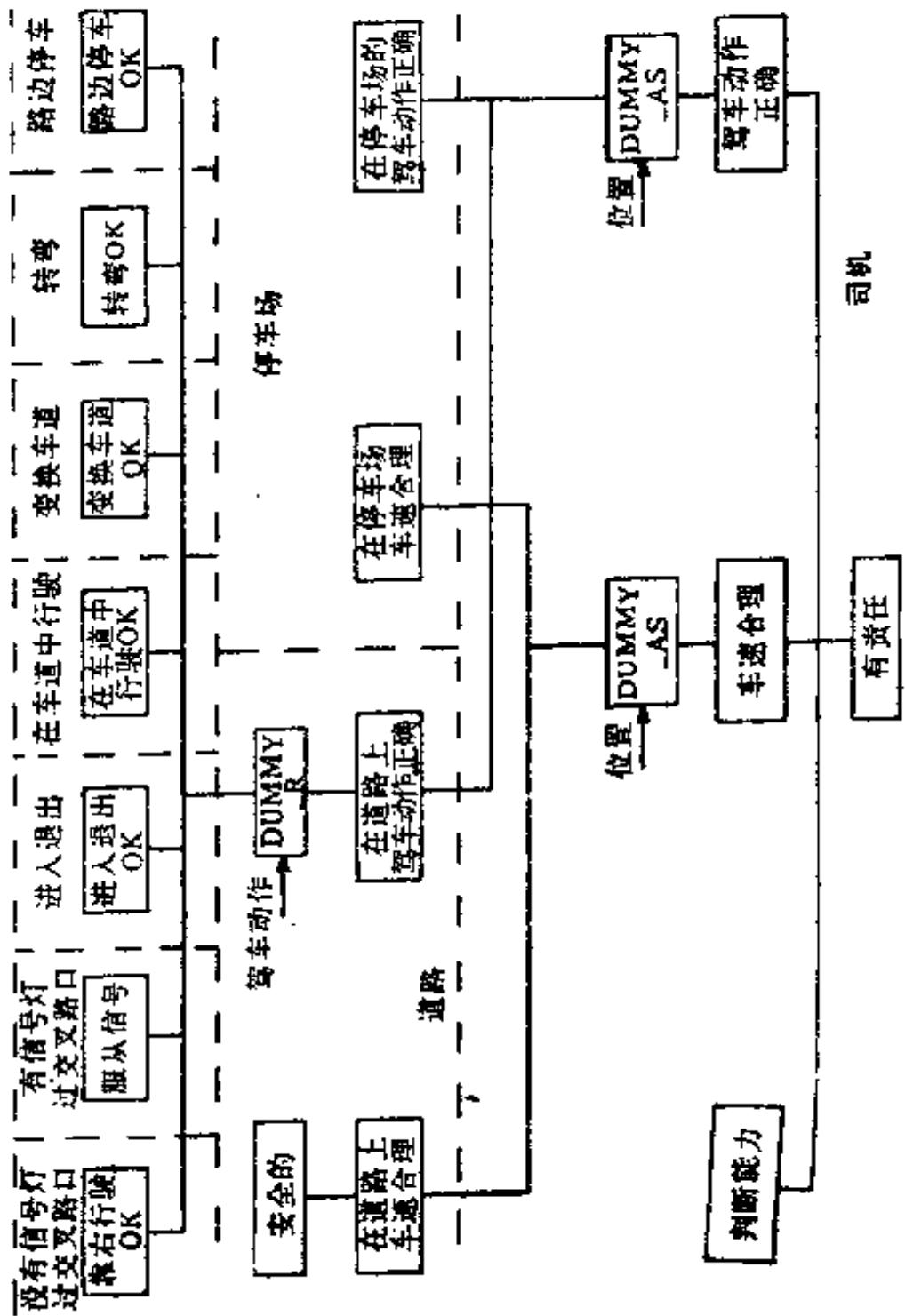
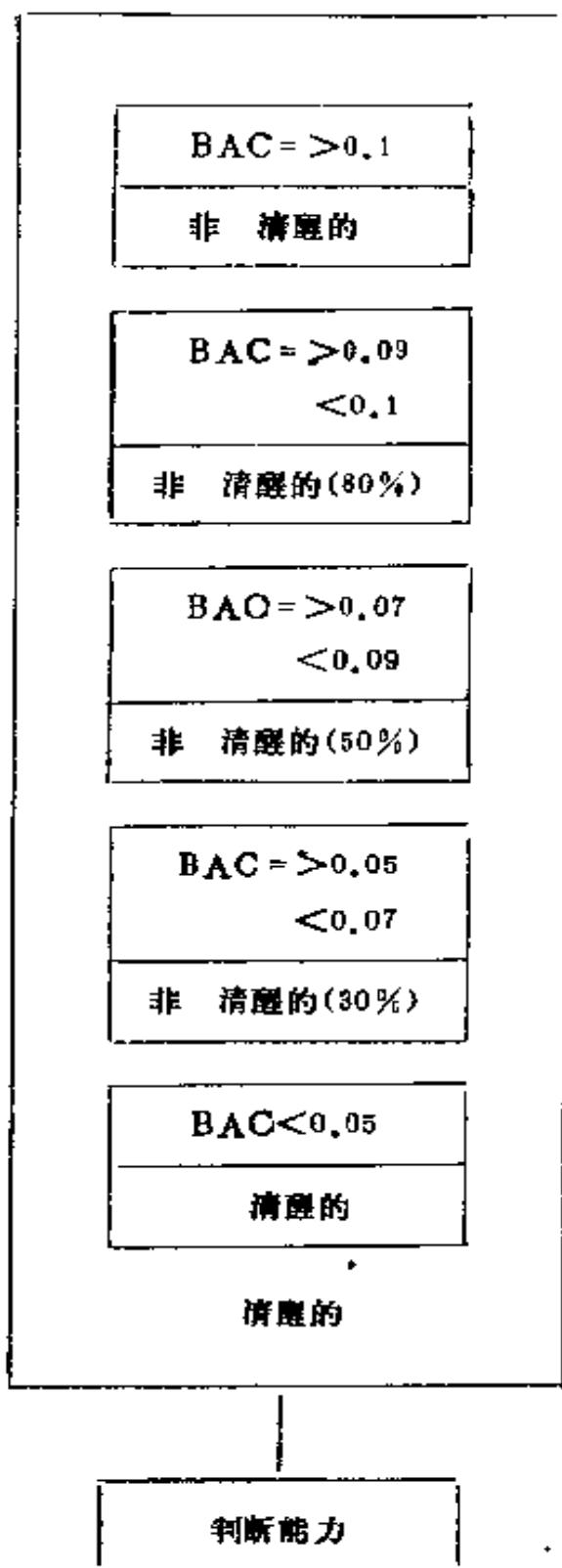


图 10-10 表示上下文相互排斥的交通事故原型规则库图



提示：

“约翰的血液中酒精含量(BAC)是多少(百分比)？”

再提示(按下“求助”键将显示的信息)：“如果司机只喝了少量酒或根本没有喝，则对BAC输入0，否则需要做BAC检查，并输入检查结果(百分比)。”

下面的图示说明BAC与清醒程度的关系：

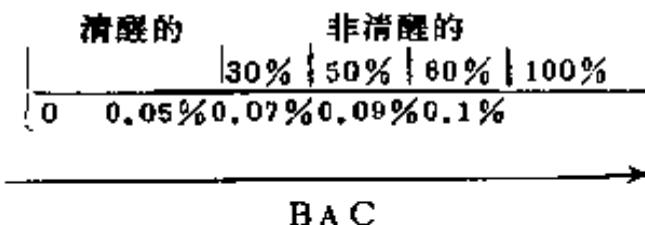


图 10-11 判断司机清醒程度的规则簇

做这件事情很容易，只要将上下文“司机”中的属性“判

断能力”、“车速合理”以及“驾车动作正确”设置成目标属性就可以了。

在推理过程中，显示一些额外的辅助信息以便说明咨询过程，这也不是一件困难的事情。譬如，当判断司机喝醉酒时，可以显示一条关于不准酒后开车的提示信息；而在另一场合，对于推理得出的结论或咨询可以给出详尽的解释。

10.3.4 协调交互作用

专家系统不但应该满足执行任务的用户，而且应该满足他所完成的任务。为了完成任务，专家系统也许要与数据库管理系统或电子表格等其它软件交换数据，也许要打开一个电子开关，或调节设备的某个部件，还可能帮助用户完成某项必做的测试等。

让我们再来看看属性“清醒的”。也许由用户自己做一个血液中酒精含量（简称BAC）的检查，并将检查的结果直接输入专家系统，比起写一大堆冗长的推理规则更为有效。在图10-11中，给出了一种具体的做法。这里增加了一个新的规则簇“清醒的”，它由五条新规则组成。用这五条规则可以表示图10-3中所示的BAC表。在图10-11中还同时给出了有关“BAC”的“提示”特性和“再提示”特性。其中“提示”特性用来向用户询问BAC的百分比，“再提示”特性则用来向用户解释为什么要这样做这种检查。

10.4 “开门”评议

尽管你在自己的专业领域中是很有造诣的，但仍不可避免会有这样或那样的漏洞，特别是在那些非专业方面。在你与你的研制小组合作开发专家系统时也会出现某些漏洞。因此，你应该敞开大门，邀请一些行家和一般用户来检查、评价或试用

你的专家系统。

应该邀请行家们来评价专家系统，特别是按图10-2中列出的性能指标来评价知识库。

也应该邀请一般用户来评价专家系统，特别是按照图10-5中给出的人机对话指标检查、评价专家系统。

第十一章 开发专家系统：一条 自我深造的途径

在最后这一章里，我们将全面地介绍建立一个实用专家系统所必须进行的工作。正如你所知，建立一个专家系统试验原型是件较容易的事情。然而，要建立一个完善、实用的专家系统却绝非易事，它需要周密的计划、丰富的知识和必要的人力、物力。

在着手进行专家系统开发工程之前，应该首先考虑保证专家系统成功的要求：

- 专家系统开发计划
- 经营管理领域的专家知识
- 专家系统开发工具
- 严谨、求实的工作作风

尽管这种工程是艰巨而耗时的，然而，由于参加这种工程活动本身肯定有助于你提高自己的专业技能，因此，相比之下你为此而付出的劳动还是非常值得的。

11.1 专家系统开发计划

图 11-1 介绍了专家系统开发计划的主要内容，它们是：

- 预研
- 建立试验原型
- 系统可行性研究
- 建立实用专家系统
- 系统维护

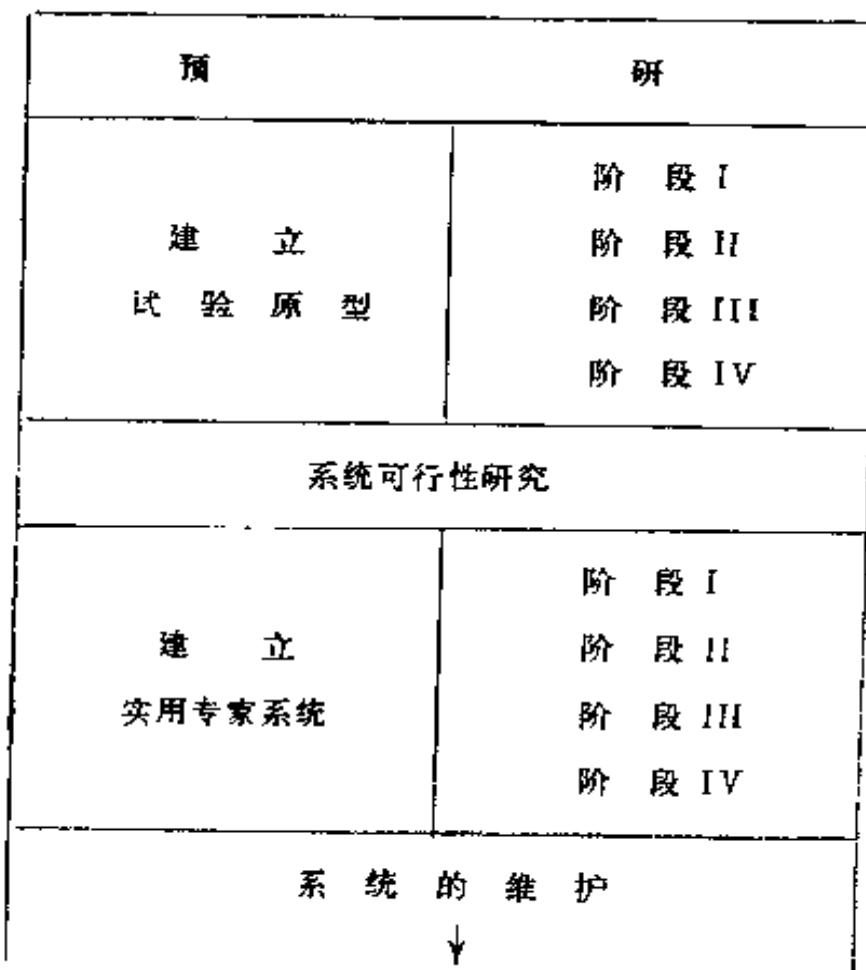


图 11-1 专家系统开发计划

11.1.1 预 研

预研的目的是决定要不要实施建立专家系统的工程，其中包含的工作有：

- 选择经营活动

- 判断所选领域是否合适
- 估计是否拥有必要的经验知识
- 选择领域的一个局部用于构造原型
- 分析收益与开销
- 制定工程进度表
- 选购专家系统开发工具

选择经营活动应用 如前所述（第二章），在企业经营活动的所有层次上，即计划、控制和执行层中，都有许多工作可以在专家系统的帮助下完成。我们在图 11-2 中重新进行了总结。

- 执行层：
 - 分析
 - 采购、生产、销售、推销、财务、法律
 - 综合
 - 系统配置、生产流水线、车间里机器的布置、加工方案、
 - 简单产品设计
- 控制层：
 - 机器
 - 测试、诊断、质量保证
 - 人员
 - 通知、动员、培训；工具和设备的使用、软件的使用、
 - 调度、会见、主持会议
 - 结算
- 计划层：分析
 - 财务、人事、生产、市场
 - 综合
 - 标准与规程

图 11-2 经营活动中的专家系统应用

判断所选领域是否合适 将你所选定的经营活动应用与已在第二章中介绍的专家系统领域的判别准则加以比较，从而判断你所选定的应用是不是专家系统领域。在图11-3中，我们重新总结了专家系统领域的判别准则。

- 有界的领域。
可以明确地划分该领域的范围
- 可分解的问题
该领域中的大问题可分解成若干具体的小问题
- 没有严格的求解方法
求解问题的方法不是算法式的（按照某个严格的公式或过程）而是启发探索式的（按照某种源于经验的过程）
- 不确定的关系和数据
关系无法完全预知，但可以用概率等方法描述其不确定性，数据也是不确定的，也可用概率描述
- 易变性
知识库建成之后，仍需要不断修正

图 11-3 专家系统领域的判别准则

估计是否拥有必要的经验知识 用已在第二章中介绍，同时在图11-4中加以概括的准则考查你在应用领域中的知识。

- 可确认的因素
能够确认影响问题求解的重要因素
- 可表述的关系
领域中的主要关系可用书面形式表述
- 可交流性
可以通过电话用二十分钟时间给用户解决一个有典型意义的问题

图 11-4 判断你是否拥有必要的经验知识

选择领域的一个局部用于构造原型 在应用领域中，选择容易确定并对它有透彻了解的重要部分进行原型开发。就求解

的问题以及应该提供的咨询而言，该原型在整个领域中应该是有代表性的部分。

分析收益与开销 可望得到的收益是否抵得上可能的开销？为了确定可望的收益应考虑下述问题：你是否确有应用需求？系统的确切目的是什么？专家系统怎样提高工作效率？它对工作效率有什么影响？它能帮助你完成哪些新工作？

为了确定可能的开销，要考虑另外一些问题：你是否有时间开发专家系统？有能够帮助你的同伴吗？它会影响你的其它工作吗？你能担负得起这种人、财、时、物的大开销吗？

制定工程进度表 如果你决定自己动手建立专家系统，那么就要为实现计划制定一个稳妥的工程进度表。

选购专家系统开发工具 在着手实现你的行动计划之前，先选购一种得心应手的专家系统开发工具。

11.1.2 建立试验原型

图11-5概括了本书的主要议题，开发专家系统的四个阶段是：

- I —— 确定专家系统领域
- II —— 开发表示知识的词汇
- III —— 建立知识库
- IV —— 测试知识库

11.1.3 系统可行性研究

系统可行性研究的目的是了解开发实用专家系统的各种需求。因此无论从哪方面看，它都是预研的继续，所不同的是，现在你可以利用在开发原型中获得的经验来指导你的论证和研究。

- 阶段 I. 确定专家系统领域**
 - 分割应用领域
 - 确定推理流程
 - 合成人机对话
 - 确定性能指标
- 阶段 II. 开发表示知识的词汇**
 - 确定属性
 - 定义属性的值
 - 编写检验词汇的规则
- 阶段 III. 建立知识库**
 - 准备书面规则库
 - 输入知识库元素
 - 检查规则的逻辑正确性
 - 检查可信因子
- 阶段 IV. 测试知识库**
 - 准备一套测试案例
 - 评价知识库
 - 改善人机对话
 - “开门”评议

图 11-5 专家系统开发步骤

从开发试验原型中获得的经验有助于你进一步确认系统的目标，范围以及知识的表示方式。利用这些经验还有助于你改进建立和调试知识库的方法，并且相应地调整工程进度计划。

主要注意力应该集中在使系统实用化上，实用专家系统的主要特征在全书的多处做了详尽的叙述，现概括汇总于此，同时见图11-6：

- 清晰的结构
- 完备性

- 正确性
 - 集成化
 - 可用性
 - 效益成本比高
 - 适应性
-
- 普通的结构
 问题和求解方法都有清晰的定义和描述
 - 完备性
 对考虑的所有情况都能提供正确的咨询
 - 正确性
 没有逻辑漏洞、冗余、矛盾或其它逻辑错误
 - 集成化
 最终的咨询是由若干分离的咨询集成的
 - 可用性
 咨询对终端用户来讲应该容易理解且实用
 - 效益成本比高
 效益超过成本
 - 适应性
 修改非常容易

图 11-6 实用专家系统的主要特征

11.1.4 建立实用专家系统

与开发试验原型一样，遵循开发专家系统的四个阶段。

11.1.5 系统维护

由于情况的不断变化，专家系统即使建成，也仍然需要周期性地加以修改。一般来讲，这种修改只需对知识库中的规则进行增补、删除或更换等操作就能做到。但应注意的是在每一次修改时，都要详细地记录对知识库的每一改变。这些记录在将来诊断专家系统时是必不可少的第一手资料。

11.2 经营管理专家系统领域中的知识

开发专家系统的关键困难不是如何编制计算机程序（尽管你偶尔可能需要求助于程序员的帮助），也不是专家系统开发工具的使用。虽然，你需要用一定的时间和精力来熟悉工具的特点并学习工具的使用，但一经掌握，它就再也没有什么神秘之处了。

自己开发专家系统的真正困难是，首先，要对你的经验知识的特点有透彻的了解，从而能用规则描述这些知识，最后，将它们存贮到知识库中去。工程刚开始时，你也许认为这是非常容易的事情。可当你研制原型时，就会发现它比你原来想象的要困难得多——你的思维过程必须比其它任何时候都更加确切而且严密。在你完成整个系统的开发之时，你甚至可能怀疑自己是否真象过去认为的那样，真正了解自己的思想和知识。

但是，你毕竟从建立专家系统的实践中增长了才干。在开发原型时获得的经验将有助于你发掘自己的知识，这其中或许包括许多以前连你自己也不曾意识到的细节。在这个过程中所获得的见识对于建立最终的知识库是大有裨益的。

通过开发原型系统，你可以获得丰富的经验。这些经验能加深你对知识表示问题的理解。知识表示也许是整个工程中最困难的任务，在图11-7中汇总了研制原型系统时应该考虑的知识表示问题，它们是：

- 你所选择的上下文
- 你选择的属性及其取值
- 你所使用的谓词和函数
- 你编写的规则
- 你给上述对象赋予的确定性因子CF

11.2.1 上下文

上下文是一个表示问题的参照系，而上下文树则象一个骨架，你编写的规则就装配在这个骨架上。应该怎样构思这个“骨架”呢？它应该描述细微还是应该表现粗犷？你应该把注意力集中于一个专门的侧面，还是应该考虑范围广大的问题？

你是只想接触问题的表皮还是想透过表面现象洞察其中的奥秘？从构造实用专家系统的目的出发，对问题的表示应深入到什么层次比较合适？

你的上下文是什么类型的？是表示物质成分的，还是表示观点概念的？究竟哪一种最适合于你的应用？

11.2.2 属性

属性用来定义你认为重要的各种因素，它们的取值则反映你是怎样对待差异的。

你的属性是描述型的，还是判断型的？“车速”是描述型属性，而“车速合理”则是判断型属性。“判断能力”显然也是一个判断型属性。判断型属性也许对事故原型已经够用了。然而，对于更实用的系统，如用于控制生产过程的系统，你可能更倾向于使用描述型属性。当然，在描述型与判断型这两个极端之间还可以有许多选择。在你的知识库中，你想选择哪种类型的属性呢？

属性应该是定性的，还是定量的？有些属性，如“美丽”，自然是定性的。但大多数属性既可定性表示也可定量描述。你可以给“判断能力”三个判断等级，即“很好”、“一般”和“很差”。也可以让它在1至100之间连续变化。究竟哪种表示方法对你最合适呢？

属性应该精细，还是粗略？“汽车安全”是一个粗略表示汽车状况的属性，我们可以用更为精细的属性代替它，如“刹

车”、“反光镜”、“轮胎”、“车灯”等。你可以用“道路安全”进行推理，但也可以用组成“道路安全”的诸因素如“可见程度”、“障碍物”，或低层因素如“气候”，或更低层因素诸如“雨”、“雪”、“雾”、“雹”等。在你的知识库中，需要哪个层次的属性呢？

11.2.3 谓词与函数

谓词刻划属性与其取值之间以及与其它属性之间的关系。函数则描述子句中的数学表达式和子句间的逻辑关系。换句话说，谓词和函数可描述事实以及事实之间的关系。你希望描述的关系是什么样的呢？

这些关系是简单的，还是复杂的？能否足以表达基本的比较判断？譬如，好的、不好、以前、以后、大于、小于等。另外，你需要用数学表达式、复杂函数或计算机程序来描述关系吗？

你想把这些关系描述成逻辑型，还是数值型？它们的性质是相近的，还是差异很大？若把“高度”规定成逻辑型的，并用是非型属性表示它，那么就可以用它来做定性比较：表示一栋房子是有“高度”，还是没有“高度”。如果你将“高度”看作数值型的，那么你就可以用它来比较不同建筑的“高度”，从而判断哪栋最高，哪栋最低，或者高度相同。

此外，关系还有绝对的和相对的之分，在绝对型关系中，你可以用一给定值与一个对象的属性进行比较；而在相对型关系中，则可用一个对象的属性与另一个对象的属性进行比较。如“实际车速”=6是一个绝对型关系，而“实际车速”>“车速限制”则是一个相对型关系。

显然，不同的关系可以满足知识库的不同需要，你能根据实际情况正确地使用各种关系类型吗？

11.2.4 规则

在你编写规则时，知识“板块”的构件以及这些“板块”之间的联系方式也就随之确定了。由于规则是可以增补，删除和替换的模块，因此，必须精心设计和选择规则中的这样一些构件，它们反映规则间相互作用的关系类型，从而影响整个问题的求解。

● 上下文 - 参照系

细微的 ~ 粗犷的
深层的 ~ 浅层的
详尽的 ~ 笔触的

● 属性 - 涉及的因素

描述的 ~ 判断的
定性的 ~ 定量的
精细的 ~ 粗略的

● 谓词和函数 - 表达事实和关系

简单的 ~ 复杂的
逻辑的 ~ 数值的
绝对的 ~ 相对的

● 规则 - 知识“板块”

构件：
——子句数目的多或寡
——概念层次的高或低
——概念覆盖面的大或小

推理：

——理论的 ~ 实用的
——专用的 ~ 通用的

● CP - 规则中的可信度

确定的 ~ 不确定的

图 11-7 知识表示任务的内涵

规则中的构件可大可小，包含的子句可多可少，涉及的概念层次可高可低，从如果部分到则部分覆盖的概念也可多可少，你个人认为选择怎样的构件合适呢？

除了构件，规则的类型也决定了推理方式和问题求解方式的性质。推理可以是严格遵循某种理论的，也可以是讲究实用的。你是强调基本理论，以解释事物发生的机制呢？还是只希望得到实用的咨询建议？或是既希望得到实用的咨询建议，又想在某种基本理论的支持下得出咨询？明确你设计专家系统的目的将有助于你做出上述判断。

推理方法可以是专用的，也可以是通用的。你是希望解决特定问题，还是想建立能概括许多现象的一般原理？每个专家系统都有自己的特点，你需要哪一种类型的呢？还是需要兼而有之的？

以规则为基础的推理可以是演绎的，也可以是归纳的。演绎推理导致确定性，即若如果部分为真，那么则部分肯定也真。归纳推理则导致不确定性，即如果部分为真，并不一定能得出则部分也真。另外，你对一条规则的主观经验可以用一个赋予它的CF值来表示。

11.2.5 可信因子CF

你能够肯定自己的规则总是正确的吗？或者说，你是否能给所有规则赋一个不确定程度？就你的经验而言，这些规则总是正确的呢，还是有些例外？如果你知道哪些规则不正确，自然就可以改写这些规则，使它们都变成正确的。但如果你不知道，就只能给这些规则以一个合适的可信因子。

你能用别的方法编写规则，来解决不确定性问题吗？如果你能够，那么就照你自己的办法去做。

11.3 专家系统开发工具

“电脑顾问”是目前软件市场上流行的几种专家系统开发工具中的一种。它是一种用户不需要编程就能使用的专家系统开发工具，在附录E中，我们将按高、中、低三档分别介绍同属这类系统的其它工具软件的技术特性。

在所有这些系统中，“电脑顾问”是出类拔萃的，尽管它也与所有的软件一样，具有自己的特殊规定。由于在建立专家系统时，你的思维必须与软件的推理协调起来，故比起其它应用程序，你必须花费较多的精力去了解和掌握专家系统开发工具的特殊规定。

在这里，有些规定产生了一些对系统使用的限制。下面我们将从几个方面考查“电脑顾问”的一些特殊规定(见图11-8)。

11.3.1 文 法

“电脑顾问”拥有形式化的文法。它采用模版表示规则，每条规则都必须隶属于一个上下文，同时它自身包含下述语言成分：函数、属性、谓词、值和 CF。其它专家系统开发工具也许采用更为自由的形式。然而，采用模版提供的结构会使你的注意力更能集中于思考问题。

11.3.2 推理方式

“电脑顾问”采用从目标到事实的反向推理方式工作，它进行的是一种称作“深度优先”的搜索过程：取规则的如果部分中的一个条件，并尽可能地反向追踪，直至遇到可观察事实。一旦对一个条件的跟踪过程结束，它便会以同样的方式追踪下一个属性。

反向推理使“电脑顾问”能够成功地处理分析型应用。那么处理综合型应用会怎样呢？有许多人可能推测综合型应用只

- 文法
 - 模版 ~ 自由形式
- 推理的方式
 - 正向推理 ~ 反向推理
- 处理不确定性的方法
 - 确定的 ~ 不确定的
 - 严格的 ~ 不严格概率演算
 - 专家控制 ~ 不控制CF的敏感性
- 人-机接口
 - 程序员的 ~ 非程序员的语言
 - 输入规则 ~ 输入例子

图 11-8 选购专家系统开发工具应考虑的主要因素

有借助正向推理才能完成。尽管有前件规则，但“电脑顾问”并不能按正向推理的方式追踪属性。事实上，“电脑顾问”可以用这种反向推理来处理综合型应用。

图11-9给出了用正向推理实现的风险投资应用，可将它与图11-10中的反向推理实现进行比较。在正向推理实现中，首先进入“投资者的财力状态”和“经济形势”上下文，它们分别使用可观察事实来确定投资者的财力状态和投资者所处环境的经济形势。在到达最后一个上下文“风险投资咨询”之前，系统将依次进入“风险投资数额”、“购买债券”、“购买股票”、“购买存款凭证”和“购买金融基金”。

与这种顺序方式相反，反向推理实现采用并行的工作方式。在图11-10中，“投资者的财力状态”和“经济形势”分别是上下文“投资者”的两个规则簇。在“投资者”的左右是其它上下文。上下文“投资者”通过五个目标：“建议（投资）数额”、“建议债券数额”、“建议股票数额”、“建议存款数

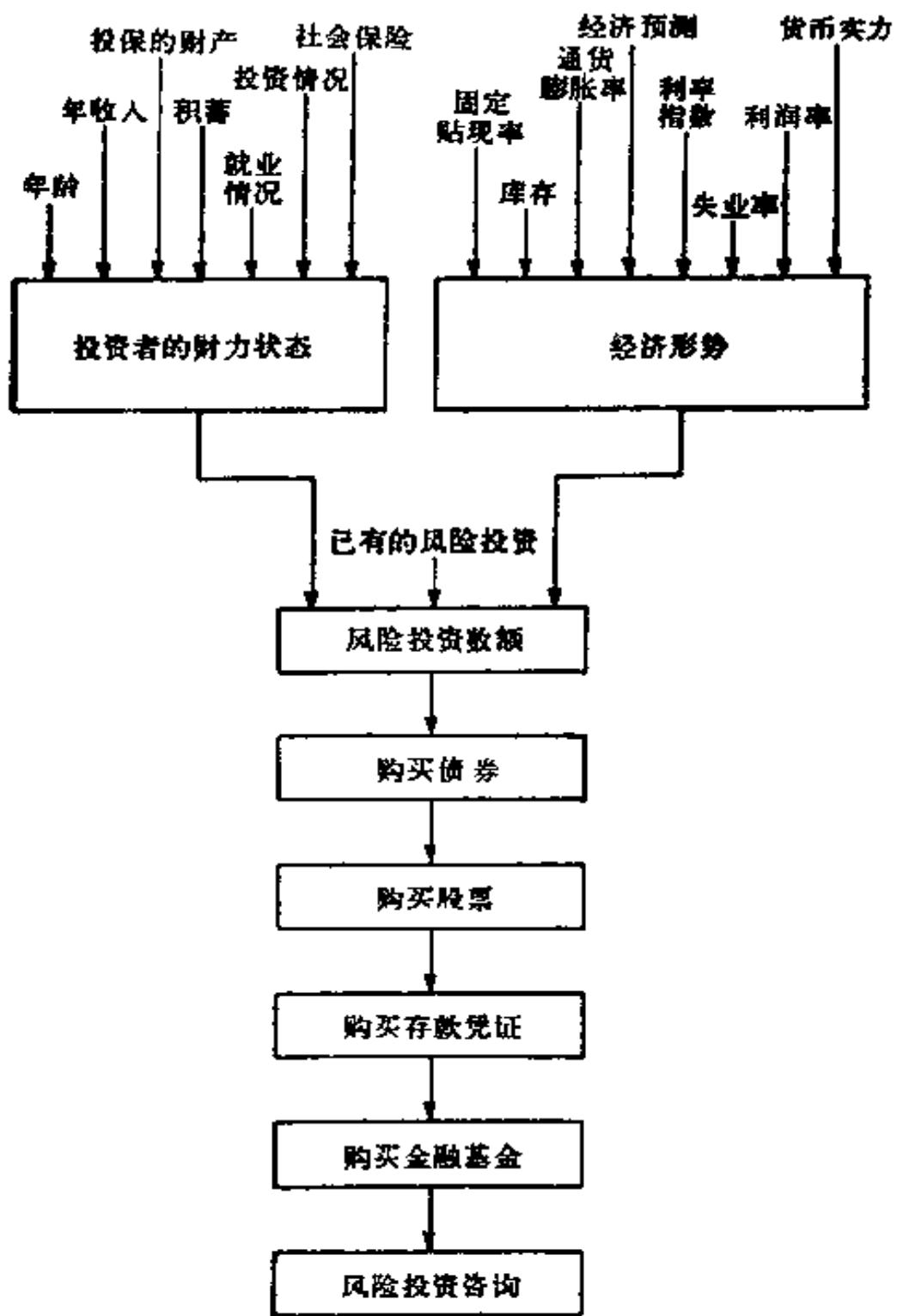


图 11-9 风险投资规则库简图——正向推演

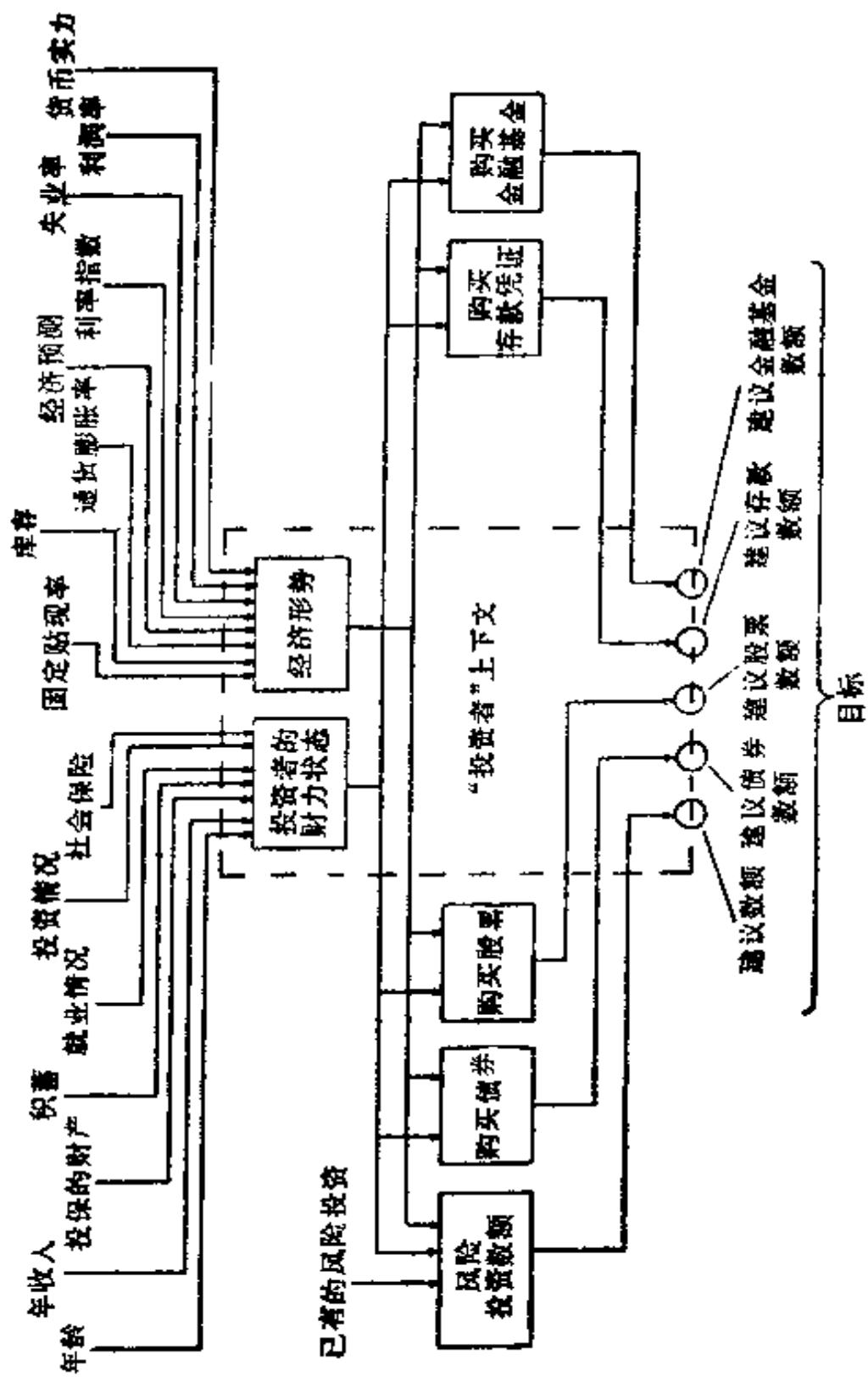


图 11-10 风险投资规则库简图——反向推理

额”和“建议金融基金数额”来控制整个推理过程。由于系统在推理中是逐个求证各目标的，因此，它实际上是一个接一个地追踪所示的并行路径。

例如，第一个目标是“建议（投资）数额”。为了求解这个目标，系统在逻辑上进行反向推理，先经由“风险投资数额”，再进入“投资者财力状态”和“经济形势”属性，直至达到可观察事实。在系统确定了这些属性的值之后，也就可以确定建议投资的金额了。然后，系统再以同样的方式追踪下一个目标。“建议债券数额”等。类似的方法也可用来解决配置型问题。

然而，如果你面临的是一个复杂的数据处理问题，即存在大量事实的问题，而且在这些事实之间存在着错综复杂的联系，那么，“电脑顾问”也许不是最理想的工具。

同样，如果你的任务是对反映实际情况的表格数据进行概括，提炼，那么，具有归纳推理功能的专家系统开发工具也许更能满足你的需要。

11.3.3 处理不确定性的方法

应该从哪些方面考查专家系统开发工具对处理不确定性的支持呢？可以从下述三个方面加以考虑：

- 能否表示不确定性
- 严格的还是不严格的概率演算
- 能否控制最终结论对CF的敏感性

能否表示不确定性 “电脑顾问”允许你建立这样的系统，其中所有的规则都可以是确定的，也可以是部分或全部都是不确定的。同时，还允许你说明属性的肯定或否定形式的信

任程度。

严格的还是不严格的概率演算 “电脑顾问”允许你说明每条规则的CF，也允许用户说明其输入事实的CF。但是系统在遍历知识库时形成CF的演算法则并不是严格按照概率论的规定的。

这种处理方法对于绝大多数应用来说，也许已经能够满足实用要求了。因为在多数实际应用中你根本无法知道准确的概率究竟是多少，因此也就没有必要追求演算过程的严格性。

如果有的应用要求必须使用严格的概率演算法则的话，那么，请不要使用“电脑顾问”。例如，你的目的是得到最佳诊断而不是一组概率值，那么，你必须采用比“电脑顾问”更为严格的方法，如贝叶斯推理算法等。

能否控制最终结论对CF的敏感性 “电脑顾问”允许专家控制最终咨询结果对CF的敏感程度，选择诸如“确定是”、“似乎不是”之类的值谓词可以控制最终咨询的CF，一般来讲，它们主要依赖于输入事实和已用过规则的CF。

当然，上述选择并不能替代精确的概率演算，只是供专家通过CF来调整专家系统的一种手段。

11.3.4 人-机接口

在建立和测试知识库时，你一定希望专家系统开发工具能够提供尽可能多的手段来帮助你，这是很自然的。可是，在实际上，必须考虑的基本因素只有两个：一个是你所使用的语言，另一个是你输入信息的方式。具体地讲，就是：

- 程序化还是非程序化的语言
- 输入规则还是输入例子

程序化还是非程序化的语言 绝大多数专家系统开发工具，特别是做正向推理的工具，都没有为非编程人员提供理想的人-机接口，这是因为使用这类系统必须具备一定的程序设计知识。使用“电脑顾问”可以在很大程度上避免上述问题。利用简写式规则语言(ARL)这种非常简便的工具，任何人都可以定义上下文、属性，甚至编写规则。

尽管如此，非编程用户使用“电脑顾问”时仍然存在着接口的问题。那就是当你要显示某条规则时，“电脑顾问”是以LISP语言的形式显示该规则，而不是以输入该规则时采用的ARL形式。

这的确是不方便之处，但你只要尝试几次，就完全可以掌握并解释这种LISP格式的规则，并不一定学习LISP的编程(详见附录B)。

在“电脑顾问”的后续产品“增强型电脑顾问”(Personal Consultant Plus)中，这种缺陷已被克服了。在这种高档软件中，你可以直接得到以ARL形式或自然语言形式显示的规则。

“电脑顾问”的另一个不足之处是，有好几个谓词和函数不能直接在ARL中使用，而只能在LISP中使用(详见附录B)。在“增强型电脑顾问”中这一缺陷也被克服了，即它们都有了相应的ARL形式。

输入规则还是例子 与许多其它做反向推理的专家系统开发工具一样，你必须将知识转变成规则，然后才能输入到“电脑顾问”中去。这种知识形式的转化过程需要耗费大量时间和精力。但这样做的好处是，通过亲自动手整理知识并编写成规则，你就可以知道自己到底了解什么，了解多少。

然而，常常有这样的情况，有些专家，他们虽然拥有丰富

的经验，但是由于某种原因，他们可能不愿意或无法将自己的经验和诀窍改写成规则。假如你正好就是这样一位专家，那么，你一定希望能有这样一种专家系统开发工具，它不要求你编写完整的规则，而只需要输入关于具体情况的事例就可以了。即这种软件会对输入的事例应用归纳推理以产生具有一般性的规则。

11.3.5 “增强型电脑顾问”

在这一节，我们将简要地介绍一下“增强型电脑顾问”。与“电脑顾问”一样，它可以在具有一个硬盘驱动器的IBM PC 及其兼容机上运行。所不同的是“增强型电脑顾问”只需要512K内存。

它的知识表示语言的语法与“电脑顾问”相似。在“增强型”系统中，用框架的概念取代了上下文。它的特点是：除了规则以外，框架还可以存储任何已编好的程序过程，包括图形程序等外部程序。

“增强型电脑顾问”使用了元规则，从而使它能进行更深一层的推理。

系统的人-机接口也得到了改善。它使用当前流行的弹出式菜单作为操作界面。更重要的是，所有的谓词和函数都可以在ARL中表示。仅此一点就使面向非编程用户的人-机接口大为改观。

11.4 严谨、求实的工作作风

在建立自己的专家系统方面，除了必须对自己所从事的工作领域有丰富的知识以外，在工作作风上，还有两个十分重要的要求，这就是：思维要逻辑严密，看问题要客观实际。逻辑严密就是要尽量避免容易产生的各种逻辑矛盾或循环；客观实

际就是，在做一件事情以前，必须考虑你所具备的客观条件以及能够得到的帮助。

11.4.1 避免常见的逻辑错误

从某种意义上讲，建立知识库不是别的，就是一项逻辑训练。应用选择表可以发现逻辑遗漏或冗余，甚至能找出矛盾的地方。然而，你若不掌握正确的思维方法，仍有可能出这样或那样的差错。因此，在开发专家系统之前，最好先读一点介绍形式逻辑的书，以便掌握和养成严谨的思维方法和习惯。

有些疏忽虽然算不上逻辑错误，但也可能导致不正确的结果。如忽视了属于问题领域或问题求解者的关键限制，这些限制可能包括由问题求解者的身体组织和心理因素引起的，以及由于疏忽、偶然性和情绪等引起的限制。所有这些都能够影响我们的感知和洞察问题的能力。应该仔细地鉴别这些限制并判断它们对知识库产生的影响。

具体的逻辑错误种类很多，在图 11-11 中，我们列举了其中的几个：

- 循环论证
- 过份概括
- 偶发相关
- 错误的推断
- 不成立的排中假设
- 不正确的类比

循环论证 当产生结论的命题反过来又依赖于结论时，就发生了所谓循环论证。这是一种类似于“鸡生蛋、蛋生鸡”的循环。在建立知识库时，合理地应用规则簇就能够很容易地避免

这种错误。

过份概括 过份概括的意思是仅从一组有限的证据就不充分地推出一个具有普遍意义的结论。从书面形式看，这种结论往往表现在有诸如“所有的”、“没有”、“总是”或“从不”等出现的描述中。在知识库里，对规则中使用的CF以及谓词类型进行仔细分析有助于排除过份概括。

偶发相关 仅仅知道在两件事物之间存在某种统计相关，并不能断定其间就存在因果联系。例如，雷声总是伴随着闪电的，但雷声并不是由于闪电而产生的。它们都是由同一种电学效应引发的。

错误的推断 错误的推断是假设昨天成立的事物延续到明年仍然成立，即过去的趋势必定继续。事实上，这并不是必然的。例如，昨天下雨，明年的这一天还肯定会下雨吗？在下结论之前应该谨慎地考虑所有相关的因素。

不成立的排中假设 许多人在论证中使用了不成立的排中假设，他们把所有可能的抉择都说成是非此即彼的，并请你从中选择。这种谬误也称为“非黑即白”谬误。为了避免这种常见的思维陷阱，需要努力发现除显而易见以外的抉择。

不正确的类比 许多推理是建立在类比基础上的。不正确的类比就是不合适的比喻。有一个顽童证明蜘蛛用腿听话的古老传说很能说明问题。故事是这样的：有名顽童先将蜘蛛放在桌子上，并命令它跳跃，此时蜘蛛果然跳跃了，然后，他用刀把蜘蛛的腿全部砍掉，再放回桌子上，并再次命令它跳跃，结果是显然的，蜘蛛“证明”了顽童的观点——它听不到跳跃的命令了。

当然，日常生活中的大多数不正确的类比没有这么明显，但也不是绝对没有。你必须经常检查你所编写的规则，才能避

- 循环论证
避免“鸡生蛋、蛋生鸡”循环
- 过份概括
只从有限证据不充分地得出普遍结论
- 偶发相关
统计相关并不揭示因果关系
- 不正确的推断
时过境迁，昨天正确的，明天不一定正确
- 不成立的排中假设
努力寻找“非黑即白”以外的可能抉择
- 不正确的类比
除开本质的，两个不同事物之间可能存在许多相似之处

图 11-11 几种逻辑谬误

免出现不正确的类比。

11.4.2 客观务实的态度

除了对关键限制的影响有足够估计和尽量避免陷入逻辑谬误以外，在决定是否需要下述人员的帮助方面也应尽量采取客观实际的态度：

- 同事们的帮助
- 知识工程师的帮助
- 用户的帮助

同事们的帮助 你有没有这样的自信：除偶尔需要向同事请教之外，你对领域中的任何问题都有解决办法？如果在完整的专家系统中全部规则的数目预计超过 300 条，那么，最好的

办法也许是成立一个专家小组，该小组成员在有关问题领域的知识方面能做到互相弥补。

知识工程师的帮助 当无法决定是否应该请一名知识工程师帮助时，最好先自问这样一个问题：你是想透彻地了解自己的领域，以至愿意多花费一些时间深入地研究情况？还是想请一位知识工程师以便节省时间？

如果系统的规则数目将超过 300 条，那么，你很可能需要一名知识工程师同你一道工作。不过，届时将有一个问题是：由谁来负责？是你还是知识工程师？

用户的帮助 在你有把握确信自己解决问题的方法是正确的之前，你能否听到来自用户的意见？在建立实用专家系统过程中是否需要倾听他们的意见和忠告？用户的意见对于判断整个系统是否可用是非常重要的。

11.5 开发自我

也许你不是直接为了工作目的而需要专家系统，也许你还不是专家，也许你对本行业的了解仍不出众，但是，建立专家系统还是有价值的。如果你想透彻地了解自己工作的领域，如果你想在业务上显露头角，如果你想成为专家，那么，建立专家系统可能是达到上述目标的一条途径。

也许你已经认识到了这些。为了建立专家系统，你必须寻找一种合适的知识表示形式，它迫使你思考究竟什么是问题的本质；你必须编写能够概括领域中基本原则的规则，因而使你的思想集中于问题的求解；你必须分析自己对各种概念和原则的掌握程度，它使你能从多个不同的角度和层次认识问题。

假设你开发了一个专家系统，尽管它不能出售或供他人所用，但对你来说，也仍然是有所裨益的。当你修改旧规则，向

知识库中输入新规则，并用问题实例来测试这些新规则时，你自己也随之增长了知识。这种主要从亲身经验中获得新知的学习方法是一种最为有效的学习方法。

无论你开始时只懂得多少，经过一段时间与专家系统的“对话”，你真有可能成为名副其实的专家。

尝试建立自己的专家系统吧！你会从中得到乐趣，并在乐趣中成为专家。

附录A “电脑顾问”的命令——ARL形式

A.1 系统定义的元素

A.1.1 算术运算符

下面各运算符的优先级逐行递减，利用括号可以改变优先级。

-	变号
*	乘
/	除
+	加
-	减

A.1.2 谓 词

值谓词 检查前提（如果部分）中参数（属性）取值的确定性程度。

谓词	自然语言翻译	CF范围
IS DEFIS	“属性”确实是“值”	1000
=	“属性”是“值”	201~1000
IS NOTDEFIS	“属性”是“值”值得怀疑	201~999
IS MIGHTBE	有证据表明“属性”是“值”	-200~1000
IS DEFNOT	“属性”确定不是“值”	-1000

=	“属性”不是“值”	- 1000~-200
IS NOTDEFNOT	“属性”不是“值”值得怀疑	- 999~201
IS THOUGHTNOT	有证据表明“属性” 不是“值”	- 1000~-201
IS VNOTKNOWN	不知道是否“属性”是“值”	- 200~200

对于是非型属性，不用短语是“值”或不是“值”。另外，=和 != 在动作（则部分）中的使用方式类似。

数值谓词 从数量上将一个属性与其取值或与别的属性的取值联系起来。

谓词	翻译
LT或<	“属性”小于“值”
GT或>	“属性”大于“值”
GE或>=	“属性”大于或等于“值”
LT或<=	“属性”小于或等于“值”

参数（属性）谓词 检查前提（如果部分）中一已知属性的确定性程度，但不管其取什么值。

谓词	自然语言翻译	CF范围
IS DEFINITE	已知属性是确定的	1000
IS KNOWN	已知属性	201~-1000
	对于是非型属性的非	-1000~-200
IS NOTDEFINITE	不知道属性的确定性	-999~999
IS NOTKNOWN	不知道属性	-1000~200
	对于是非型属性	-200~200

A.1.3 函数

联结各子句或显示动作(则部分)中的正文。

函数	如果部分／则部分	意 义
AND	在如果部分中	所有子句都真才激活规则
	在则部分中	所有子句都被执行
OR	在如果部分中	只要任何一个子句为真就激活规则
SPRINTT	在则部分中	当规则被激活时，将引号中括起来的正文显示或打印

A.1.4 可信因子

这里介绍出现在则部分中的CF的意义。

CF范围	自然语言翻译(当用在规则中时)
1000	可确定(100%)“属性”是“值”
800~1000	有很强的证据(CF)表明“属性”是“值”
500~800	有证据(CF)表明“属性”是“值”
0~500	有微弱证据(CF)表明“属性”是“值”
-1000	可确定(100%)“属性”不是“值”
-1000~-800	有很强的证据(CF)表明“属性”不是“值”
-800~-500	有证据(CF)表明“属性”不是“值”
-500~-0	有微弱证据(CF)表明“属性”不是“值”

对于是非型属性，若表示“是”则去掉短句是“值”，若表示“非”则将“属性”不是“值”换成“非”“属性”。

A.1.5 变量

为了改善人机对话，系统提供一种称作“领域变量”的变量。它可以出现在属性的“期望值”特性中。用户可以对它进行修改。

“领域变量”	用途
\$ \$ TITLE	通过SPRINTT函数建立标题屏幕
\$ \$ IE-MESSAGE	输入当一定条件成立时在提示行中显示的信息，下面是系统提供的缺省值。
\$ \$ IE-FIXP—	“输入一个定点数”
\$ \$ IE-NUMB—	“输入一个数”
\$ \$ IE-POSONUMB—	“输入一个正数”
\$ \$ IE-STRING—	“输入你的回答”
\$ \$ IE-ASK ALL—	“选择所有可能的回答”
\$ \$ IE-VALUE—	“选择下选项目之一”
\$ \$ IE-HELP—	“（按F3键：求助）”

A.2 开发者定义的元素

A.2.1 上下文

操作类型	特 性	意 义
描述	ASSOCWITH	所有的父辈上下文。
	OFFSPRING	上下文的所有子辈。
翻译	DISPLAYRESULTS	显示“目标”的结果。
	PROMPTEVER	进入本上下文时，在显示屏幕上将显示的正文。
	PROMPT1ST	给用户的提示信息，系统

	PROMPT2ND	将按用户的回答判断是否进入该上下文。
	TRANS	给用户的提示信息，系统根据用户的回答判断是否再一次进入上下文。
	PRINTID	判别此上下文是否被进入了1, 2, …次。
	SYN	可选特性，上下文的正文形式标识符。
	UNIQUE	与SYN合用，以减少不必要的正文。
		注意：请参见后面的例子，看怎样利用PRINTID, SYN和UNIQUE将属性与上下文关联起来。
联系	INITIALDATA	当进入此上下文时，请求用户提供各属性的值。
	GOALS	当进入此上下文时，将对所说明的属性逐一进行求值。
	PARMGROUP	与此上下文关联的所有属性。
	SUBJECT	与此上下文关联的规则集的名称。
系统维护	CONTEXTTYPES	全部上下文的表。
	RULETYPES	与此上下文关联的所有规则集的表。
	TREEROOT	根上下文。

A.2.2 参数(属性)

操作类型	特 性	意 义
翻译	TRANS	用于规则和提示中的自然语言翻译(见下面的例子)。出现在TRANS中的下述动词的否定形式由系统自动地进行语法变形获得： 是(are, was, were, is), 做(do, does, did), 有(have, has), 应该(should)。
	DICTIONARY	当用户提问“怎样推”时应列举的属性名称，可用的选择为： 属性——通常的属性名 INTERNAL——不显示字符串——引号(“”)中的正文。
	PROMPT	向用户请求回答时使用的句子(请见下面使用T和*的例子)
	REPROMPT	帮助用户回答提示(PROMPT)的自然语言解释信息，当用户在回答提示时键入“求助”命令则在

	屏幕上显示。
EXPECT	在“单值”和“多值一次询问”属性中使用，向用户提供回答提问时可供选择的值。
LEGALVALS	说明“多值”属性的所有合法取值，可包括下述值： ANY——用户可用任何数据回答； TEXT——在规则的则部分中使用。
MULTIVALUED	“多值”属性，它可以同时完全确定地取多个值，可包括： T——对每个可能取值逐个给予提示，用户可以“是”或“非”来选择（与LEGALVALS特性配合使用。）（请见下面使用T和VALU的例子。） ASK-ALL——“多值一次询问”，用户在一个提示下，一次选择所有合适的值（与EXPECT配合使用）。
逻辑控制 ASKFIRST	在开始追踪属性之前，系

系统维护	USED-BY	统首先向用户询问属性的值。
	UPDATED-BY	其如果部分中包含该属性的全部规则。
	CONTAINED-IN	其则部分中包含该属性的全部规则。
	ANTECEDENT-IN	其则部分中包含该属性，但并不推得任何值的全部规则。
	UPDATED-IN	其如果部分中包含该属性的所有前件规则。
		其则部分包含该属性的所有前件规则。

A.2.3 规则

操作类型	特 性	意 义
描述	PREMISE	表示激活本规则条件的子句，即如果部分。
	ACTION	表示本规则的结论或显示给用户的相应文本信息，即则部分。
联系	SUBJECT	本规则所隶属的“规则组”(RULEGROUP)的名称。
逻辑控制	ANTECEDENT	说明本规则只能用于正向推理。

A.3 命令

操作类型	命令名称	意 义
菜单	GO	启动人-机咨询过程。
	QUIT	终止系统的执行，但不保存此次执行的任何信息。
	LISP	将知识库翻译成 IQLISP 程序（除非你非常熟悉 LISP 程序设计，否则不要使用这个菜单项目）。
	PARAMETERS	对属性及其特性进行输入、删除或编辑等操作。
	RULES	输入、删除或编辑规则及其特性。
	CONTEXTS	输入、删除或编辑上下文及其特性。
	VARIABLES	输入、删除或编辑变量。
	FUNCTIONS	输入、删除或编辑用户定义函数。
	LIST	将所有属性和／或规则在屏幕或打印机上列出。
	SAVE	将当前知识库的内容转存到磁盘文件中。
	TRACE	启动咨询，并在每个询问点追踪逻辑流程。
	RECORD	启动咨询，并将咨询过程记录到磁盘文件中，以备后用。

	PLAYBACK	再现已用 RECORD 记录 的咨询过程。
	PLAYBACK & TRACE	再现已记录的咨询过程，同时追踪。
功能键	HELP	显示能够帮助用户回答询问的信息。
	WHY	用向用户显示被询问属性所在规则的方法，系统解释“为什么”询问该问题。
	HOW	用户键入此命令之后，系统给出一个属性表，以便向用户解释系统是“怎样推”得属性的当前值的。
	LIST	系统显示出属性或规则清单。
	EDIT	转向IQLISP编辑器。
	STOP	结束当前操作 并返回到“主菜单”。
	PP	美观化显示，以易读的形式显示规则的LISP代码。
	DONE	当用户按照一个显示表的提示输入多个项目时，按此键表示输入过程结束。
特殊键	INS	用于插入一个新增加的上下文、属性或规则，或它们的特性。
	DEL	用于删除一个上下文、属性或规则，以及其特性。

ESC	与STOP功能相同。
SHIFT/DEL	启动内存恢复功能，在你必须停止工作时使用。
BRK/PAUS	暂停屏幕显示。
SHIFT/PRNT	将文本输出从屏幕转向打印机。
ALT/PRNT	将图形输出从屏幕转向打印机。
CTRL/PRNT	将图形和文本输出从屏幕转向打印机。
HOME	光标移到显示表的第一个项目处。
←	光标移到显示表中的前一个项目处。
→	光标移到显示表中的下一个项目处。
↑	光标移到显示表中的前一个项目处，如果是水平表，则前移12个项目。
↓	光标移到显示表中的下一个项目处，如果是水平表，则后移12个项目。
从LISP错误处理程序返回	回到IQLISP的命令解释状态下，即得到IQLISP的提示符。
BUILDER	回到“电脑顾问”的控制状态。

A.4 实例

A.4.1 PROMPT特性的使用

是非型属性

属性	Trans	Prompt	Printid
FREEZING	(The temperature was at or near 32 degree F) (温度达到或接近华氏32度)	T	
询问:	Is it true that the tem- perature was at or near 32 degree F? 温度达到或接近华氏32度是 否属实?		
CALM	(* was calm enough to pay attention to his dri- ving) (* 相当镇静足以安全驾车)	T	DRIVER-
询问:	Is it true that DRIVER-1 was calm enough to pay attention to his driving? 一号司机相当镇静足以安全 驾车是否属实?		

单值参数(属性)

属性	Trans	Prompt	Printid
NO-LANES	(the number of lanes) (车道数目)	T	

询问:	What is the number of lanes?
	车道数目是什么?
DRIVER-NAME (the name of *)	T DRIVER-
	(* 的名字)
询问:	What is the name of DRIVER-1?
	一号司机的名字是什么?

A.4.2 PRINTID, SYN, UNIQUE 特性的使用

关于上下文

上下文	PRINTID	SYN	UNIQUE
DIV	DIV-		
DEPARTMENT	DEPT-	((DEPT-NAME, *) (the DEPT-NAME; of *))	T
SECTION	SECT-	((SECT-NAME, *) (the SECT-NAME; of the *))	

SECTION的PROMPT1ST特性:

In what section of * are you in?

你处在*的什么段中?

询问(没有UNIQUE特性):

In what section of the Production Department of
DIV-1 are you in?

你在DIV-1的生产部的什么工段中?

询问(有UNIQUE特性):

In what section of the Production Department are you in?

你在生产部的什么工段中?

上下文中的属性

上下文	PRINTID	SYN	UNIQUE
DIV	DIV-		
DEPARTMENT	DEPT-	((DEPT-NAME, *) (the DEPT-NAME, of *))	T
SECTION	SECT-	((SECT-NAME, *) (the SECT-NAME, of the *))	

属性

TRANS

PROMPT

BUDGET	(the budget of *) (* 的预算)	T
--------	--------------------------------	---

询问(没有UNIQUE特性): What
is the budget of the Production Sec-
tion of the Electronics Department
of DIV-1?

DIV-1的电子部的生产段的预算是什么?

询问(有UNIQUE特性): What is
the budget of the Production Section
of the Electronic Department?

电子部的生产段的预算是什么?

A.5 从ARL到自然语言的翻译

A.5.1 ARL规则

PREMIS: (\neg SOBER OR \neg DRUG-FREE)

前提: (\neg “清醒的” 或 \neg “未服兴奋剂”)

ACTION: (JUDGEMENT=POOR AND SPRINTT,
“Drinking alcohol or using abusive before
driving are the worst a person can do.”)

动作: (“判断能力” = “很差” 且 SPRINTT: “在开
车之前酗酒或服用兴奋剂是最不可取的。”)

A.5.2 属性

属性 TRANS

SOBER (* was free of the influence of alcohol)

清醒的 (* 未受酒精的影响)

DRUG-FREE (* was free of abusive drugs)

未服兴奋剂 (* 未服用兴奋剂)

JUDGEMENT (the judgement of *)

判断能力 (* 的判断能力)

A.5.3 用自然语言表述的规则

If 1) The driver was not free of the influence of
alcohol, or
2) The driver was not free of abusive drugs

Then 1) It is definite (100%) that the judgement of
DRIVER-1 is poor, and

2) Inform the user of this decision

如果 1) 司机不是头脑清醒的，或

2) 司机不是未服兴奋剂的

则 1) 可确定(100%)一号司机的判断能力很差，且

2) 将这一判断通知用户

A.5.4 在多值属性中使用PROMPT特性

多值属性和多值询问(ASK-ALL)特性

属性	PROMPT	EXPECT
DAMAGE (What was damaged?)	(CAR OTHER-CAR DRIVER OTHER-DRIVER PASSENGERS OTHER-CAR~PASSENGERS PEDESTRIAN)	

QUERY: What was damaged?

.....
Select all applicable responses
.....

CAR

OTHER-CAR

DRIVER

OTHER-DRIVER

PASSENGERS

OTHER-CAR-PASSENGERS

PEDESTRIAN

询问： 什么受到了损害？

.....
请选择所有可能的答案
.....

汽车（自己的）

别人的汽车

司机

别的司机

乘客（自己车上的）

别人汽车上的乘客

行人

没有多值询问特性的多值属性

属性

PROMPT

EXPECT

DAMAGE (Did damage occur to VALUE?)

(CAR OTHER-CAR DRIVER
OTHER-DRIVER PASSENGERS
OTHER-CAR-PASSENGERS PEDESTRIAN)

如果部分产生提示： (DAMAGE=DRIVER)

QUERY: Did damage occur to DRIVER?

.....
Select one of the following
.....

YES

NO

如果部分产生提示：（DAMAGE=PEDESTRIAN）

QUERY: Did damage occur to PEDESTRIAN?

.....
Select one of the following
.....

YES

NO

如果部分产生提示：（损害=司机）

询问： 司机是否受到损害？

.....
请从下述项目中选择其一
.....

是

非

如果部分产生提示：（损害=行人）

询问： 行人是否受到损害？

.....
请从下述项目中选择其一
.....

是

非

附录B “电脑顾问”的命令——LISP形式

B.1 系统定义的元素

B.1.1 算术运算符

下面各运算符的优先级逐行递减，利用括号可以改变运算次序。

LISP	ARL	意 义
MINUS	-	变号
EXP		e (自然对数的底) 的幕
TIMES	*	乘
FQUOTIENT	/	除
PLUS	+	加
DIFFERENCE	-	减
FIX \$		返回不超过给定值的最大整数

B.1.2 谓词

数值谓词 从数量上将一个属性与其取值或与别的属性的取值联系起来。（请见实例）

谓 词	ARL 谓词
LESSP*	LT或<
GREATERP*	GT或>

GREATER*	GE或>=
LESSER*	LE或<=
BETWEEN*	LT和GE联合使用

值谓词 检查如果部分中属性取值的确定性程度。

谓 词	ARL谓词	CF范围
DEFIS	IS DEFIS	1000
SAME	=	201~1000
NOTDEFIS	IS NOTDEFIS	201~999
MIGHTBE	IS MIGHTBE	- 200~1000
DEFNOT	IS DEFNOT	- 1000
NOTSAME	!=	- 1000~200
NOTDEFNOT	IS NOTDEFNOT	- 999~-201
THOUGHTNOT	IS THOUGHTNOT	- 1000~-201
VNOTKNOWN	IS VNOTKNOWN	- 200~200

参数（属性）谓词 检查如果部分中一给定属性的确定性程度，而不管其取值情况。

谓 词	ARL谓词	CF范围
DEFINITE	IS DEFINITE	1000
KNOWN	IS KNOWN	201~1000
	对于是非型属性的非	- 1000~-200
NOTDEFINITE	IS NOTDEFINITE	- 999~999
NOTKNOWN	IS NOTKNOWN	- 1000~200
	对于是非型属性	- 200~200

属性追踪谓词 对属性进行追踪，在ARL中不能用。（见实例一节）

谓 词 意 义

ONCEKNOWN 追踪一属性，返回被追踪属性的CF.

B.1.3 函数

与ARL的作用相同的 也是联系各子句，并在则部分中引入正文信息。

函 数	ARL函数	
\$ AND	AND	只在如果部分中使用
\$ OR	OR	只在如果部分中使用
DO-ALL	AND	只在则部分中使用
SPRINTT	SPRINTT	只在则部分中使用
CONCLUDE	=、!=、!	得出属性的值

仅在LISP中可用 （在“增强型电脑顾问”中，它们在ARL中也同样可用）

函 数 意 义

MPRINTT	打印任意数目的变元
CONCLUDE*	得出属性的几个带CF的值
CONCLUDET	根据测试得出属性的几个可能值
ONCEKOWN*	追踪几个属性——返回每个被追踪属性的CF值
PRINTCONCLUSIONS	打印属性的值；可以包括表头信

	息(见后面的实例)
TEXT	说明文本(见后面的实例)
CONCLUDETEXT	给属性赋文本值
MEASURE1	确定属性的最高CF
VAL	确定属性的几个带CF的值
VAL1	确定具有最高CF的属性值
DONTASK	追踪属性但不向用户询问属性的值
NOTRELEVANT	属性与本上下文无关;不必向用户询问该属性的值或追踪该属性

B.1.4 可信因子CF

与ARL中的完全相同。

B.1.5 变量

也与ARL中的完全相同。

B.2 开发者定义的元素

B.2.1 上下文

与ARL中的完全相同。

B.2.2 参数(属性)

与ARL中的完全相同。

B.2.3 规则

除格式不同以外,与ARL中的完全一样。图B-1说明了它们之间的主要差别。

B.3 命令

除了下述LISP编辑命令以外,都与ARL中的相同:

命令 意义

PP	美观打印；或以易读的形式显示子句、前提或则部分
P	显示当前表达式
IA	将后续数据插入给定表的给定项目之后
IB	将后续数据插入给定表的给定项目之前
DEL	删除指定的项目
REP	用后续数据替换给定表中的给定项目
REP*	替换当前表达式
X	退出LISP的编辑器，返回到“电脑顾问”
CW	清屏
O	跳出一个表达式层
UL	清除表，即取消一对括号
SL	建立子表，即增加一对括号

B.4 实例

B.4.1 算术表达式

LISP, (PLUS(VAL1 CNTXT A) (VAL1 CNTXT B))

ARL, (A + B)

LISP, (TIMS(VAL1 CNTXT A) (VAL1 CNTXT B))

ARL, (A * B)

LISP, (FQUOTIENT (VAL1 CNTXT CAR-SPEED)10)

ARL, (CAR-SPEED/10)

B.4.2 数值谓词

LISP: (GREATERP* (VAL1 CNTXT A) (VAL1
CNTXT B))

ARL: (A>B)

LISP: (LESSEQ* (VAL1 CNTXT A) 10)

ARL: (A<=10)

LISP: (BETWEEN* (VAL1 CNTXT A) (VAL1 CNTXT
B) (VAL1 CNTXT C))

ARL: (A=> B AND A < C)

B.4.3 前提(如果部分)子句

LISP: (SAME CNTXT 车速合理)

ARL: (车速合理)

LISP: (NOTSAME CNTXT 驾车动作正确)

ARL: (! 驾车动作正确)

LISP: (SAME CNTXT 判断能力 很好)

ARL: (判断能力 = 很好)

LISP: (MIGHTBE CNTXT 清醒的)

ARL: (清醒的 IS MIGHTBE)

LISP: (ONCEKNOWN* CNTXT '(机械安全 道路安
全))

ARL: 没有相应形式，它的意思是已收集了有关“机械安

	函数	属性	谓词	值	CP
前 提 (如果部分)	或 或	(判断能力 车速合理 驾车动作正确)	=	很差	
动 作 (则部分)	(有责任			(1000))

A. ARL

	函 数	谓 词		属 性	值	CF
前 提 (如果部分)	(\$ AND OR	(SAME (THOUGHT -NOT (THOUGHT -NOT	CNTXT ONTXT CNTXT	判断能 力 车速合 理) 驾车动 作正 确)	很差)	
动 作 (则部分)	(DO-ALL (CONCLU- DE		CNTXT	有责任	是	TALLY 1000))

B. LISP

图 B-1 规则的ARL格式和LISP格式

全”和“道路安全”的数据。

B.4.4 前提(如果部分)

LISP: (\$ AND (SAME CNTXT 判断能力 一般))

ARL: (判断能力=一般)

LISP: (\$ AND

(SAME CNTXT 判断能力 很好)

(SAME CNTXT 车速合理)

(SAME CNTXT 驾车动作正确))

ARL: (判断能力=很好 且 车速合理 且 驾车 动作 正确)

LISP: (\$ AND

(\$ OR (SAME CNTXT 判断能力 很差)

(NOTSAME CNTXT 车速合理)

(NOTSAME CNTXT 驾车动作正确)))

ARL: (判断能力=很差 或 !车速合理 或 !驾车 动作 正确)

B.4.5 动作(则部分)子句

LISP: (CONCLUDE CNTXT 有责任 TALLY 800)

ARL: (有责任 (800))

LISP: (CONCLUDE* CNTXT 损害 TALLY
((司机 800) (别的司机 600)))

ARL: 没有对应的形式

LISP, (CONCLUDET CNTXT (VAL1 CNTXT 损害的代价))

'((LT 100 600 700)
(GE 100 800 900))

TALLY 损害' (司机 别的司机))

ARL, 没有相应的形式。上述LISP语句的意思是：如果“损害的代价”小于100元，则(这个)“司机”受到“损害”的确定性就是60%，而“别的司机”受到“损害”的确定性则为70%；如果“损害的代价”大于等于100元，则(这个)“司机”受到“损害”的确定性为80%，“别的司机”是90%。

LISP, (CONCLUDETEXT CNTXT 车速合理)

(TEXT NIL"在这个地区行驶的合理车速不能超过(VAL1 CNTXT
车速限制)) TALLY 1000)

ARL, 没有对应的形式。但将告诉用户下述信息：在这个地区行驶的合理车速不能超过每小时30公里(这是属性“车速限制”的当前值)。

LISP, (MPRINTT"在这个地区行驶的合理车速不能超过

'(TEXT TAG1))

属性：TAG1

类型：TEXTAG

翻译：“(每小时30公里)”

ARL, 没有对应的内容。但将告诉用户下述信息：在这个地区行驶的合理车速不能超过每小时30公里。

B.4.6 动作(则部分)

LISP: (DO-ALL

(CONCLUDE CNTXT 清醒的 非 TALLY
1000)

(SPRINTT “酒后开车是最不可取的事情”))

ARL: (: 清醒的 且 SPRINTT “酒后开车是最不可取
的事情”)

LISP: (DO-ALL

(CONCLUDE CNTXT 车速合理 非 TALLY
1000)

(PRINTCONCLUSIONS CNTXT 车速合理
“在这个地区行驶的合理车速” T)

“车速合理”的当前值是30.

ARL: 没有相应的内容。但将告诉用户下述信息：在这个地
区行驶的合理车速不能超过每小时30公里。

B.5 用户定义的函数

B.5.1 过 程

调用“函数”(FUNCTION)命令之后，按照下述步骤操
作：

1. 函数名称——键入名称并按INS键；

2. 函数定义——输入IQLISP形式的LAMBDA 表达式；

3. 变元模式——输入变元，它们可以是下述符号之一：

CF, CNTXT, IGNORE, PARM, TALLY, TEST, VALU
(PARM的值), VALUES (PARM的VALUE-CF偶表)；

4. 翻译——将本函数翻译成自然语言；
5. 类型——从下述内容选择一个：
PREMISE——用作前提；
ACTION——用作动作；
EXPRESSION——用在前提或动作中的表达式。

B.5.2 实例

计算“纳税后收入” (After-Tax Earning) = $(1 - \text{TAX-RATE}) * \text{EARN} + \text{TAX-RATE} * \text{DEPR}$

1. 函数名称: **A-TAX-EARN**

2. 函数A-TAX-EARN 的定义:

$$\begin{aligned} & (\text{LAMBDA } (\text{TAX-RATE } \text{EARN } \text{DEPR}) \\ & \quad (+ (* (- 1 \text{ TAX-RATE}) \text{EARN})) \\ & \quad (* \text{ TAX-RATE } \text{DEPR}))) \end{aligned}$$

3. A-TAX-EARN 的变元模式,

$$(\text{VALU } \text{VALU } \text{VALU})$$

4. A-TAX-EARN 的翻译:

$$(((1 - (1)) * (2)) + ((1) * (3)))$$

5. A-TAX-EARN 的类型,

EXPRESSION

附录C 交通事故原型专家系统的规则库

下面是交通事故原型规则库中各规则的图形表示。这些规则以卡片的形式给出，每一张卡片上记录一个规则簇。（共16页）

为方便阅读，对每张卡片中的英文缩写名称，我们给出了相应的汉语名称或注释。

CONTEXT: DRIVER

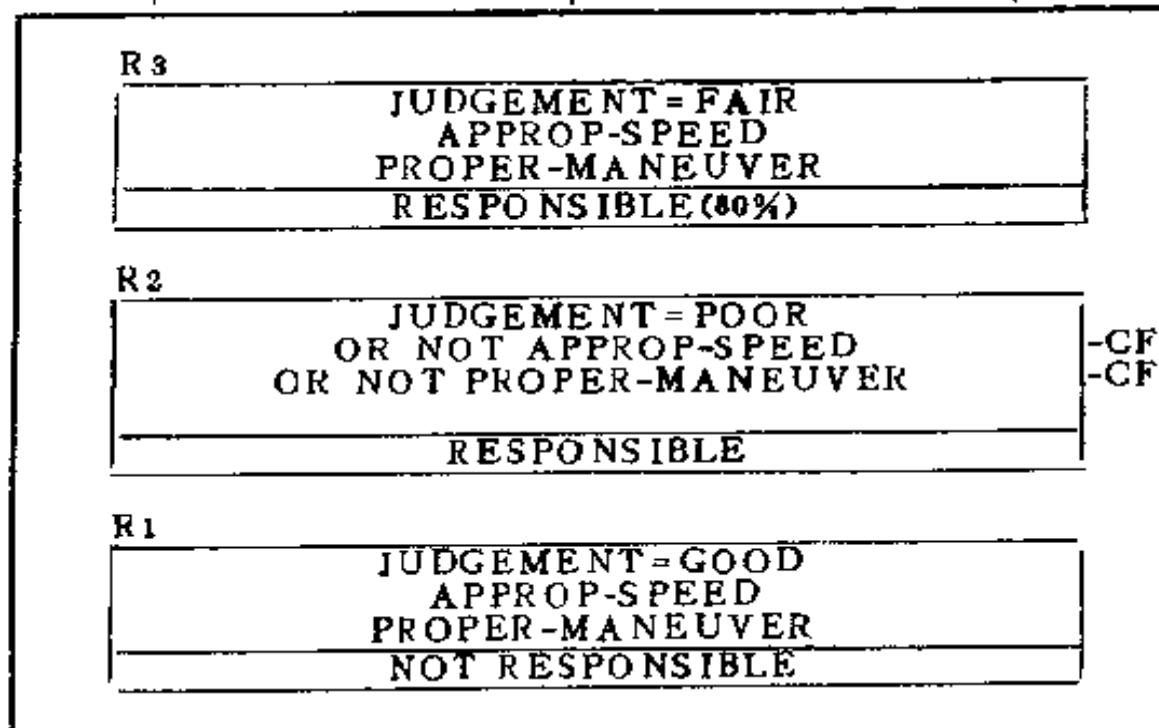
Page 1

RULE STRAND: RESPONSIBLE

JUDGEMENT
(p.2)

APPROP-SPEED
(p.3)

PROPER-MANEUVER
(p.4)



CONTEXT 上下文

RULE STRAND 规则簇

DRIVER 司机

RESPONSIBLE 有责任

JUDGEMENT 判断能力

APPROP-SPEED 车速合理

PROPER-MANEUVER 驾车动作正确

FAIR 一般

POOR 很差

GOOD 很好

OR 或

NOT 非, 不是

CONTEXT: DRIVER
RULE STRAND: RESPONSIBLE

Page 2

R 6	ALERT SOBER DRUG-FREE NOT CALM JUDGEMENT = FAIR
R 7	NOT SOBER OR NOT DRUG-FREE JUDGEMENT = POOR
R 8	NOT ALERT SOBER DRUG-FREE NOT CALM JUDGEMENT = POOR
R 5	ALERT SOBER DRUG-FREE CALM JUDGEMENT = GOOD
R 4	NOT ALERT SOBER DRUG-FREE CALM JUDGEMENT = FAIR

ALERT 有警觉性

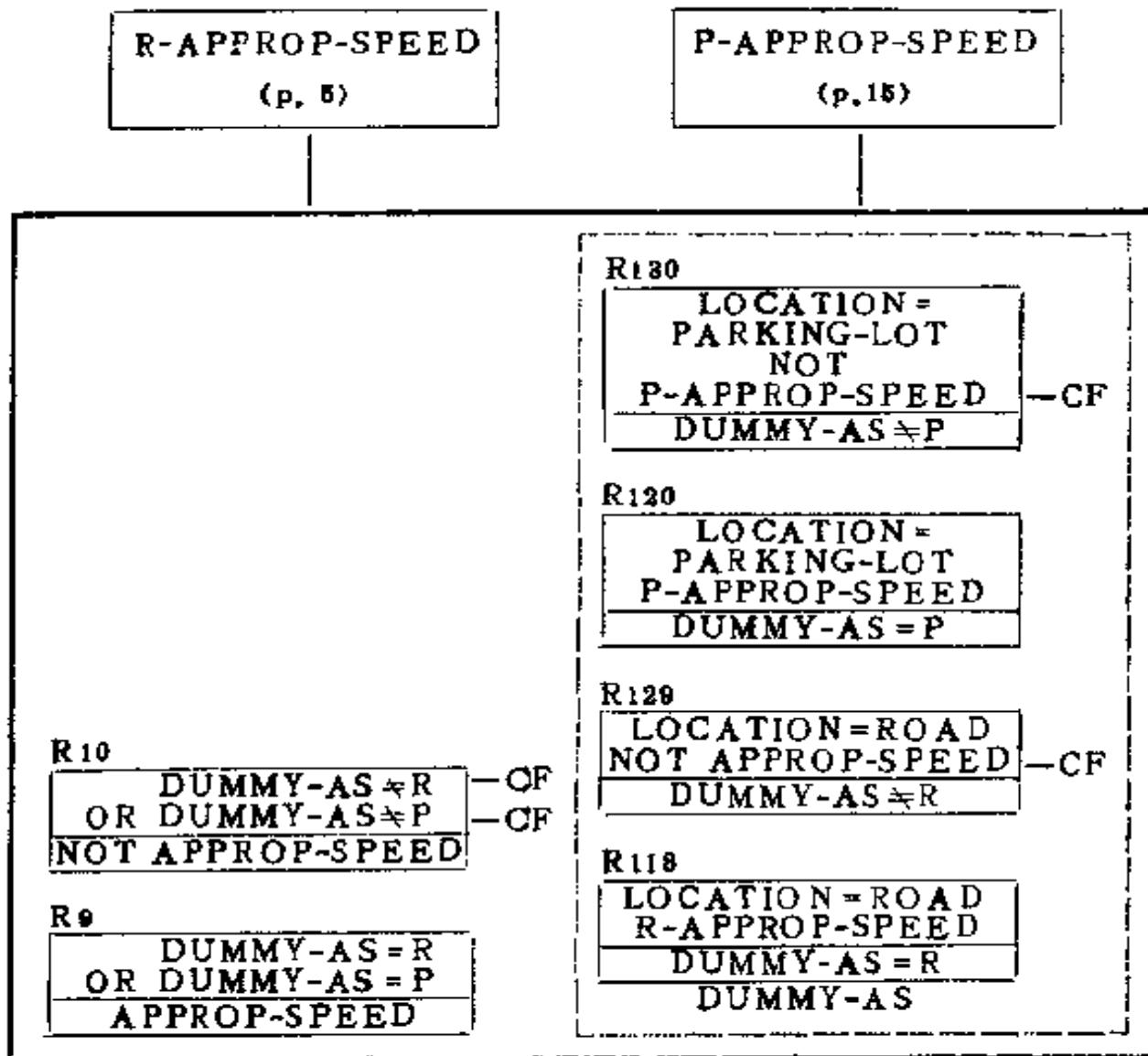
SOBER 清醒的

DRUG-FREE 未服兴奋剂

CALM 镇静的

**CONTEXT, DRIVER
RULE STRAND, APPROP-SPEED**

Page 3



R-APPROP-SPEED 在道路上车速合理

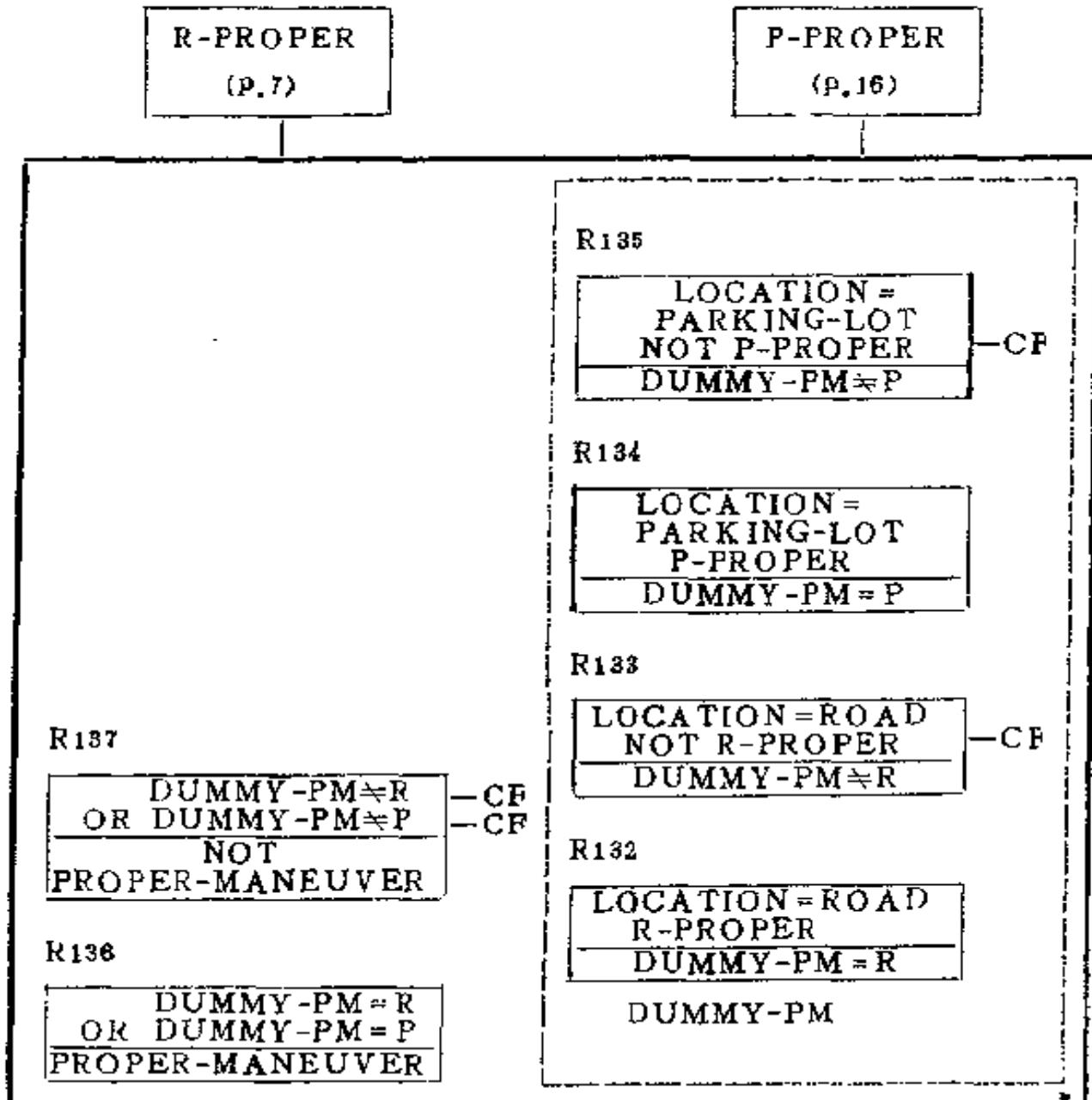
P-APPROP-SPEED 在停车场的车速管理

LOCATION (行车的) 位置

PARKING-LOT 停车场

ROAD 道路

DUMMY-AS 关于车速合理的辅助属性



R-PROPER 在道路上驾车动作正确

P-PROPER 在停车场的驾车动作正确

DUMMYPM 关于驾车动作正确的辅助属性

RULE STRAND: R-APPROP-SPEED

SAFE

(p.6)

R19

EMERG-VEH
CAR-SPEED>0

NOT R-APPROP-SPEED

R18

CAR-SPEED>SPEED-ZONE
NOT R-APPROP-SPEED

R17

SAFE
CAR-SPEED<= SPEED-ZONE
SPEED-VS-LANE = SAMESPEED
NOT EMERG-VEH
R-APPROP-SPEED

R16

SAFE
SPEED-VS-LANE = SLOWER
OR = FASTER
NOT R-APPROP-SPEED(80%)

R15

NOT SAFE
CAR-SPEED<= SPEED-ZONE
NOT R-APPROP-SPEED(80%)

SAFE 安全的

EMERG-VEH 突然上路
CAR-SPEED 实际车速
SPEED-ZONE 车速限制
SPEED-VS-LANE 在车道上的相对车速
SAMESPEED 车速相同
SLOWER 车速较慢
FASTER 车速较快

CONTEXT: ROAD
RULE STRAND: SAFE

Page 6

R23

MECH-SAFE
ROAD-SAFE
SAFE

R22

MECH-SAFE
NOT ROAD-SAFE
NOT SAFE(80%)

R21

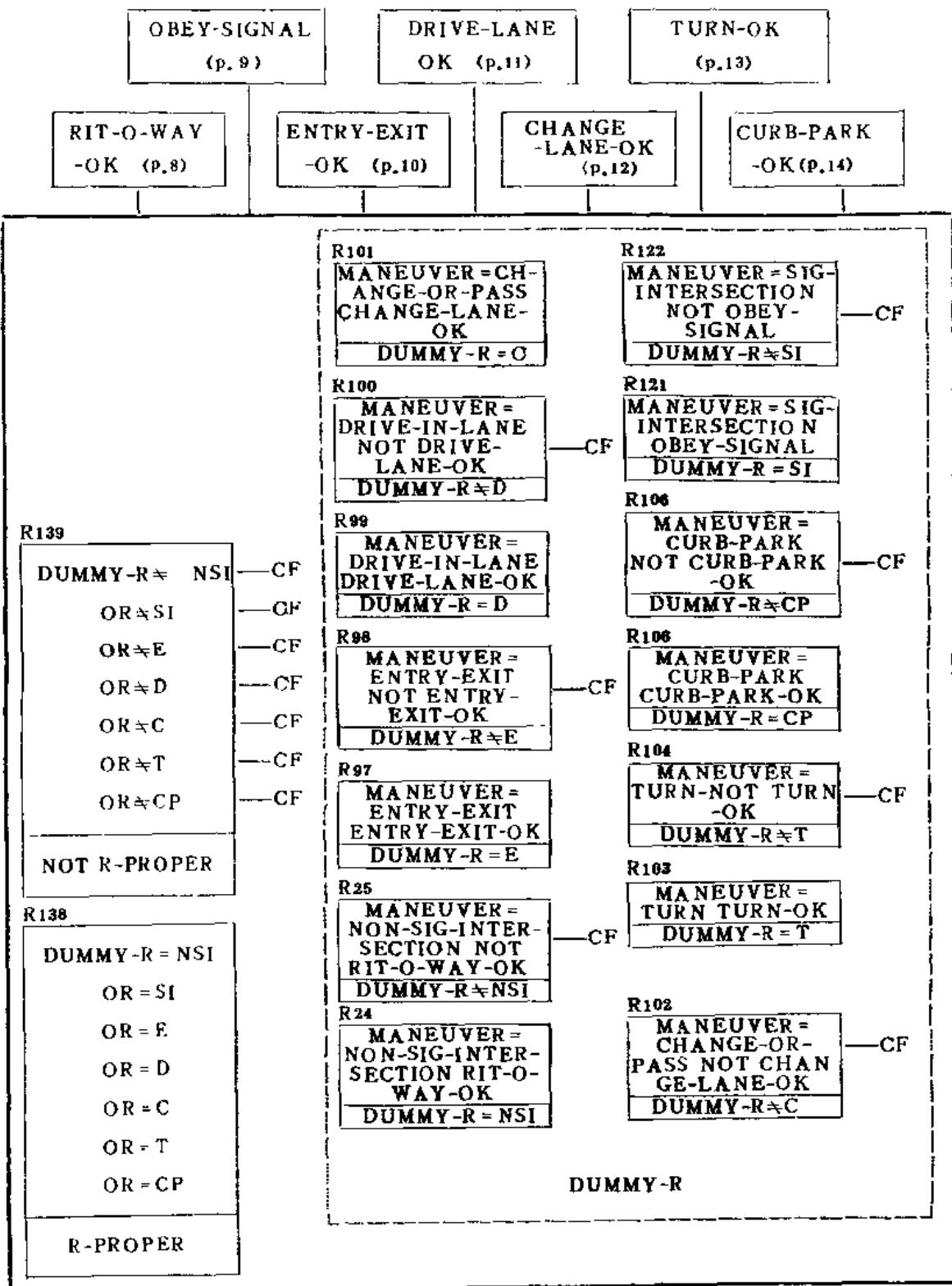
NOT MECH-SAFE
NOT ROAD-SAFE
NOT SAFE

R20

NOT MECH-SAFE
ROAD-SAFE
NOT SAFE(80%)

MECH-SAFE (车辆的) 机械安全

ROAD-SAFE 道路安全





RIT-O-WOK-OK 靠右行驶
OBY-SIGNAL 服从信号
ENTRY-EXIT-OK 进入退出OK
DRIVER-LANE-OK 在车道中行驶OK
CHANG-LANE-OK 变换车道OK
TURN-OK 转弯OK
CURB-PARK-OK 路边停车OK
MANEUVER 驾车动作
CHANGE-OR-PASS 变换车道或超车
DRIVER-IN-LANE 在车道中行驶
ENTRY-EXIT 进入退出
NON-SIG-INTERSECTION 没有信号灯过交叉路口
SIG-INTERSECTION 有信号灯过交叉路口
CURB-PARK 路边停车
TURN 转弯
DUMMY-R 关于在道路上驾车动作正确的辅助属性

CONTEXT, NON-SIG-INTERSECT
RULE STRAND, RIT-O-WAY-OK

Page 8

R 34

NSI-MAN = L-TURN
ENTRY-TIME = SAMETIME
OTHER-CAR-LOC = RIGHT
NOT RIT-O-WAY-OK

R 40

NSI-MAN = U-TURN
ENTRY-TIME = SAMETIME
OTHER-CAR-LOC ≠ RIGHT
NOT OPPOSING-TRAFFIC
RIT-O-WAY-OK

R 33

NSI-MAN = L-TURN
ENTRY-TIME = AFTER
NOT RIT-O-WAY-OK

R 39

NSI-MAN = U-TURN
ENTRY-TIME = SAMETIME
OTHER-CAR-LOC = RIGHT
NOT RIT-O-WAY-OK

R 32

NSI-MAN = L-TURN
ENTRY-TIME = BEFORE
RIT-O-WAY-OK

R 38

NSI-MAN = U-TURN
ENTRY-TIME = BEFORE
NOT OPPOSING-TRAFFIC
RIT-O-WAY-OK

R 31

NSI-MAN = R-TURN
ENTRY-TIME = AFTER
NOT RIT-O-WAY-OK

R 37

NSI-MAN = U-TURN
ENTRY-TIME = AFTER
NOT RIT-O-WAY-OK

R 30

NSI-MAN = R-TURN
ENTRY-TIME ≠ AFTER
RIT-O-WAY-OK

R 36

NSI-MAN = U-TURN
OPPOSING-TRAFFIC
NOT RIT-O-WAY-OK

R 29

YIELD-SIGN
NOT YIELD
NOT RIT-O-WAY-OK

R 35

NSI-MAN = L-TURN
ENTRY-TIME = SAMETIME
OTHER-CAR-LOC ≠ RIGHT
RIT-O-WAY-OK

R 28

YIELD-SIGN
YIELD
RIT-O-WAY-OK

NSI-MAN 没有信号灯过交叉路口时的驾车动作
L-TURN 左转弯
R-TURN 右转弯
U-TURN 掉头
ENTRY-TIME 进入（路口）的时间
SAMETIME 同时
AFTER 以后
BEFORE 以前
OTHER-CAR-LOC 另一汽车的位置
RIGHT 右边
YIELD-SIGN 有让车信号
YIELD 让车
OPPOSING-TRAFFIC 方向相反的车流

CONTEXT: SIG-INTEREST
RULE STRAND: OBEY-SIGNAL

Page 9

R 45

SI-MAN = R-TURN
T-SIGNAL = GREEN

OBEY-SIGNAL

R 44

SI-MAN = R-TURN
T-SIGNAL ≠ GREEN
NOT INTERSECT-TRAFFIC

OBEY-SIGNAL

R 43

ST-MAN = R-TURN
T-SIGNAL ≠ GREEN
INTERSECT-TRAFFIC

NOT OBEY-SIGNAL

R 42

SI-MAN = CROSS
T-SIGNAL ≠ RED

OBEY-SIGNAL

R 41

SI-MAN = CROSS
T-SIGNAL = RED

NOT OBEY-SIGNAL

R 48

SI-MAN = U-TURN
OR = L-TURN
T-SIGNAL ≠ RED
NOT OPPOSING-TRAFFIC

OBEY-SIGNAL

R 47

SI-MAN = U-TURN
OR = L-TURN
T-SIGNAL ≠ RED
OPPOSING-TRAFFIC

NOT OBEY-SIGNAL

R 46

SI-MAN = U-TURN
OR = L-TURN
T-SIGNAL = RED

NOT OBEY-SIGNAL

SI-MAN 有信号灯过交叉路口时的驾车动作

CROSS 横穿（交叉路口）

INTERSECT-TRAFFIC 交叉路口的交通拥挤

T-SIGNAL 交通信号

GREEN 绿色（信号灯）

RED 红色（信号灯）

CONTEXT: ENTRY-EXIT
RULE STRAND: ENTRY-EXIT-OK

Page 10

R 54

E-MAN=GET-OFF
NOT CORRECT-SIGNAL

NOT ENTRY-EXIT-OK

R 53

E-MAN=GET-OFF
CORRECT-SIGNAL
NOT OBEY-SIGN

NOT ENTRY-EXIT-OK

R 52

E-MAN=GET-OFF
CORRECT-SIGNAL
OBEY-SIGN

ENTRY-EXIT-OK

R 51

E-MAN=GET-ON
NOT CORRECT-SIGNAL
NOT ENTRY-EXIT-OK

R 50

E-MAN=GET-ON
CORRECT-SIGNAL
NOT LANE-OPENING

NOT ENTRY-EXIT-OK

R 49

E-MAN=GET-ON
CORRECT-SIGNAL
LANE-OPENING

ENTRY-EXIT-OK

E-MAN 进入退出的驾车动作
GET-OFF 退出（道路）
GET-ON 上路
CORRECT-SIGNAL 信号正确
OBEY-SIGN 服从交通标志
LANE-OPENING 车道开放

R143

TOW
NOT 2-RIGHT-LANES
NOT DRIVE-LANE-OK

R145

TOW
2-RIGHT-LANES
NOT OTHER-CAR-TOWING
CAR-LENGTHS>CAR-SPEED/10
DRIVE-LANE-OK

R144

TOW
2-RIGHT-LANES
OTHER-CAR-TOWING
NOT OVER-300-FT
NOT DRIVE-LANE-OK

R58

TOW
2-RIGHT-LANES
OTHER-CAR-TOWING
OVER-300-FT
DRIVE-LANE-OK

R57

NOT TOW
CAR-LENGTHS>CAR-SPEED/10
DRIVE-LANE-OK

R146

CAR-LENGTHS<=CAR-SPEED/10
NOT DRIVE-LANE-OK

TOW 拖挂（车斗或拖车）

2-RIGHT-LANES 双右车道；司机在其中一条车道中行驶

OTHER-CAR-TOWING 前面的车是拖车

CAR-LENGTHS 车身距离；与前面的车辆相距的车身数

OVER-300-FT 车距（与前面的车辆之间的距离）大于三

百英尺

RULE STRAND: CHANGE-LANE-OK

R 60

C-MAN = WEAVE
NOT CHANGE-LANE-OK

R 64

C-MAN = CHANGE
 CORRECT-SIGNAL
 NOT LANE-OPENING
 NOT CHECK-BLIND-SPOT
NOT CHANGE-LANE-OK

R 63

C-MAN = CHANGE
 CORRECT-SIGNAL
 NOT LANE-OPENING
 CHECK-BLIND-SPOT
NOT CHANGE-LANE-OK

R 62

C-MAN = CHANGE
 CORRECT-SIGNAL
 LANE-OPENING
 NOT CHECK-BLIND-SPOT
CHANGE-LANE-OK (90%)

R 61

C-MAN = CHANGE
 CORRECT-SIGNAL
 LANE-OPENING
 CHECK-BLIND-SPOT
CHANGE-LANE-OK

R 65

NOT CORRECT-SIGNAL
NOT CHANGE-LANE-OK

R 70

C-MAN = PASS
 SPEED-VS-RLANE
 ≠ FASTER
NOT CHANGE-LANE-OK
 (80%)

R 69

C-MAN = PASS
 SPEED-VS-RLANE
 = FASTER
 NOT OBEY-SIGN
NOT CHANGE-LANE-OK

R 68

C-MAN = PASS
 SPEED-VS-RLANE
 = FASTER
 OBEY-SIGN
 VISIBILITY ≠ GOOD
NOT CHANGE-LANE-OK

R 67

C-MAN = PASS
 SPEED-VS-RLANE
 = FASTER
 OBEY-SIGN
 VISIBILITY = GOOD
CHANGE-LANE-OK

C-MAN 变换车道的驾车动作

CHANG 变换车道

WEAVE 迂回行进

PASS 超车

CHECK-BLIND-SPOT 观察身后的车流情况

VISIBILITY 可见程度

CONTEXT: TURNING
RULE STRAND: TURN-OK

Page 13

R 76

T-MAN = ENTER-TRAFFIC
NOT LANE-OPENING

NOT TURN-OK

R 75

T-MAN = ENTER-TRAFFIC
LANE-OPENING
CORRECT-SIGNAL
OBEY-SIGN
PROPER-LANE

TURN-OK

R 74

T-MAN = L-TURN
OR = U-TURN
NOT OPPOSING-TRAFFIC
CORRECT-SIGNAL
OBEY-SIGN
PROPER-LANE

TURN-OK

R 73

T-MAN = L-TURN
OR = U-TURN
OPPOSING-TRAFFIC

NOT TURN-OK

R 72

NOT CORRECT-SIGNAL

NOT TURN-OK

R 71

NOT OBEY-SIGN

NOT TURN-OK

T-MAN 转弯的驾车动作
ENTER-TRAFFIC 驶入道路
PROPER-LANE 行驶的车道正确

CONTEXT: CURB-PARKING
RULE STRAND: CURB-PARK-OK

Page 14

R82

CP-MAN = ENTER-SPACE
CURB-SPACE
CORRECT-SIGNAL
OBEY-SIGN
CURB-PARK-OK

R81

CP-MAN = ENTER-SPACE
NOT OBEY-SIGN
NOT CURB-PARK-OK

R80

CP-MAN = ENTER-SPACE
NOT CURB-SPACE
NOT CURB-PARK-OK

R79

CP-MAN = ENTER TRAFFIC
LANE-OPENING
CORRECT-SIGNAL
CURB-PARK-OK

R78

CP-MAN = ENTER-TRAFFIC
NOT LANE-OPENING
NOT CURB-PARK-OK

R77

NOT CORRECT-SIGNAL
NOT CURB-PARK-OK

CP-MAN 路边停车的驾车动作

ENTER-SPACE 驶入空地

CURB-SPACE 有供停车的空地

R85

PCAR-SPEED <= 10
NOT PSAFE
NOT P-APPROP-SPEED

R84

PCAR-SPEED <= 10
PSAFE
P-APPROP-SPEED

R83

PCAR-SPEED > 10
NOT P-APPROP-SPEED

PCAR-SPEED 停车时的实际车速

PSAFE 停车安全

R 88

P-MAN = PARK
OR = UNPARK
DIRECTION-OK
WITHIN-SPACE
ROOM

P-PROPER

R 87

P-MAN = ENTRY
OR = EXIT
OR = TURN
OR = IN-AISLE
OR = PARK
OR = UNPARK
NOT ROOM

NOT P-PROPER

R 92

P-MAN = PARKED
WITHIN-SPACE

P-PROPER

R 90

P-MAN = PARK
OR = UNPARK
OR = PARKED
NOT WITHIN-SPACE

NOT P-PROPER

R 88

P-MAN = ENTRY
OR = EXIT
OR = TURN
OR = IN-AISLE
DIRECTION-OK
ROOM

P-PROPER

P-MAN = ENTRY

OR = EXIT
OR = TURN
OR = IN-AISLE
OR = PARK
OR = UNPARK
NOT DIRECTION-OK

NOT P-PROPER

P-MAN 在停车场的驾车动作

PARK 停车

UNPARK 启动

PARKED 已停车

IN-AISLE 在停车场的通道中行驶

DIRECTION-OK 方向正确

WITHIN-SPACE 在停放车辆的界线之内

ROOM 有允许车辆通过的空地

附录D 交通事故原型专家系统的规则和参数(属性)清单

注意：标有记号¹的名字是上下文的名称，用它可以替换相关属性的“翻译”(TRANS)特性中的*。为了增强可读性，可将上下文名称用司机的姓名(DRIVER-NAME)代替。实际上，这是系统执行时，规则和参数(属性)实际可读的形式。

Rule Group CHANGE-PASSRULES

RULE061 【CHANGE-PASSRULES】

.....

- If 1) the lane change maneuver change-pass¹ did execute is CHANGE, and
2) change-pass¹ signaled his move properly, and
3) there was a proper traffic opening in the lane, and
4) change-pass¹ turned around to check traffic behind him,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly.

RULE062 【CHANGE-PASSRULES】

.....

If 1) the lane change maneuver change-pass¹ did execute is CHANGE, and
2) change-pass¹ signaled his move properly, and
3) there was a proper traffic opening in the lane, and
4) change-pass¹ turned around to check traffic behind him is not true,

Then there is strongly suggestive evidence (90%) that change-pass¹ did change lanes or pass properly.

RULE063 [CHANGE-PASSRULES]

.....

If 1) the lane change maneuver change-pass¹ did execute is CHANGE, and
2) change-pass¹ signaled his move properly, and
3) there was not a proper traffic opening in the lane, and
4) change-pass¹ turned around to check traffic behind him,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly is not true.

RULE064 [CHANGE-PASSRULES]

.....

If 1) the lane change maneuver change-pass¹ did

- execute is CHANGE, and
- 2) change-pass¹ signaled his move properly, and
 - 3) there was not a proper traffic opening in the lane, and
 - 4) change-pass¹ turned around to check traffic behind him is not true,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly is not true.

RULE065 [CHANGE-PASSRULES]

.....

If change-pass¹ signaled his move properly is not true,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly is not true.

RULE066 [CHANGE-PASSRULES]

.....

If the lane change maneuver change-pass¹ did execute is WEAVE,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly is not true.

RULE067 [CHANGE-PASSRULES]

.....

If 1) the lane change maneuver change-pass¹ did

- execute is PASS, and
- 2) the speed of change-pass¹ relative to the lane he was passing is FASTER, and
 - 3) change-pass¹ obeyed the sign, and
 - 4) the view change-pass¹ had of traffic and the road is GOOD,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly.

RULE068 [CHANGE-PASSRULES]

.....

- If 1) the lane change maneuver change-pass¹ did execute is PASS, and
- 2) the speed of change-pass¹ relative to the lane he was passing is FASTER, and
 - 3) change-pass¹ obeyed the sign, and
 - 4) the view change-pass¹ had of traffic and the road is not GOOD,

Then it is definite (100%) that change-pass¹ did change lanes or pass properly is not true.

RULE069 [CHANGE-PASSRULES]

.....

- If 1) the lane change maneuver change-pass¹ did execute is PASS, and
- 2) the speed of change-pass¹ relative to the lane he was passing is FASTER, and

3) change-pass¹ obeyed the sign is not true,
Then it is definite (100%) that change-pass¹ did
change lanes or pass properly is not true.

RULE070 [CHANGE-PASSRULES]

.....

If 1) the lane change maneuver change-pass¹ did execute is PASS, and
2) the speed of change-pass¹ relative to the lane he was passing is not FASTER,
Then there is strongly suggestive evidence (80%) that change-pass¹ did change lanes or pass properly is not true.

Rule Group CURB-PARKINGRULES

RULE077 [CURB-PARKINGRULES]

.....

If curb-parking¹ signaled his move properly is not true,
Then it is definite (100%) that curb-parking¹ parked properly at the curb is not true.

RULE078 [CURB-PARKINGRULES]

.....

If 1) the curb-parking maneuver curb-parking¹ did execute is ENTER-TRAFFIC, and

- 2) there was not a proper traffic opening in the lane,

Then it is definite (100%) that curb-parking¹ parked properly at the curb is not true.

RULE079 [CURB-PARKINGRULES]

.....

- If
- 1) the curb-parking maneuver curb-parking¹ did execute is ENTER-TRAFFIC, and
 - 2) there was proper traffic opening in the lane, and
 - 3) curb-parking¹ signaled his move properly,

Then it is definite (100%) that curb-parking¹ parked properly at the curb.

RULE080 [CURB-PARKINGRULES]

.....

- If
- 1) the curb-parking maneuver curb-parking¹ did execute is ENTER-SPACE, and
 - 2) there was not enough space at the curb for the car of curb-parking, 1

Then it is definite (100%) that curb-parking¹ parked properly at the curb is not true.

RULE081 [CURB-PARKINGRULES]

.....

- If
- 1) the curb-parking maneuver curb-parking¹ did

execute is ENTER-SPACE, and
2) curb-parking¹ obeyed the sign is not true,
Then it is definite (100%) that curb-parking¹ parked
properly at the curb is not true.

RULE082 [CURB-PARKINGRULES]

.....

If 1) the curb-parking maneuver curb-parking¹
did execute is ENTER-SPACE, and
2) there was enough space at the curb for the
car of curb-parking¹, and
3) curb-parking¹ signaled his move properly, and
4) curb-parking¹ obeyed the sign,

Then it is definite (100%) that curb-parking¹ parked
properly at the curb.

Rule Group DRIVE-IN-LANERULES

RULE057 [DRIVE-IN-LANERULES]

.....

If 1) drive-in-lane¹ was not towing a vehicle or
other object, and
2) the distance drive-in-lane¹ kept between
himself and the vehicle ahead of him in car
lengths is greater than [the speed drive-in
-lane¹ did go when the accident occurred
divided by 10],

Then it is definite (100%) that drive-in-lane¹ drove in lane correctly.

RULE058 [DRIVE-IN-LANERULES]

.....

If 1) drive-in-lane¹ was towing a vehicle or other object, and
2) drive-in-lane¹ stayed in one of the 2 right lanes, and
3) the car ahead was towing a vehicle or a big truck, and
4) the distance between the driver and the vehicle ahead was over 300 feet,

Then it is definite (100%) that drive-in-lane¹ drove in lane correctly.

RULE143 [DRIVE-IN-LANERULES]

.....

If 1) drive-in-lane¹ was towing a vehicle or other object, and
2) drive-in-lane¹ stayed in one of the 2 right lanes is not true,

Then it is definite (100%) that drive-in-lane¹ drove in lane correctly is not true.

RULE144 [DRIVE-IN-LANERULES]

.....

- If 1) drive-in-lane¹ was towing a vehicle or other object, and
2) drive-in-lane¹ stayed in one of the 2 right lanes, and
3) the car ahead was towing a vehicle or a big truck, and
4) the distance between the driver and the vehicle ahead was not over 300 feet,

Then it is definite (100%) that drive-in-lane¹ drove in lane correctly is not true.

RULE145 [DRIVE-IN-LANERULES]

.....

- If —1) drive-in-lane¹ was towing a vehicle or other object, and
2) drive-in-lane¹ stayed in one of the 2 right lanes, and
3) the car ahead was not towing a vehicle or a big truck, and
4) the distance drive-in-lane¹ kept between himself and the vehicle ahead of him in car lengths is greater than [the speed drive-in-lane¹ did go when the accident occurred divided by 10],

Then it is definite (100%) that drive-in-lane¹ drove in lane correctly.

RULE146 【DRIVE-IN-LANERULES】

.....

If the distance drive-in-lane¹ kept between himself and the vehicle ahead of him in car lengths is less than or equal to [the speed drive-in-lane¹ did go when the accident occurred divided by 10],
Then it is definite (100%) that drive-in-lane¹ drove in lane correctly is not true.

Rule Group DRIVERRULES

RULE001 【DRIVERRULES】

.....

If 1) the judgement driver¹ did exercise is GOOD,
and
2) driver¹ traveled at the appropriate speed,
and
3) driver¹ made a proper maneuver,
Then it is definite (100%) that driver¹ is not responsible for the accident.

RULE002 【DRIVERRULES】

.....

If 1) the judgement driver¹ did exercise is POOR,
or
2) there is evidence driver¹ traveled at the

appropriate speed is not true, or

- 3) there is evidence driver¹ made a proper maneuver is not true,

Then it is definite (100%) that driver¹ is responsible for the accident.

RULE003 [DRIVERRULES]

.....

- If
- 1) the judgement driver¹ did exercise is FAIR,
and
 - 2) driver¹ traveled at the appropriate speed,
and
 - 3) driver¹ made a proper maneuver,

Then there is strongly suggestive evidence (80%) that driver¹ is responsible for the accident.

RULE004 [DRIVERRULES]

.....

- If
- 1) driver¹ was not alert, and
 - 2) driver¹ was sober, and
 - 3) driver¹ was free of abusive drugs, and
 - 4) driver¹ was calm enough to pay attention to driving,

Then it is definite (100%) that the judgement driver¹ did exercise is FAIR.

RULE005 [DRIVERRULES]

.....

If 1) driver¹ was alert, and
2) driver¹ was sober, and
3) driver¹ was free of abusive drugs, and
4) driver¹ was calm enough to pay attention
to driving,

Then it is definite (100%) that the judgement driver¹
did exercise is GOOD.

RULE006 [DRIVERRULES]

.....

If 1) driver¹ was not alert, and
2) driver¹ was sober, and
3) driver¹ was free of abusive drugs, and
4) driver¹ was not calm enough to pay attention
to driving,

Then it is definite (100%) that the judgement
driver¹ did exercise is POOR.

RULE007 [DRIVERRULES]

.....

If 1) driver¹ was not sober, or
2) driver¹ was not free of abusive drugs,

Then it is definite (100%) that the judgement driver¹
did exercise is POOR.

RULE008 [DRIVERRULES]

.....

- If 1) driver¹ was alert, and
2) driver¹ was sober, and
3) driver¹ was free of abusive drugs, and
4) driver¹ was not calm enough to pay attention
to driving,

Then it is definite (100%) that the judgement driver¹
did exercise is FAIR.

RULE009 [DRIVERRULES]

.....

- If 1) A dummy for entering a subordinate context
from APPROP-SPEED is R, or
2) A dummy for entering a subordinate context
from APPROP-SPEED is P,

Then it is definite (100%) that driver¹ traveled at
the appropriate speed.

RULE010 [DRIVERRULES]

.....

- If 1) there is evidence that a dummy for entering
a subordinate context from APPROP-SPEED
is not R, or
2) there is evidence that a dummy for entering
a subordinate context from APPROP-SPEED

is not P,

Then it is definite (100%) that driver¹ traveled at the appropriate speed is not true.

RULE118 [DRIVERRULES]

.....

If 1) the location of the accident is ROAD, and
2) driver¹ followed an appropriate speed for the conditions of the road,

Then it is definite (100%) that a dummy for entering a subordinate context from APPROP-SPEED is R.

RULE120 [DRIVERRULES]

.....

If 1) the location of the accident is PARKING-LOT, and
2) driver¹ did go at the proper speed for a parking-lot,

Then it is definite (100%) that a dummy for entering a subordinate context from APPROP-SPEED is P.

RULE129 [DRIVERRULES]

.....

If 1) the location of the accident is ROAD,
and

2) there is evidence driverⁱ followed an appropriate speed for the conditions of the road is not true,

Then it is definite (100%) that a dummy for entering a subordinate context from APPROP-SPEED is not R.

RULE130 [DRIVERRULES]

.....

If 1) the location of the accident is PARKING-LOT, and

2) there is evidence driverⁱ did go at the proper speed for a parking lot is not true,

Then it is definite (100%) that a dummy for entering a subordinate context from APPROP-SPEED is not P.

RULE132 [DRIVERRULES]

.....

If 1) the location of the accident is ROAD, and

2) driverⁱ did a proper road maneuver,

Then it is definite (100%) that the dummy for entering the subordinate context from PROPER-MANEUVER is R.

RULE133 [DRIVERRULES]

.....

If 1) the location of the accident is ROAD, and
2) there is evidence driver¹ did a proper road
maneuver is not true,

Then it is definite (100%) that the dummy for
entering the subordinate context from PROPER
-MANEUVER is not R.

RULE134 【DRIVERRULES】

.....

If 1) the location of the accident is PARKING
-LOT, and
2) driver¹ did maneuver correctly on the par-
king lot,

Then it is definite (100%) that the dummy for
entering the subordinate context from PROPER
-MANEUVER is P.

RULE135 【DRIVERRULES】

.....

If 1) the location of the accident is PARKING
-LOT, and
2) there is evidence driver¹ did maneuver
correctly on the parking lot is not true.

Then it is definite (100%) that the dummy for
entering the subordinate context from PROPER
-MANEUVER is not P.

RULE136 [DRIVERRULES]

.....

- If 1) the dummy for entering the subordinate context from PROPER-MANEUVER is R, or
- 2) the dummy for entering the subordinate context from PROPER-MANEUVER is P.

Then it is definite (100%) that driverⁱ made a proper maneuver.

RULE137 [DRIVERRULES]

.....

- If 1) there is evidence that the dummy for entering the subordinate context from PROPER-MANEUVER is not R, or
- 2) there is evidence that the dummy for entering the subordinate context from PROPER-MANEUVER is not P,

Then it is definite (100%) that driverⁱ made a proper maneuver is not true.

Rule Group ENTRY-EXTRULES

RULE049 [ENTRY-EXTRULES]

.....

- If 1) the entry-exit maneuver entry-exitⁱ did execute is GET-ON, and

- 2) entry-exit¹ signaled his move properly, and
- 3) there was a proper traffic opening in the lane.

Then it is definite (100%) that entry-exit¹ did enter or exit the road properly.

RULE050 [ENTRY-EXTRULES]

.....

- If
- 1) the entry-exit maneuver entry-exit¹ did execute is GET-ON, and
 - 2) entry-exit¹ signaled his move properly, and
 - 3) there was not a proper traffic opening in the lane,

Then it is definite (100%) that entry-exit¹ did enter or exit the road properly is not true.

RULE051 [ENTRY-EXTRULES]

.....

- If
- 1) the entry-exit maneuver entry-exit¹ did execute is GET-ON, and
 - 2) entry-exit¹ signaled his move properly is not true,

Then it is definite (100%) that entry-exit¹ did enter or exit the road properly is not true.

RULE052 [ENTRY-EXTRULES]

.....

If 1) the entry-exit maneuver entry-exit¹ did execute is GET-OFF, and
2) entry-exit¹ signaled his move properly,
and
3) entry-exit¹ obeyed the sign,

Then it is definite (100%) that entry-exit¹ did enter or exit the road properly.

RULE053 [ENTRY-EXTRULES]

.....

If 1) the entry-exit maneuver entry-exit¹ did execute is GET-OFF, and
2) entry-exit¹ signaled his move properly, and
3) entry-exit¹ obeyed the sign is not true,

Then it is definite (100%) that entry-exit¹ did enter or exit the road properly is not true.

RULE054 [ENTRY-EXTRULES]

.....

If 1) the entry-exit maneuver entry-exit¹ did execute is GET-OFF, and
2) entry-exit¹ signaled his move properly is not true,

Then it is definite (100%) that entry-exit¹ did enter or exit the road properly is not true.

Rule Group NON-SIG-INTERSECTRULES

RULE028 [NON-SIG-INTERSECTRULES]

.....

- If 1) there was a yield sign at the intersection,
and
2) the context¹ did yield to the car in the
cross street,

Then it is definite (100%) that the context¹ did
follow the right of way laws.

RULE029 [NON-SIG-INTERSECTRULES]

.....

- If 1) there was a yield sign at the intersection, and
2) the context¹ did yield to the car in the
cross street is not true,

Then it is definite (100%) that the context¹ did
follow the right of way laws is not true.

RULE030 [NON-SIG-INTERSECTRULES]

.....

- If 1) the maneuver the context¹ made at the
intersection is R-TURN, and
2) the time the context¹ entered the intersec-
tion relative to the other driver is not
AFTER,

Then it is definite (100%) that the context¹ did

follow the right of way laws.

RULE031 {NON-SIG-INTERSECTRULES}

.....

- If 1) the maneuver the context¹ made at the intersection is R-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is AFTER,

Then it is definite (100%) that the context¹ did follow the right of way laws is not true.

RULE032 {NON-SIG-INTERSECTRULES}

.....

- If 1) the maneuver the context¹ made at the intersection is L-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is BEFORE,

Then it is definite (100%) that the context¹ did follow the right of way laws.

RULE033 {NON-SIG-INTERSECTRULES}

.....

- If 1) the maneuver the context¹ made at the intersection is L-TURN, and
2) the time the context¹ entered the intersec-

tion relative to the other driver is AFTER,
Then it is definite (100%) that the context¹ did
follow the right of way laws is not true.

RULE034 [NON-SIG-INTERSECTRULES]

.....

- If 1) the maneuver the context¹ made at the intersection is L-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is SAMETIME, and
3) the location of the other car relative to the context¹ is RIGHT,

Then it is definite (100%) that the context¹ did follow the right of way laws is not true.

RULE035 [NON-SIG-INTERSECTRULES]

.....

- If 1) the maneuver the context¹ made at the intersection is L-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is SAMETIME, and
3) the location of the other car relative to the context¹ is not RIGHT,

Then it is definite (100%) that the context¹ did follow the right of way laws.

RULE036 [NON-SIG-INTERSECTRULES]

.....

- If 1) the maneuver the context¹ made at the intersection is U-TURN, and
2) there was a lot of traffic in the opposite direction,

Then it is definite (100%) that the context¹ did follow the right of way laws is not true.

RULE037 [NON-SIG-INTERSECTRULES]

.....

- If 1) the maneuver the context¹ made at the intersection is U-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is AFTER,

Then it is definite (100%) that the context¹ did follow the right of way laws is not true.

RULE038 [NON-SIG-INTERSECTRULES]

.....

- If 1) the maneuver the context¹ made at the intersection is U-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is BEFORE,
and
3) there was not a lot of traffic in the opposite

direction,

Then it is definite (100%) that the context¹ did follow the right of way laws.

RULE039 [NON-SIG-INTERSECTRULES]

.....

If 1) the maneuver the context¹ made at the intersection is U-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is SAMETIME, and
3) the location of the other car relative to the context¹ is RIGHT,

Then it is definite (100%) that the context¹ did follow the right of way laws is not true.

RULE040 [NON-SIG-INTERSECTRULES]

.....

If 1) the maneuver the context¹ made at the intersection is U-TURN, and
2) the time the context¹ entered the intersection relative to the other driver is SAMETIME, and
3) the location of the other car relative to the context¹ is not RIGHT, and
4) there was not a lot of traffic in the opposite direction,

Then it is definite (100%) that the context¹ did follow the right of way laws.

Rule Group PARKING-LOTRULES

RULE083 [PARKING-LOTRULES]

.....

If the speed parking-lot¹ went on the parking lot is greater than 10,

Then it is definite (100%) that parking-lot¹ did go at the proper speed for a parking lot is not true.

RULE084 [PARKING-LOTRULES]

.....

If 1) the speed parking-lot¹ went on the parking lot is less than or equal to 10, and
2) conditions were safe for the use of the car in the parking lot,

Then it is definite (100%) that parking-lot¹ did go at the proper speed for a parking lot.

RULE085 [PARKING-LOTRULES]

.....

If 1) the speed parking-lot¹ went on the parking lot is less than or equal to 10, and
2) conditions were safe for the use of the car

in the parking lot is not true,

Then it is definite (100%) that parking-lot¹ did go at the proper speed for a parking lot is not true.

RULE086 [PARKING-LOTRULES]

.....

If 1) 1) the maneuver parking-lot¹ did execute in the parking lot is ENTRY, or
2) the maneuver parking-lot¹ did execute in the parking lot is EXIT, or
3) the maneuver parking-lot¹ did execute in the parking lot is TURN, or
4) the maneuver parking-lot¹ did execute in the parking lot is IN AISLE, and
2) the car was going in the allowed direction, and
3) there was room available for driving through,

Then it is definite (100%) that parking-lot¹ did maneuver correctly on the parking lot.

RULE087 [PARKING-LOTRULES]

.....

If 1) 1) the maneuver parking-lot¹ did execute in the parking lot is ENTRY, or
2) the maneuver parking-lot¹ did execute

- in the parking lot is EXIT, or
 - 3) the maneuver parking-lot¹ did execute in the parking lot is TURN, or
 - 4) the maneuver parking-lot¹ did execute in the parking lot is IN AISLE, or
 - 5) the maneuver parking-lot¹ did execute in the parking lot is PARK, or
 - 6) the maneuver parking-lot¹ execute in the parking lot is UNPARK, and
- 2) there was not room available for driving through,

Then it is definite (100%) that parking-lot¹ did maneuver correctly on the parking lot is not true.

RULE088 [PARKING-LOTRULES]

.....

- If
- 1) 1) the maneuver parking-lot¹ did execute in the parking lot is PARK, or
 - 2) the maneuver parking-lot¹ did execute in the parking lot is UNPARK, and
- 2) The car was going in the allowed direction, and
 - 3) parking-lot¹ did stay within the confines of the parking space, and
 - 4) there was room available for driving through,

Then it is definite (100%) that parking-lot¹ did

maneuver correctly on the parking lot.

RULE089 [PARKING-LOTRULES]

.....

If 1) 1) the maneuver parking-lot¹ did execute in the parking lot is ENTRY,
2) the maneuver parking-lot¹ did execute in the parking lot is EXIT, or
3) the maneuver parking-lot¹ did execute in the parking lot is TURN, or
4) the maneuver parking-lot¹ did execute in the parking lot is IN AISLE, or
5) the maneuver parking-lot¹ did execute in the parking lot is PARK, or
6) the maneuver parking-lot¹ did execute in the parking lot is UNPARK, and
2) The car was not going in the allowed direction,

Then it is definite (100%) that parking-lot¹ did maneuver correctly on the parking lot is not true.

RULE090 [PARKING-LOTRULES]

.....

If 1) 1) the maneuver parking-lot¹ did execute in the parking lot is PARK, or
2) the maneuver parking-lot¹ did execute

- in the parking lot is UNPARK, or
- 3) the maneuver parking-lot¹ did execute in the parking lot is PARKED, and
- 2) parking-lot¹ did stay within the confines of the parking space is not true,

Then it is definite (100%) that parking-lot¹ did maneuver correctly on the parking lot is not true.

RULE092 [PARKING-LOTRULES]

.....

- If 1) the maneuver parking-lot¹ did execute in the parking lot is PARKED, and
- 2) parking-lot¹ did stay within the confines of the parking space,

Then it is definite (100%) that parking-lot¹ did maneuver correctly on the parking lot.

Rule Group . ROADRULES

RULE015 [ROADRULES]

.....

- If 1) conditions were safe for driving is not true, and
- 2) the speed road¹ did go when the accident occurred is less than or equal to the speed limit of the zone where the accident occur-

ed,

Then there is strongly suggestive evidence (80%) that road¹ followed an appropriate speed for the conditions of the road is not true.

RULE016 [ROADRULES]

.....

If 1) conditions were safe for driving and,
2) 1) the speed the car did go with respect to traffic in the same lane is SLOWER, or
2) the speed the car did go with respect to traffic in the same lane is FASTER,

Then there is strongly suggestive evidence (80%) that road¹ followed an appropriate speed for the conditions of the road is not true.

RULE017 [ROADRULES]

.....

If 1) conditions were safe for driving, and
2) the speed road¹ did go when the accident occurred is less than or equal to the speed limit of the zone
3) the speed the car did go with respect to traffic in the same lane is SAME SPEED, and
4) an emergency vehicle did come down the road is not true,

Then it is definite (100%) that road¹ followed an appropriate speed for the conditions of the road.

RULE018 [ROADRULES]

.....

If the speed road¹ did go when the accident occurred is greater than the speed limit of the zone where the accident occurred,

Then it is definite (100%) that road¹ followed an appropriate speed for the conditions of the road is not true.

RULE019 [ROADRULES]

.....

If 1) an emergency vehicle did come down the road, and
2) the speed road¹ did go when the accident occurred is greater than 0,

Then it is definite (100%) that road¹ followed an appropriate speed for the conditions of the road is not true.

RULE020 [ROADRULES]

.....

If 1) the car was not safe for driving, and
2) road conditions were safe for driving,

Then there is strongly suggestive evidence (80%) that conditions were safe for driving is not true.

RULE021 [ROADRULES]

.....

If 1) the car was not safe for driving, and
2) road conditions were safe for driving is not true,

Then it is definite (100%) that conditions were safe for driving is not true.

RULE022 [ROADRULES]

.....

If 1) the car was not safe for driving, and
2) road conditions were safe for driving is not true,

Then there is strongly suggestive evidence (80%) that conditions were safe for driving is not true.

RULE023 [ROADRULES]

.....

If 1) the car was safe for driving, and
2) road conditions were safe for driving,

Then is is definite (100%) that conditions were safe for driving.

RULE024 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is
NON-SIGNAL-INTERSECTION, and
2) road¹ did not follow the right of way laws,
Then it is definite (100%) that the dummy that
determines entry into the subordinate context
of R-PROPER is NSI.

RULE025 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is
NON-SIGNAL-INTERSECTION, and
2) there is evidence road¹ did follow the right
of way laws, is not true.
Then it is definite (100%) that the dummy that
determines entry into the subordinate context
of R-PROPER is NSI.

RULE097 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is
ENTRY-EXIT, and
2) road¹ did enter or exit the road properly,
Then it is definite (100%) that the dummy that
determines entry into the subordinate context
of R-PROPER is E.

RULE098 [ROADRULES]

.....

- If 1) the road maneuver road¹ did execute is
ENTRY-EXIT, and
2) there is evidence road¹ did enter or exit
the road properly is not true,

Then it is definite (100%) that the dummy that
determines entry into the subordinate context
of R-PROPER is not E.

RULE099 [ROADRULES]

.....

- If 1) the road maneuver road¹ did execute is
DRIVE-IN-LANE, and
2) road¹ drove in lane correctly,

Then it is definite (100%) that the dummy that
determines entry into the subordinate context
of R-PROPER is D.

RULE100 [ROADRULES]

.....

- If 1) the road maneuver road¹ did execute is
DRIVE-IN-LANE, and
2) there is evidence road¹ drove in lane
correctly is not true.

Then it is definite (100%) that the dummy that

determines entry into the subordinate context of R-PROPER is not D.

RULE101 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is CHANGE-OR-PASS, and
2) road¹ did change lanes or pass properly,
Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is C.

RULE102 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is CHANGE-OR-PASS, and
2) there is evidence road¹ did change lanes or pass properly is not true.
Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is not C.

RULE103 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is TURN, and
2) road¹ made a turn properly,

Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is T.

RULE104 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is TURN, and
2) there is evidence road¹ made a turn properly is not true.

Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is not T.

RULE105 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is CURB-PARK, and
2) road¹ parked properly at the curb,

Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is CP.

RULE106 [ROADRULES]

.....

If 1) the road maneuver road¹ did execute is CURB-PARK, and

- 2) there is evidence road¹ parked properly at the curb is not true,

Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is not CP.

RULE121 [ROADRULES]

.....

- If 1) the road maneuver road¹ did execute is SIGNAL-INTERSECTION, and
2) road¹ obeyed the signal lights,

Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is SI.

RULE122 [ROADRULES]

.....

- If 1) the road maneuver road¹ did execute is SIGNAL-INTERSECTION, and
2) there is evidence road¹ obeyed the signal lights is not true,

Then it is definite (100%) that the dummy that determines entry into the subordinate context of R-PROPER is not SI.

RULE138 [ROADRULES]

.....

- If 1) the dummy that determines entry into the subordinate context of R-PROPER is NSI, or
2) the dummy that determines entry into the subordinate context of R-PROPER is SI, or
3) the dummy that determines entry into the subordinate context of R-PROPER is E, or
4) the dummy that determines entry into the subordinate context of R-PROPER is D, or
5) the dummy that determines entry into the subordinate context of R-PROPER is C, or
6) the dummy that determines entry into the subordinate context of R-PROPER is T, or
7) the dummy that determines entry into the subordinate context of R-PROPER is CP,

Then it is definite (100%) that roadⁱ did a proper road maneuver.

RULE139 [ROADRULES]

.....

- If 1) there is evidence that the dummy that determines entry into the subordinate context of R-PROPER is not NSI, or
2) there is evidence that the dummy that determines entry into the subordinate context of R-PROPER is not SI, or
3) there is evidence that the dummy that determines entry into the subordinate context

- of R-PROPER is not E, or
- 4) there is evidence that the dummy that determines entry into the subordinate context of R-PROPER is not D, or
- 5) there is evidence that the dummy that determines entry into the subordinate context of R-PROPER is not C, or
- 6) there is evidence that the dummy that determines entry into the subordinate context of R-PROPER is not T, or
- 7) there is evidence that the dummy that determines entry into the subordinate context of R-PROPER is not CP,

Then it is definite (100%) that road¹ did a proper road maneuver is not true.

Rule Group SIG-INTERSECTRULES

RULE041 [SIG-INTERSECTRULES]

.....

- If
- 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is CROSS, and
 - 2) the color of the traffic light at the intersection is RED,

Then it is definite (100%) that sig-intersect¹ obeyed the signal lights is not true.

RULE042 [SIG-INTERSECTRULES]

.....

- If 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is CROSS, and
- 2) the color of the traffic light at the intersection is not RED,

Then it is definite(100%) that sig-intersect¹ obeyed the signal lights.

RULE043 [SIG-INTERSECTRULES]

.....

- If 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is RTURN, and
- 2) the color of the traffic light at the intersection is not GREEN, and
- 3) there was traffic moving in the cross street,

Then it is definite(100%) that sig-intersect¹ obeyed the signal lights is not true.

RULE044 [SIG-INTERSECTRULES]

.....

- If 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is RTURN, and
- 2) the color of the traffic light at the intersection is not GREEN, and
- 3) there was not traffic moving in the cross street,

Then it is definite (100%) that sig-intersect¹ obeyed the signal lights.

RULE045 [SIG-INTERSECTRULES]

.....

If 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is RTURN, and
2) the color of the traffic light at the intersection is GREEN,

Then it is definite (100%) that sig-intersect¹ obeyed the signal lights.

RULE046 [SIG-INTERSECTRULES]

.....

If 1) 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is U-TURN, or
2) the maneuver sig-intersect¹ did at the signal-controlled intersection is L-TURN,
and
2) the color of the traffic light at the intersection is RED.

Then it is definite (100%) that sig-intersect¹ obeyed the signal lights is not true.

RULE047 [SIG-INTERSECTRULES]

.....

If 1) 1) the maneuver sig-intersect¹ did at the

- signal-controlled intersection is U-TURN, or
- 2) the maneuver sig-intersect¹ did at the signal-controlled intersection is L-TURN, and
 - 2) the color of the traffic light at the intersection is not RED, and
 - 3) there was a lot of traffic in the opposite direction,

Then it is definite (100%) that sig-intersect¹ obeyed the signal lights is not true.

RULE048 [SIG-INTERSECTRULES]

.....

- If 1) 1) the maneuver sig-intersect¹ did at the signal-controlled intersection is U-TURN, or
- 2) the maneuver sig-intersect¹ did at the signal-controlled intersection is L-TURN, and
 - 2) the color of the traffic light at the intersection is not RED, and
 - 3) there was not a lot of traffic in the opposite direction,

Then it is definite(100%)that sig-intersect¹ obeyed the signal lights.

Rule Group TURNINGRULES

RULE071 [TURNINGRULES]

.....

If turning¹ obeyed the sign is not true,
Then it is definite(100%)that turning¹ made a turn
properly is not true.

RULE072 [TURNINGRULES]

.....

If turning¹ signaled his move properly is not true.
Then it is definite(100%)that turning¹ made a turn
properly is not true.

RULE073 [TURNINGRULES]

.....

If 1) 1) the turning maneuver turning¹ did execute is L-TURN, or
2) the turning maneuver turning¹ did execute is U-TURN, and
2) there was a lot of traffic in the opposite direction,

Then it is definite(100%)that turning¹ made a turn
properly is not true.

RULE074 [TURNINGRULES]

.....

If 1) 1) the turning maneuver turning¹ did execute is L-TURN, or
2) the turning maneuver turning¹ did execute is U-TURN, and
2) there was not a lot of traffic in the opposite direction, and
3) turning¹ signaled his move properly, and
4) turning¹ obeyed the sign, and
5) turning¹ drove in the appropriate lane,

Then it is definite(100%)that turning¹ made a turn properly.

RULE075 [TURNINGRULES]

.....

If 1) the turning maneuver turning¹ did execute is ENTER-TRAFFIC, and
2) there was a proper traffic opening in the lane, and
3) turning¹ signaled his move properly, and
4) turning¹ obeyed the sign, and
5) turning¹ drove in the appropriate lane,

Then it is definite(100%)that turning¹ made a turn properly.

RULE076 [TURNINGRULES]

- If 1) the turning maneuver turning¹ did execute
is ENTER-TRAFFIC, and
2) there was not a proper traffic opening in
the lane,

Then it is definite(100%) that turning¹ made a turn
properly is not true.

Parameter Group CHANGE-PASS-PARMS

C-MAN [CHANGE-PASS-PARMS]

.....

TRANS: (the lane change maneuver*did execute)

PROMPT: T

ASKFIRST: T

EXPECT: (CHANGE PASS WEAVE)

USED-BY: (RULE061 RULE062 RULE063 RULE064
RULE066 RULE067 RULE068 RULE069
RULE070)

Parameter Group CONTEXTTYPES

CHANGE-PASS [CONTEXTTYPES]

.....

TRANS: (*was changing lanes or passing)

SYN: ((DRIVER-NAME; *) (DRIVER-NAME; *)))

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (CHANGE-PASSRULES)

PARMGROUP: CHANGE-PASS-PARMS

PRINTID: CHANGE-PASS-

PROMPT2ND: NIL

ASSOCWITH: (ROAD DRIVER)

CURB-PARKING [CONTEXTTYPES]

.....

TRANS: (*executed a maneuver related to curb-parking)

SYN: (((DRIVER-NAME; *)(DRIVER-NAME; *)))

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (CURB-PARKINGRULES)

PARMGROUP: CURB-PARKING-PARMS

PRINTID: CURB-PARKING-

PROMPT2ND: NIL

ASSOCWITH: (ROAD DRIVER)

DRIVE-IN-LANE [CONTEXTTYPES]

.....

TRANS: (*was driving in his lane)

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (DRIVE-IN-LANERULES)

PARMGROUP: DRIVE-IN-LANE PARMS

PRINTID: DRIVE-IN-LANE-
PROMPT2ND: NIL
ASSOCWITH: (ROAD DRIVER)
SYN: (((DRIVER-NAME, *)(DRIVER-NAME, *)))

DRIVER [CONTEXTTYPES]

.....

TRANS: (the driver of the car that was involved
in the accident)

OFFSPRING: (ROAD PARKING~LOT)

UNIQUE: T

SYN: (((DRIVER-NAME, *)(DRIVER-NAME, *)))

DISPLAYRESULTS: T

GOALS: (RESPONSIBLE JUDGEMENT APPROP
-SPEED PROPER-MANEUVER)

INITIALDATA: (DRIVER-NAME)

RULETYPES: (DRIVERRULES)

PARMGROUP: DRIVER-PARMS

PRINTID: DRIVER-

PROMPTEVER: (The purpose of this expert system
is to determine whether the driver
of a car is responsible for an acci-
dent)

ENTRY-EXIT [CONTEXTTYPES]

.....

TRANS: (*made an entry to or an exit from a

highway)

SYN: (((DRIVER-NAME; *)(DRIVER-NAME; *)))

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (ENTRY-EXTRULES)

PARMGROUP: ENTRY-EXIT-PARMS

PRINTID: ENTRY-EXIT-

PROMPT2ND: NIL

ASSOCWITH: (ROAD DRIVER)

NON-SIG-INTERSECT [CONTEXTTYPES]

.....

TRANS: (the accident occurred at an intersection
uncontrolled by signal lights)

SYN: (((DRIVER-NAME; *)(DRIVER-NAME; *)))

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (NON-SIG-INTERSECTRULES)

PARMGROUP: NON-SIG-INTERSECT-PARMS

PRINTID: NON-SIG-INTERSECT-

PROMPT2ND: NIL

PROMPT1ST: NIL

ASSOCWITH: (ROAD DRIVER)

PARKING-LOT [CONTEXTTYPES]

.....

TRANS: (parking-lot)

INITIALDATA, NIL
RULETYPES, (PARKING-LOTRULES)
PARMGROUP, PARKING-LOT-PARMS
PRINTID, PARKING-LOT-
PROMPT2ND, NIL
ASSOCWITH, (DRIVER)
SYN, (((DRIVER-NAME; *)(DRIVER-NAME; *)))
UNIQUE, T

ROAD [CONTEXTTYPES]

.....

TRANS, (the road)
UNIQUE, T
SYN, (((DRIVER-NAME; *)(DRIVER-NAME; *)))
OFFSPRING, (NON-SIG-INTERSECT SIG-INTERSECT
 ENTRY-EXIT DRIVE-IN-LANE
 CHANGE-PASS TURNING CURB-PARKING
 NON-SIG-INTERSECT SIG-INTERSECT)

INITIALDATA, NIL
RULETYPES, (INTERSECTION ROADRULES)
PARMGROUP, ROAD-PARMS
PRINTID, ROAD-
PROMPT2ND, NIL
ASSOCWITH, (DRIVER)

SIG-INTERSECT [CONTEXTTYPES]

.....

TRANS: (the accident occurred at an intersection
which was controlled by traffic signal li-
ghts)

SYN: (((DRIVER-NAME; *)(DRIVER-NAME; *)))

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (SIG-INTERSECTRULES)

PARMGROUP: SIG-INTERSECT-PARMS

PRINTID: SIG-INTERSECT-

PROMPT2ND: NIL

PROMPT1ST: NIL

ASSOCWITH: (ROAD DRIVER)

TURNING [CONTEXTTYPES]

.....

TRANS: (*was turning)

SYN: (((DRIVER-NAME; *)(DRIVER-NAME; *)))

INITIALDATA: NIL

GOALS: NIL

RULETYPES: (TURNINGRULES)

PARMGROUP: TURNING-PARMS

PRINTID: TURNING-

PROMPT2ND: NIL

ASSOCWITH: (ROAD DRIVER)

Parameter Group
CURB-PARKING-PARMS

CP-MAN [CURB-PARKING-PARMS]

.....

TRANS, (the curb-parking maneuver*did execute)

PROMPT, T

ASKFIRST, T

EXPECT, (ENTER-SPACE ENTER-TRAFFIC)

USED-BY, (RULE078 RULE079 RULE080 RULE081
 RULE082)

CURB-SPACE [CURB-PARKING-PARMS]

.....

TRANS, (there was enough space at the curb for
 the car of*)

PROMPT, T

USED-BY, (RULE080 RULE082)

Parameter Group
DRIVE-IN-LANE-PARMS

2-RIGHT-LANES [DRIVE-IN-LANE-PARMS]

.....

TRANS, (*stayed in one of the 2 right lanes)

PROMPT: T

USED-BY: (RULE058 RULE144 RULE145 RULE143)

OVER-300-FT [DRIVE-IN-LANE-PARMS]

.....

TRANS: (the distance between the driver and the vehicle ahead was over 300 feet)

PROMPT: T

ASKFIRST: T

USED-BY: (RULE058 RULE144)

Parameter Group DRIVER-PARMS

ALERT [DRIVER-PARMS]

.....

TRANS: (*was alert)

PROMPT: T

USED-BY: (RULE004 RULE005 RULE006 RULE008)

APPROP-SPEED [DRIVER-PARMS]

TRANS: (*traveled at the appropriate speed)

UPDATED-IN: NIL

USED-BY: (RULE001 RULE002 RULE003)

UPDATED-BY: (RULE009 RULE010)

CALM [DRIVER-PARMS]

.....

TRANS: (*was calm enough to pay attention to driving)

PROMPT: T

USED-BY: (RULE004 RULE005 RULE006 RULE008)

DRIVER-NAME [DRIVER-PARMS]

.....

TRANS: (the name of the driver that was involved in the accident)

PROMPT: T

ASKFIRST: T

EXPECT: ANY

DRUG-FREE [DRIVER-PARMS]

.....

TRANS: (*was free of abusive drugs)

PROMPT: T

USED-BY: (RULE004 RULE005 RULE006 RULE007 RULE008)

DUMMY-AS [DRIVER-PARMS]

.....

TRANS: (A dummy for entering a subordinate context from APPROP-SPEED)

DICTIONARY: INTERNAL

EXPECT: (RP)

USED-BY: (RULE009 RULE010)

UPDATED-BY: (RULE118 RULE129 RULE120 RULE
130)

DUMMY-PM [DRIVER-PARMS]

.....

TRANS: (the dummy for entering the subordinate
context from PROPER-MANEUVER)

DICTIONARY: INTERNAL

EXPECT: (RP)

USED-BY: (RULE136 RULE137)

UPDATED-BY: (RULE132 RULE133 RULE134 RULE
135)

JUDGEMENT [DRIVER-PARMS]

.....

TRANS: (the judgement*did exercise)

EXPECT: (GOOD FAIR POOR)

USED-BY: (RULE001 RULE002 RULE003)

UPDATED-BY: (RULE004 RULE005 RULE006 RULE
007 RULE008)

LOCATION [DRIVER-PARMS]

.....

TRANS: (the location of the accident)

PROMPT: (Where did the accident occur?)

ASKFIRST: T

EXPECT: (ROAD PARKING-LOT)

**USED-BY, (RULE118 RULE129 RULE120 RULE130
RULE132 RULE133 RULE134 RULE135)**

P-APPROP-SPEED [DRIVER-PARMS]

.....

**TRANS, (*did go at the peoper speed for a parking
lot)**

UPDATED-IN, NIL

USED-BY, (RULE120 RULE130)

UPDATED-BY, (RULE083 RULE084 RULE085)

P-PROPER [DRIVER-PARMS]

.....

**TRANS, (*did maneuver correctly on the parking
lot)**

UPDATED-IN, NIL

USED-BY, (RULE134 RULE135)

**UPDATED-BY, (RULE086 RULE087 RULE088 RULE
089 RULE090 RULE092)**

PROPER-MANEUVER [DRIVER-PARMS]

.....

TRANS, (*made a proper maneuver)*

USED-BY, (RULE001 RULE002 RULE003)

UPDATED-BY, (RULE136 RULE137)

R-APPROP-SPEED [DRIVER-PARMS]

.....
TRANS: (*followed an appropriate speed for the conditions of the road)

USED-BY: (RULE118 RULE129)

UPDATED-BY: (RULE015 RULE016 RULE017 RULE018 RULE019)

R-PROPER [DRIVER-PARMS]

.....
TRANS: (*did a proper road maneuver)

USED-BY: (RULE132 RULE133)

UPDATED-BY: (RULE138 RULE139)

RESPONSIBLE [DRIVER-PARMS]

.....
TRANS: (*is responsible for the accident)

UPDATED-BY: (RULE001 RULE002 RULE003)

SOBER[DRIVER-PARMS]

.....
TRANS: (*was sober)

PROMPT: T

REPROMPT: (The effect drinks have on driving depends upon the number of drinks consumed, the weight of the driver and the time elapsed since drinking. If you have doubt about the driver's sobriety,

look up the BAC(Blood Alcohol Concentration)charts.)

**USED-BY, (RULE004 RULE005 RULE006 RULE007
RULE008)**

**Parameter Group
ENTRY-EXIT-PARMS**

E-MAN [ENTRY-EXIT-PARMS]

.....

TRANS, (the entry-exit maneuver*did execute)

PROMPT, T

ASKFIRST, T

EXPECT, (GET-ON GET-OFF)

**USED-BY, (RULE049 RULE050 RULE051 RULE052
RULE053 RULE054)**

**Parameter Group
NON-SIG-INTERSECT-PARMS**

ENTRY-TIME [NON-SIG-INTERSECT-PARMS]

.....

**TRANS, (the time*entered the intersection relative
to the other driver)**

**PROMPT, (When did*enter the intersection relative
to the other driver?)**

ASKFIRST, T
EXPECT, (BEFORE SAMETIME AFTER)
USED-BY, (RULE030 RULE031 RULE032 RULE033
 RULE034 RULE035 RULE037 RULE038
 RULE039 RULE040)

NSI-MAN [NON-SIG-INTERSECT-PARMS]

.....

TRANS, (the maneuver*made at the intersection)
PROMPT, (What maneuver did*execute at the non
 -signal intersection?)

ASKFIRST, T
EXPECT, (L-TURN R-TURN U-TURN)
USED-BY, (RULE030 RULE031 RULE032 RULE033
 RULE034 RULE035 RULE036 RULE037
 RULE038 RULE039 RULE040)

OTHER-CAR-LOC [NON-SIG-INTERSECT-PARMS]

.....

TRANS, (the location of the other car relative to*)
PROMPT, (What was the location of the other car
 relative to*)?

ASKFIRST, T
EXPECT, (RIGHT LEFT)
USED-BY, (RULE034 RULE035 RULE039 RULE040)

YIELD [NON-SIG-INTERSECT-PARMS]

.....

TRANS: (*did yield to the car in the cross street)

PROMPT: T

ASKFIRST: T

USED-BY: (RULE028 RULE029)

YIELD-SIGN [NON-SIG-INTERSECT-PARMS]

.....

TRANS: (there was a yield sign at the intersection)

PROMPT: T

ASKFIRST: T

USED-BY: (RULE028 RULE029)

Parameter Group

PARKING-LOT-PARMS

DIRECTION-OK [PARKING-LOT-PARMS]

.....

TRANS: (the car was going in the allowed direction)

PROMPT: T

UPDATED-IN: NIL

USED-BY: (RULE086 RULE088 RULE089)

P-MAN [PARKING-LOT-PARMS]

.....

TRANS: (the maneuver*did execute in the parking

lot)

PROMPT, T

ASKFIRST, T

EXPECT, (PARK UNPARK ENTRY EXIT TURN IN
- AISLE PARKED)

USED-BY, (RULE086 RULE087 RULE088 RULE089
RULE090 RULE092)

PCAR-SPEED [PARKING-LOT-PARMS]

.....

TRANS, (the speed*went on the parking lot)

PROMPT, T

UPDATED-IN, NIL

EXPECT, POSNUMB

USED-BY, (RULE083 RULE084 RULE085)

PSAFE [PARKING-LOT-PARMS]

.....

TRANS, (conditions were safe for the use of the
car in the parking lot)

PROMPT, T

UPDATED-IN, NIL

USED-BY, (RULE084 RULE085)

ROOM [PARKING-LOT-PARMS]

.....

TRANS, (there was room available for driving

through)
PROMPT, T
UPDATED-IN, NIL
USED-BY, (RULE086 RULE087 RULE088)

WITHIN-SPACE [PARKING-LOT-PARMS]

.....
TRANS, (*did stay within the confines of the parking space)
PROMPT, T
UPDATED-IN, NIL
USED-BY, (RULE088 RULE090 RULE092)

Parameter Group ROAD-PARMS

CAR-DIST[ROAD-PARMS]

.....
TRANS, (the distance*kept between himself and the car ahead)
PROMPT, T
EXPECT, POSNUMB
USED-BY, NIL

CAR-LENGTHS [ROAD-PARMS]

.....
TRANS, (the distance*kept between himself and the vehicle ahead of him in car lengths)

PROMPT: (What distance in car lengths did*keep between himself and the vehicle ahead of him?)

REPROMPT: (For the sake of safety a driver should make sure that there is sufficient distance between himself and the vehicle ahead of him. Sufficient distance means that the number of car lengths between the vehicles is at least equal to the driver's car speed divided by 10)

EXPECT: POSNUMB

USED-BY: (RULE146 RULE057 RULE145)

CAR-SPEED [ROAD-PARMS]

.....

TRANS: (the speed*did go when the accident occurred)

PROMPT: (How fast was the driver going when the accident occurred?)

EXPECT: POSNUMB

USED-BY: (RULE015 RULE017 RULE018 RULE019
RULE146 RULE057 RULE145)

CHANGE-LANE-OK [ROAD-PARMS]

.....

TRANS: (*did change lanes or pass properly)

UPDATED-IN: NIL

USED-BY, (RULE101 RULE102)
UPDATED-BY, (RULE061 RULE062 RULE063 RULE
 064 RULE066 RULE067 RULE068
 RULE069 RULE070 RULE065)

CHECK-BLIND-SPOT [ROAD-PARMS]

.....
**TRANS, (*turned around to check traffic behind
him)**

PROMPT, T

USED-BY, (RULE061 RULE062 RULE063 RULE064)

CORRECT-SIGNAL [ROAD-PARMS]

.....
TRANS, (*signaled his move properly)

PROMPT, T

USED-BY, (RULE049 RULE050 RULE051 RULE052
 RULE053 RULE054 RULE072 RULE074
 RULE075 RULE077 RULE079 RULE082
 RULE061 RULE062 RULE063 RULE064
 RULE065)

CURB-PARK-OK [ROAD-PARMS]

.....
TRANS, (*parked properly at the curb)

UPDATED-IN, NIL

USED-BY, (RULE105 RULE106)

UPDATED-BY: (RULE077 RULE078 RULE079 RULE
080 RULE081 RULE082)

DRIVE-LANE-OK [ROAD-PARMS]

.....

TRANS: (*drove in lane correctly)

USED-BY: (RULE099 RULE100)

UPDATED-BY: (RULE146 RULE057 RULE058 RULE
144 RULE145 RULE143)

DUMMY-R [ROAD-PARMS]

.....

TRANS: (the dummy that determines entry into the
subordinate context of RPROPER)

DICTIONARY: INTERNAL

EXPECT: (I E D C T CP)

USED-BY: (RULE138 RULE139)

UPDATED-BY: (RULE024 RULE025 RULE097 RULE
098 RULE099 RULE100 RULE101 RULE
102 RULE103 RULE104 RULE105 RULE
106 RULE121 RULE122)

EMERG-VEH [ROAD-PARMS]

.....

TRANS: (an emergency vehicle did come down the
road)

PROMPT: T

USED-BY, (RULE017 RULE019)

ENTRY-EXIT-OK [ROAD-PARMS]

.....

TRANS, (*did enter or exit the road properly)

UPDATED-IN, NIL

ANTECEDENT-IN, NIL

USED-BY, (RULE097 RULE098)

**UPDATED-BY, (RULE049 RULE050 RULE051 RULE
052 RULE053 RULE054)**

LANE-OPENING [ROAD-PARMS]

.....

**TRANS, (there was a proper traffic opening in the
lane)**

PROMPT, T

**USED-BY, (RULE049 RULE050 RULE075 RULE076
RULE078 RULE079 RULE061 RULE062
RULE063 RULE064)**

MANEUVER [ROAD-PARMS]

.....

TRANS, (the road maneuver*did execute)

**PROMPT, (What road maneuver did * execute when
the accident occurred?)**

ASKFIRST, T

EXPECT, (NON-SIGNAL-INTERSECTION SIGNAL

-INTERSECTION ENTRY-EXIT DRIVE-IN-LANE CHANGE-OR-PASS TURN-CURB-PARK)

USED-BY, (RULE024 RULE025 RULE097 RULE098
RULE099 RULE100 RULE101 RULE102
RULE103 RULE104 RULE105 RULE106
RULE121 RULE122)

MECH-SAFE [ROAD-PARMS]

.....

TRANS, (the car was safe for driving)

PROMPT, T

REPROMPT, (Whether a car is safe for driving depends upon its condition, especially the condition of tires, brakes, mirrors and lights.)

USED-BY, (RULE020 RULE021 RULE022 RULE023)

OBEY-SIGN [ROAD-PARMS]

.....

TRANS, (*obeyed the sign)

PROMPT, T

USED-BY, (RULE052 RULE053 RULE071 RULE074
RULE075 RULE081 RULE082 RULE067
RULE068 RULE069)

OBEY-SIGNAL [ROAD-PARMS]

.....

TRANS: (*obeyed the signal lights)

USED-BY: (RULE121 RULE122)

UPDATED-BY: (RULE041 RULE042 RULE043 RULE
044 RULE045 RULE046 RULE047
RULE048)

OPPOSING-TRAFFIC [ROAD-PARMS]

.....

TRANS: (there was a lot of traffic in the opposite direction)

PROMPT: T

USED-BY: (RULE036 RULE038 RULE040 RULE073
RULE074 RULE047 RULE048)

OTHER-CAR-TOWING [ROAD-PARMS]

.....

TRANS: (the car ahead was towing a vehicle or a big truck)

PROMPT: T

USED-BY: (RULE058 RULE144 RULE145)

PROPER-LANE [ROAD-PARMS]

.....

TRANS: (*drove in the appropriate lane)

PROMPT: T

USED-BY: (RULE074 RULE075)

RIT-O-WAY-OK [ROAD-PARMS]

.....

TRANS: (*did follow the right of way laws)

USED-BY: (RULE024 RULE025)

UPDATED-BY: (RULE028 RULE029 RULE030

 RULE031 RULE032 RULE033

 RULE034 RULE035 RULE036

 RULE037 RULE038 RULE039

 RULE040)

ROAD-SAFE [ROAD-PARMS]

.....

TRANS: (road conditions were safe for driving)

PROMPT: T

REPROMPT: (Factors affecting road safety are visibility, the slipperiness of the road and obstructions. Visibility and slipperiness, in turn, may be affected by weather conditions.)

USED-BY: (RULE020 RULE021 RULE022 RULE023)

SAFE [ROAD-PARMS]

.....

TRANS: (conditions were safe for driving)

USED-BY: (RULE015 RULE016 RULE017)

UPDATED-BY: (RULE020 RULE021 RULE022

RULE023)

SPEED-VS-LANE [ROAD-PARMS]

.....
TRANS: (the speed the car did go with respect to traffic in the same lane)

PROMPT: T

EXPECT: (SLOWER SAMESPEED FASTER)

USED-BY: (RULE016 RULE017)

SPEED-VS-RLANE [ROAD-PARMS]

.....
TRANS: (the speed of*relative to the lane he was passing)

PROMPT: T

EXPECT: (FASTER SAMESPEED SLOWER)

USED-BY: (RULE067 RULE068 RULE069 RULE070)

SPEED-ZONE [ROAD-PARMS]

.....
TRANS: (the speed limit of the zone where the accident occurred)

PROMPT: T

EXPECT: POSNUMB

USED-BY: (RULE015 RULE017 RULE018)

TOW [ROAD-PARMS]

.....

TRANS: (*was towing a vehicle or other object)

PROMPT: T

USED-BY: (RULE057 RULE058 RULE144 RULE145
 RULE143)

TURN-OK [ROAD-PARMS]

.....

TRANS: (*made a turn properly)

UPDATED-IN: NIL

ANTECEDENT-IN: NIL

USED-BY: (RULE103 RULE104)

UPDATED-BY: (RULE071 RULE072 RULE073
 RULE074 RULE075 RULE076)

VISIBILITY [ROAD-PARMS]

.....

TRANS: (the view*had of traffic and the road)

PROMPT: (What sort of view did* have of traffic
 and the road?)

EXPECT: (GOOD FAIR POOR)

USED-BY: (RULE067 RULE068)

Parameter Group

SIG-INTERSECT-PARMS

INTERSECT-TRAFFIC [SIG-INTERSECT-PARMS]

.....
TRANS: (there was traffic moving in the cross street)

PROMPT: T

ASKFIRST: T

USED-BY: (RULE043 RULE044)

SI-MAN [SIG-INTERSECT-PARMS]

.....
TRANS: (the maneuver*did at the signal-controlled intersection)

PROMPT: T

ASKFIRST: T

EXPECT: (L-TURN R-TURN U-TURN CROSS)

USED-BY: (RULE041 RULE042 RULE043 RULE044
 RULE045 RULE046 RULE047 RULE048)

T-SIGNAL [SIG-INTERSECT-PARMS]

.....
TRANS: (the color of the traffic light at the intersection)

PROMPT: (What was the color of the traffic light at the intersection?)

ASKFIRST: T

EXPECT: (RED YELLO GREEN)

USED-BY: (RULE041 RULE042 RULE043 RULE044
 RULE045 RULE046 RULE047 RULE048)

Parameter Group TURNING-PARMS

T-MAN [TURNING-PARMS]

.....

TRANS, (the turning maneuver*did execute)

PROMPT, T

ASKFIRST, T

EXPECT, (L-TURN U-TURN ENTER-TRAFFIC)

USED-BY, (RULE073 RULE074 RULE075 RULE076)

System parameters

DOMAIN [SYSVARS]

.....

Value, "CARS"

TREEROOT [SYSVARS]

Value, DRIVER

System parameters

CHANGE-PASSRULES [RULEGROUPS]

.....

CONTEXT, (CHANGE-PASS)

SVAL, (change-pass)

CTRANS, "change-passes"

**Value, (RULE061 RULE062 RULE063 RULE064
RULE065 RULE066 RULE067 RULE068
RULE069 RULE070)**

CURB-PARKINGRULES [RULEGROUPS]

.....
**CONTEXT, (CURB-PARKING)
SVAL, (curb-parking)
CTRANS, "curb-parkings"
Value, (RULE077 RULE078 RULE079 RULE080
RULE081 RULE082)**

DRIVE-IN-LANERULES [RULEGROUPS]

.....
**CONTEXT, (DRIVE-IN-LANE)
SVAL, (drive-in-lane)
CTRANS, "drive-in-lanes"
Value, (RULE057 RULE058 RULE143 RULE144
RULE145 RULE146)**

DRIVERRULES [RULEGROUPS]

.....
**CONTEXT, (DRIVER)
SVAL, (driver)
CTRANS, "drivers"
Value, (RULE001 RULE002 RULE003 RULE004
RULE005 RULE006 RULE007 RULE008)**

RULE009 RULE010 RULE118 RULE120
RULE129 RULE130 RULE132 RULE133
RULE134 RULE135 RULE136 RULE137)

ENTRY-EXTRULES [RULEGROUPS]

.....

CONTEXT: (ENTRY-EXIT)

SVAL: (entry-exit)

CTRANS: "entry-exits"

Value: (RULE049 RULE050 RULE051 RULE052
RULE053 RULE054)

INTERSECTION [RULEGROUPS]

.....

Value: NIL

INTERSECTIONRULES [RULEGROUPS]

.....

CONTEXT: (INTERSECTION)

SVAL: (intersection)

CTRANS: "intersections"

Value: NIL

NON-SIG-INTERSECTRULES [RULEGROUPS]

.....

CONTEXT: (NON-SIG-INTERSECT INTERSECTION)

SVAL: NIL

**Value, (RULE030 RULE031 RULE032 RULE033
RULE034 RULE035 RULE036 RULE037
RULE038 RULE039 RULE040 RULE028
RULE029)**

PNOTCONTROLLEDRULES [RULEGROUPS]

.....

CONTEXT, (NOTCONTROLLED)

SVAL, (notcontrolled)

CTRANS, "notcontrolleds"

Value, NIL

PPARKING-LOTRULES [RULEGROUPS]

.....

CONTEXT, (PARKING-LOT)

SVAL, (parking-lot)

CTRANS, "parking-lots"

**Value, (RULE083 RULE084 RULE085 RULE086
RULE087 RULE088 RULE089 RULE090
RULE092)**

PRIT-O-WAY-OKRULES [RULEGROUPS]

.....

SVAL, NIL

CONTEXT, (NOTCONTROLLED)

Value, NIL

ROADRULES [RULEGROUPS]

.....

CONTEXT: (ROAD)

SVAL: (road)

CTRANS: "roads"

Value: (RULE015 RULE016 RULE017 RULE018
 RULE019 RULE020 RULE021 RULE022
 RULE023 RULE024 RULE025 RULE097
 RULE098 RULE099 RULE100 RULE101
 RULE102 RULE103 RULE104 RULE105
 RULE106 RULE138 RULE139 RULE121
 RULE122)

SIG-INTERSECTRULES [RULEGROUPS]

...

CONTEXT: (SIG-INTERSECT)

SVAL: (sig-intersect)

CTRANS: "sig-intersects"

Value: (RULE041 RULE042 RULE043 RULE044
 RULE045 RULE046 RULE047 RULE048)

SIGNALRULES [RULEGROUPS]

.....

CONTEXT: (SIGNAL)

SVAL: (signal)

CTRANS: "signals"

Value, NIL

TURNINGRULES [RULEGROUPS]

.....

CONTEXT, (TURNING)

SVAL, (turning)

CTRANS, "turnings"

**Value, (RULE071 RULE072 RULE073 RULE074
RULE075 RULE076)**

System parameters

CHANGE-PASS-PARMS [PARMGROUPS]

.....

Value, (C-MAN)

CONTEXTTYPES[PARMGROUPS]

.....

**Value, (DRIVER ROAD ENTRY~EXIT DRIVE-IN
-LANE CHANGE-PASS TURNING CURB
-PARKING PARKING-LOT NON-SIG-INTER-
SECT SIG-INTERSECT)**

CURB-PARKING-PARMS [PARMGROUPS]

.....

Value, (CURB-SPACE CP-MAN)

DRIVE-IN-LANE-PARMS [PARMGROUPS]

.....
Value: (2-RIGHT-LANES OVER-300-FT)

DRIVER-PARMS [PARMGROUPS]

.....
Value: (DRIVER-NAME RESPONSIBLE JUDGEMENT
APPROP-SPEED P-APPROP-SPEED P-PRO-
PER DRUG-FREE SOBER CALM ALERT R
-APPROP-SPEED R-PROPER PROPER-MAN-
EUVER LOCATION DUMMY-AS DUMMY
-PM)

ENTRY-EXIT-PARMS [PARMGROUPS]

.....
Value: (E-MAN)

NON-SIG-INTERSECT-PARMS[PARMGROUPS]

.....
Value: (ENTRY-TIME OTHER-CAR-LOC YIELD
-SIGN YIELD NSI-MAN)

PARKING-LOT-PARMS [PARMGROUPS]

.....
Value: (PCAR-SPEED PSAFE ROOM DIRECTION-OK
WITHIN-SPACE P-MAN)

ROAD-PARMS [PARM GROUPS]

.....

Value: (EMERG-VEH SAFE SPEED-VS-LANE SPEED
-ZONE CAR-SPEED ROAD-SAFE MECH-SAFE
CURB-PARK-OK TURN-OK CHANGE-LANE
-OK DRIVE-LANE-OK ENTRY-EXIT-OK
OPPOSING-TRAFFIC LANE-OPENING CO-
RECT-SIGNAL OBEY-SIGN OTHER-CAR
-TOWING CAR-DIST TOW CHECK-BLIND
-SPOT VISIBILITY SPEED-VS-RLANE PRO-
PER-LANE MANEUVER DUMMY-R R1T-O
-WAY-OK OBEY-SIGNAL CAR-LENGTHS)

SIG-INTERSECT-PARMS [PARM GROUPS]

.....

Value: (T-SIGNAL INTERSECT-TRAFFIC SI-MAN)

TURNING-PARMS [PARM GROUPS]

.....

Value: (T-MAN)

附录E 可在个人计算机上运行的 专家系统开发工具

这里介绍的各种专家系统开发工具，都具有以下两个共同特点：能在个人计算机上运行；用户不用编程就能建立专家系统。下面，我们将按功能强弱以及市面价格把这些软件包分为高、中、低三档，并逐个进行介绍。

E.1 中档产品

ES/P ADVISOR

在IBM PC及其兼容机上运行，要求有256K内存，并且最好配有硬磁盘。它能够处理近300条规则，这些规则需由外部的字处理软件编辑、输入。推理方式采用反向推理，它不能处理不确定性问题。使用该工具时，最好能了解一些有关PROLOG语言的知识。

设计者：Expert Systems International, 1700 Walnut Street, Philadelphia, PA 19103, (215)735-8510.

ExpertEase

它可以在IBM PC和一部分兼容机上运行，只需要128K内存和两个软磁盘驱动器。它不能在MS-DOS上工作，而需要UCSD-P操作系统的支持。具有归纳机制但不能对自己的推理进行解释。

设计者：Human Edge Software Corporation, 2445

Faber Place, Palo Alto, CA 94303; 1-800-624-5227(「美」
加利福尼亚州1-800-824-7325)

ExpertEdge

它能在具有512K内存和两个软磁盘驱动器的IBM PC及其兼容机上运行，采用演绎推理方式，用贝叶斯方法处理不确定性。它还可以处理等式，并具有解释推理过程和向用户提供帮助等功能。该产品可以接受用自然语言描述的数据。

设计者：同上。

M.1

它在具有512K内存的IBM PC及其兼容机上运行，是一个具有正向推理能力的反向推理系统，可以处理大约1 000条规则。它采用一种“英文知识表示语言”描述规则和事实。据它的设计者称，该系统是专门针对“非熟练编程人员”设计的。目前，国内已有汉化版本，称为CM.1.

设计者：Teknowledge, Inc., 525 University Avenue,
Palo Alto, CA 94301; (415) 327-6600.

INSIGHT2+

在IBM PC及其兼容机上运行。它采用IF/THEN/ELSE规则表示知识，既可做正向推理也可做反向推理，能处理近2 000条规则。它能处理确定性因子并能给出推理解释，可以做等式处理，并能够与dBASE II等其它外部程序联接。

设计者：Level 5 Research, 4980 S Hwy A1A, Melbourne FL 32951; (305) 729-9046.

KDS2

能在具有512K内存的IBM PC及其兼容机上运行，该系统可对例子进行归纳，并形成带有确定性因子的规则。据设计者称，它可处理的规则数目多达16 000条。它既能做正向推理也

能做反向推理，还具有元规则能力。它的人机接口语言是英语。

设计者：KDS Corp., 932 Hunter Road, Wilmette, IL 60091; (312)251-2621.

Personal Consultant (电脑顾问)

可在 TI PC, IBM PC 及其兼容机上运行，需要 768K (在 IBM PC 上需要 640K) 内存和一个硬磁盘驱动器。它能够处理 800 多条规则，能做正向和反向推理。它采用结构化的属性，可以处理等式，提供解释能力，还可以与其它软件联结。

设计者：Texas Instruments Co., 12501 Research Blvd., MS 2223, Austin, TX 78769; (512)250-7533.

XSYS

据设计该系统的公司声称，该软件在有 640K 内存和一个数学协处理器的 IBM PC 机上可以处理的规则数量不受限制。它采用反向推理，支持确定性因子，能提供解释能力。人机接口可以是英语文本、代数表达式或使用“菜单”选择。

设计者：California Intelligence, 912 Powell St. #8, San Francisco, CA 94108; (415)391-4846.

1STCLASS

在具有 256K 内存的 IBM PC 及其兼容机上运行，能进行归纳和综合正向、反向两种推理方式，能够处理不确定性。具有解释推理的能力，还拥有一个图形知识库编辑器，且效率较高。

设计者：Programs in Motion, 10 Syncamore Road, Wayland, MA 01778-9990; (508)266-1652.

E.2 低档产品

Advisor

可在个人计算机Commodore644/128、Apple II、Atari 800上运行。它的知识描述语句为简单的是非判断形式。它能处理大约255条规则，可做正向和反向推理，并能不断地提供解释。还可以与其它软件联结，但不能进行数学处理。

设计者：Ultimate Media Inc. 275 Magnolia Avenue, Larkspur, CA 94939; (415)924-3644.

INSIGHT1

在IBM PC、DEC Rainbow 和 Victor 9000 上运行，需要128K内存和一个软磁盘驱动器。可用 IF/THEN/ELSE 型规则做正向和反向推理。

设计者：Level 5 Research, 4980 S.Hway A1A, Melbourne, FL 32951; (305)729-9046.

Micro Expert

可在 Apple II、Macintosh 和 IBM PC 上运行，需要 128K (Apple II版本只需要 64K) 内存。能容纳 400 条规则，采用简单的反向推理。需要用外部的文本编辑器输入、编辑规则。

设计者：McGraw-Hill Book Co., Professional and Reference Div, 1221 Avenue of the America, New York, N.Y.10020; (212)512-2000.

E.3 高档产品

GURU

在IBM PC及其兼容机上运行，需要 512K 内存和一个硬磁盘驱动器。它是集成化的软件包，带有文字处理程序、电子表格、数据库管理、图形和通信程序。此外，由于具有自然语言理解能力，使专家可以方便地与系统交换数据和命令。由它生成的专家系统可使用规则和元规则进行正向和反向推理，并可以用几种不同的方式处理不确定性，它还能为推理提供解释。

并且能处理的规则数目不受限制。

设计者：Micro-Data Base Systems, Inc. P.O. Box 248,
Lafayette, Indiana 47902; 1-800-344-5382.

KES

这种高级软件已经在许多大型机上实现，同时也可在 IBM PC/XT 上运行。它是一种混合系统：它使用 IF/THEN 型规则进行反向推理，在一种语义网络型结构中使用贝叶斯公式进行统计模式分析；为了求解诊断型问题使用框架进行假设推理和测试。它能够处理非常大的知识库。

设计者：Software Architecture and Engineering, 1500 Wilson Blvd, Suite 800, Arlington, VA 22209; (703)276-7910.

NEXPERT

在具有 512K 内存的 Macintosh 微机上运行。它能采用反向和正向结合的方式进行推理，能处理约 500 条规则。在咨询执行过程中，它能提供图形化的规则逻辑流程显示和解释推理的功能，具有处理不确定性的能力。

设计者：Neuron Data Corp., 444 High Street, Palo Alto, CA 94301; (415)321-4488.

Personal Consultant Plus (增强型电脑顾问)

它是“电脑顾问”的高级版本，它使用框架组织问题领域，具有处理元规则的能力。此外，它还具有对其他软件的调用能力，并采用独具特色的下拉式菜单作为主要人机接口。

设计者：Texas Instruments Co., 12501 Research Blvd, MS 2223, Austin, TX 78769, (512)250-7533.

SAGE

可在具有 256K 内存的 IBM PC 及其兼容机上运行。它拥有一种非常容易使用的语言，使用这种语言可以建立规则数目不

受限制的知识库。它能处理异或关系，做正向推理，使用贝叶斯公式做不确定性演算并能够提供对推理的解释。

设计者：Systems Designers Software Inc., 444 Washington Street, Suite 407, Woburn, MA01801; (617)935-3070.

TIMM-PC

在IBM PC及其兼容机上运行，需要640K内存和一个10MB（10兆）硬盘。它能从例子数据中归纳得到规则，并能处理近500条规则。专家可以直接使用英语与系统对话，并通过这种对话建立和“训练”专家系统，它也可以处理不确定信息。

设计者：General Research Corp., 7655 Old Springhouse Rd, McLean, VA22101; (703)893-5915.

AutoES

AutoES是本书编译者在系统地分析目前较有影响的几种专家系统工具的基础上，结合自己多年工作体会和当前专家系统工具的技术发展趋势研制成的中英文通用的专家系统开发工具。它可以在IBM PC/XT、/AT及其兼容机上运行。

在知识表示方面，AutoES提供了框架、参数、规则、过程和方法等知识表示成分。框架可以有效地分割不同类型的知识并将它们组织成层次结构。参数用来描述事实，规则描述事实之间的逻辑关系。过程和方法则用来描述与任务有关的特殊处理功能。

在推理方面，AutoES支持正向推理、反向推理以及两者的混合。另外，利用框架中的继承性和特殊的推理控制原语可使推理效率提高。

AutoES的知识库开发机制别具特色，它向用户提供一套方便的菜单和多窗口操作界面，通过这个界面，用户可以直观

地观察到框架的树型结构以及各框架中的知识库元素；各种知识库元素都可以利用类似本书叙述的模版进行输入和修改；知识库的调试可按本书介绍的几种方法进行，并能从显示屏上直观地监视推理过程。

设计者：长沙国防科学技术大学，系统工程与数学系信息
系统工程教研室。邮政编码410073.

参 考 文 献

- [1] Minsky, Marvin, "A Framework for Representing Knowledge" in "Psychology of Computer Vision", Ed. P.H. Winston, pp.211-277, McGraw-Hill, NY. 这是一篇作者论述人类思维理论和框架概念的著名论文。
- [2] Duda, R.O., Hart, P.E., Barrett, P., Gaschnig.J., Konolige, K., Reboh, R., and Slocum, J., "Development of the PROSPECTOR Consultant System for Mineral Exploration", Final Report for SRI projects 5821 and 6415, Artificial Intelligence Center, SRI International. 在这篇技术报告中，参考和引用了许多有关PROSPECTOR（勘探者）专家系统的文献。
- [3] Buchanan, Bruce G., Shortliffe, Edward H., "Rule-Based Expert Systems, The MYCIN Experiments of the Stanford Heuristic Programming Project", Addison-Wesley Publishing Company, Reading, MA, 1984. 这是一部十分难得和完整的珍贵文献，其中包含的所有论文都是由曾经直接参加过MYCIN医疗诊断专家系统开发的学者撰写的。这些论文不但全面、深入地介绍了有关专家系统开发的各个方面，而且通俗易懂，可读性很强，即使是编程经验甚少的读者也不会发生很大困难，因此，值得一读。对于想深入了解有关课题细节的读者，本书提供了丰富、全面的文献目录可供查阅。
- [4] McDermott, J., "R1:A Rule-Based Configurer of Computer Systems", Computer Science Dept., Carnegie Mellon U., Pittsburgh, PA 15213, 1980. 这是一篇论述如何开发正向推理专家系统的优秀论文，它是由专家系统 R1 的研制者亲自撰写的。
- [5] Bennett, J.S., Creary, L., Engelmore, R., and Melosh, R.,

"SACON: A Knowledge-Based Consultant for Structural Analysis", Report No. HPP-78-23, Computer Science Dept, Stanford University, 1978. 该技术报告介绍如何通过专家系统指导用户使用大型仿真程序包。但如果你对结构分析技术不熟悉，要读懂它也许有一定的困难。

- (6) Cohen, Daniel, "Re: Thinking", M. Evans and Co., NY, 1982. 一本介绍如何科学地思维的好书，其中包括许多对逻辑悖论的精彩讨论。
- (7) Harmon, Paul and King, David, "Expert Systems", John Wiley and Sons, NY, 1985. 以行政管理人员为对象，对专家系统进行了内容广泛的综述性介绍。
- (8) K.J. Gilhooly, "Thinking—Directed, Undirected and Creative", Academic Press, NY, 1982. 这是一本关于用信息处理方法求解问题的精彩论著。它既包括一般性问题，也有疑难问题。论述了演绎和归纳两种问题求解方法，以及怎样培养创造性思维能力等内容。
- (9) Janet Goldenberg, "Experts on Call", PC World, Sept. 1985, pp.192-201. 专家系统商品汇编。
- (10) Patrick Henry Winston, "Artificial Intelligence", Addison Wesley Press, 1984. 一本关于人工智能基础的教科书。
- (11) Robert H. Michaelson, Donald Michie, and Albert Boulanger, "The Technology of Expert Systems", in Byte, April, 1983. 在这篇论文中，作者详细地介绍了什么是专家系统以及它的工作原理。
- (12) Donald Michie, editor, "Introductory Readings in Expert Systems", Gordon and Breach, London, and New York, 1982. 在本书中，有多篇写得很好的基础性论文和综述。

索引

二 面

- 人机对话 system-client communication, 17
人机对话 system-client communication, 20
“开门”评价 getting others to evaluate, 159
改善～ improving the, 148
协调交互作用 harmonizing the interaction, 159
使咨询变得实用 pragmatize the advice in, 156
使提问的次序合理 ordering the questions, 150
提问次序安排 question sequencing in, 108
阐明对话内容 clarifying the dialogue in, 148
人-机接口 system-developer interface
 输入规则还是例子 entry rules or examples in, 179
 ～语言 language of, 178

三 面

- 上下文 context
 ～的概念 concept of, 39
 ～实例 instances of, 40
 子辈～ offspring of, 39
 父辈～ parent, 39
 根～ root, 39
上下文干扰 context violation
 检验～的规则 rules for testing, 80 (图6-3)
上下文树 context tree, 38, 65
 ～推理流程图 inference flow diagram, 67 (图5-7)
个人计算机 personal computers
 ～上运行的专家系统开发工具 expert-system builders for, 318
“已知”参数(属性)谓词 IS KNOWN, 104
子句 clauses, 44

四 画

- “为什么”命令 WHY, 112
计划层 planning level, 27
～的应用 applications in, 31
开发机 development engine, 97
元规则 meta rules, 91
专家系统 expert systems
 人机对话 system-client communication, 17,20
 开发～ developing, 5 (图1-1)
 分析～的收益与开销 analyzing benefits vs. costs of, 165
 ～在经营管理领域的应用 business applications for, 22
 完备性 completeness of, 142
 估计是否拥有必要的经验知识 deciding on necessary know-how, 164
 ～的组合领域应用 combined-domain applications for, 35
 ～的定义 definition of, 5
 ～的开发阶段 developmental phases of, 47
 ～的效率 efficiency of, 145
 ～推理的解释 explanations in, 19
 ～的可维护性 maintainability of, 167
 ～的主要特征 major characteristics of, 20,167
 ～开发计划 planning, 47,.61
 ～的预研 preliminary study for, 162
 ～的可行性研究 system feasibility study for, 165
 ～的逻辑正确性 validity of, 142
选择经营管理应用 choosing business applications for, 166
选择用于构造原型的领域 choosing domain for prototype, 164
选购专家系统开发工具 purchasing expert-system builder, 165
测试知识库 testing knowledge base of, 56
制定工程进度表 preparing schedule for, 165
确定～领域 defining the domain of, 89,166
建立～的知识库 building the knowledge base of, 54
建立实用～ developing a practical system for, 167
建立试验原型 developing prototype for, 165
专家系统开发工具 Expert-System Builder, 5 (图1-1),97(图8-1),173
～的人机接口 system-developer interfacing in, 178
～的文法 grammar of, 173
～的选择因素 choice factors in, 173

~处理不确定性的方法 method of handling uncertainty in, 177
个人计算机与~ personal computers and, 318
开发自我 developing Your expertise for, 184
求实 objectivity in, 180
严谨 logic in, 180
推理方式 reasoning method in, 174
控制结论对CF的敏感性 expertcontrol of CF sensitivity, 178
能否表示不确定性 certainty or uncertainty in, 177
专家系统领域 expert system domain
关系的可交流性 communicability of, 35
关系的可表述性 expressible relationships in, 34
关系和数据的不确定性 uncertain relationships and data in, 33
分割~ compartmentalizing the, 50
示范实例 example of, 59
因素的可确认性 identifiable factors in, 34
问题的可分解性 decomposable problems in, 33
合成人机对话 composing system-user communication,
没有严格的求解方法 non-rigorous solution approaches in, 33
评价专家经验的准则 testing your know-how for, 34
完整的问题领域和~ 35 (图2-8)
经营管理~中的知识 business knowledge of, 163
易变性 change in, 34
重建上下文树 redoing context tree of, 65
重绘推理流程图 redoing inference flow diagram of, 66
领域的有界性 boanded domain in, 32
检验~的准则 criteria for testing, 32
确定~ defining the, 59
确定推理流程 defining inference flow in, 61
确定性能指标 defining performance standards of, 68
不正确的类比 false analogy, 182
不确定性 uncertainty, 19,33,92
合成事实与规则的~ combined fact-rule, 93
不成立的排中假设 false dilemma, 182
不自然的提问 awkward questions, 149
分类 categorization, 24
分析 analysis, 24
经营管理中的~, 27
分析与综合 analysis vs.synthesis, 23 (图2-1)
反向推理 backward reasoning, 14,88,89
反向链接 backward chaining, 14,88,89

风险投资规则 IRA investment rules, 12 (图1-5), 13 (图1-6),
175 (图11-10), 176 (图11-11)

五 画

- 汉化M.I CM.I, 319
正向推理 forward reasoning, 14, 87
正向链接 forward chaining, 14
可推演事实 inferable facts, 23 (图2-1)
可信因子 confidence factor, 45, 94 (图7-7)
计算如果部分的～ calculating for IF, 93
合成如果部分和则部分的～ combining IF and THEN, 93
检查～ checking, 137, 138 (图9-9)
可分解的问题 decomposable problem, 33
可观察事实 observable facts, 23 (图2-1)
布鲁斯·巴克兰 Buchanan, Bruce G., 96
“打印”函数 SPRINTT, 114
归纳推理 inductive reasoning, 16
目标 goals, 23
电脑顾问 Personal Consultant, 96
～的ARL命令 ARL commands for, 186
～的LISP命令 LISP commands for, 204
人机对话 developing system-user communication for, 108
专家系统开发工具 expert system builder, 173
对话 dialogue in, 111
列清单 listing in, 115
严格的还是不严格的概率演算 rigorous or nonrigorous probability calculation in, 178
定义 definitions in, 114
追踪逻辑流程 tracing logic strands of, 115
测试知识库 testing knowledge base of, 114
能否表示不确定性 certainty or uncertainty in, 177
能够控制最终结论对CF的敏感性 expert control of CF sensitivity, 178
调试 troubleshooting with the, 114
推理方式 reasoning method in, 174
概述 description of, 96
解释 explanations in, 115
增强型电脑顾问 Personal Consultant Plus, 180
处理 manipulation, 38

对话 dialogue, 108

~中提问的次序 ordering questions in, 109 (图8-9)

司机手册 Drivers Hand Book, 60

六 圆

交通事故原型 accident prototype

~的规则库 rule base for, 215-238

~的规则和参数(属性)清单 rule and parameter (attribute) listing, 239-317

~的属性表 attribute worksheet for, 75

交通事故图例 accident diagram, 63 (图5-3)

问题 problems

~的类型 class of, 22

可分解的~ decomposable, 33

人员~ people, 30

机器~ machines, 30

关系的可表述性 expressible relationship, 34

动作 action, 104

有界的领域 bounded domain, 32

过份概括 overgeneralization, 182

执行时实用程序 run-time utilities, 97, 98

执行层 operations level, 27

~的应用 applications of, 27

“优先提示”特性 PROMPTST, 110

血液中酒精含量图 blood-alcohol chart, 144

“似乎不是”值谓词 IS THOUGHTNOT, 102

后件规则 consequent, 91

后件驱动的推理 consequent driven reasoning, 91

合成事实与规则的不确定性 combined Fact-Rule uncertainty

计算如果部分的可信因子 calculating confidence factors for, 93

合成如果部分和则部分的可信因子 combine IF-PART and THEN ~PART CFs, 93

如果-则关系 if/then relationships, 10

如果-则规则结构 if/then rule structure, 11

七 圆

没有严格的求解方法 nonrigorous solution approaches, 33

启发式程序开发 heuristic programming, 95
诊断 diagnosis, 25
时间表 time frames, 48
“求助”命令 HELP, 111
含混不清的提问 unclear questions, 149
鸡生蛋，蛋生鸡逻辑循环 chicken and egg loop, 147

八 画

实际领域 pragmatic domain, 146
表示知识的词汇 knowledge vocabulary, 51,70
开发～ developing the, 51,70
用于检验词汇的“判断能力”规则 judgement rules to test, 76
(图8-4)
定义属性的值 defining attribute values of, 53,75
确定属性 defining attributes of, 53,70
编写检验词汇的规则 writing rules to test, 79
规则 rules, 10
前件～ antecedent, 9.
后件～ consequent, 91
元～ meta, 91,92 (图7-6)
不确定～ uncertain, 93
不确定性 uncertainty in 92
规则库 rule base, 55,120,83 (图7-1)
交通事故原型 accident prototype, 239-317
规则库图 rule-base diagram, 83 (图7-1)
规则模板 rule template, 43
～中的子句 clauses in, 44
～中的可信因子 confidence factor in, 45
～的例子 examples of, 46
～的函数 functions in, 44
规则簇 rule strand, 84,85
～的完备性 completeness of, 123
“有责任”～ responsible, 85 (图7-2)
记录～的活页卡 loose-leaf book of, 121
“判断能力”～ judgement, 86 (图7-3)
检查～逻辑 checking logic of, 131 (图9-6)
输入～的次序 sequence of entry, 127 (图9-4)
规则的构件 grain of rule, 172

规则的逻辑正确性 rule-logic integrity, 129
事实 facts, 6
可观察～ observable, 23 (图2-1)
可推演～ inferable, 23 (图2-1)
不确定～ uncertain, 92
事实模板 fact template, 40
～中的值 values in, 41
～中的谓词 predicates in, 41
～中的例子 examples of, 42
～中的属性 attributes in, 41
知识 knowledge
～的定义 definition of, 6
表示～ representing, 7
知识工程师 knowledge engineer, 3,184
知识库 knowledge base, 6,20
“开门”评价 getting others to evaluate the, 159
评价～ evaluating the, 58,159
评价～的一般准则 general criteria for, 142
改善人机对话 improving system-user communication in, 58
建立～ building the, 54,120
测试～ testing the, 56,140
准备一套测试案例 preparing test case library for, 58,140
准备书面规则库 recording rule base for, 55 (图4-3),120
获得清单 obtaining listing for, 128
检查可信因子 checking confidence factors of, 56,137
检查规则的逻辑正确性 checking rule logic of, 56,128
输入～元素 entering elements into, 55,106,126
输入上下文 entering contexts to, 106,126
输入规则 entering rules into, 106,128
输入参数(属性) entering parameters (attributes), 108
输入属性 entering attributes of, 128
避免～中常见的逻辑错误 avoiding logical fallacies in, 181
图形化规则语言 graphical rule language, 38
上下文树 context tree in, 38
处理 manipulation in, 38
事实模板 fact template, 40
经营管理专家系统 business expert system
～的上下文 contexts in, 169
～的可信因子 confidence factors in, 172
～的规则 rules in, 171

~的谓词与函数 predicates and functions in, 170
~的属性 attributes in, 169
经验知识 rule-of-thumb, 11
参数 parameter, 89
组合型系统 linked system, 36
函数 functions, 44-104
文本~ text, 105
用户定义~ user-defined, 105
逻辑~ logic, 105

九 画

测试 testing, 25
前提 premise, 98
前件规则 antecedent rules, 91
前件驱动的推理 antecedent driven reasoning, 91
咨询型系统 advisor, 36
语法 grammar
 算术运算符 arithmetic operators in, 100
 谓词 predicates in, 100
语义网络 semantic networks, 8,8 (图1-8)
封闭的盒子 closed box, 130 (图9-5)
“是确定的”参数(属性)谓词 IS DEFINITE, 104
“怎样推”命令 HOW, 114
“重现(追踪过程)”命令 PLAYBACK command, 119
复合领域 combined domain
 ~的应用 applications for, 35
 ~中的组合型系统 linked systems in, 36
 ~的咨询型系统 advisor, 36
信息 information, 6
“追踪”命令 TRACE, 118 (图8-12)
独立的推理机 separate reasoning capability, 20

十 画

调试 troubleshooting
~知识库 the knowledge base, 114
解释 explanations, 115
列清单 listing, 115
追踪逻辑流程 tracing logic strand, 116

框架 frames, 9 (图1-4), 180
配置 configuration, 26
爱德华·肖特利夫 Shortliffe, Edward H., 29
租凭规则 leasing rules, 29
值 value
多值 multi-, 41
单值 single, 41
定义数值属性的单位和范围 defining units and range for numeric attributes, 78
预定义～ predefining, 77
～的要求 requirements of, 75
使用是非型～ use of Yes/no, 77
有值属性的～ valued attributes and, 77
预测 predicting, 25

十一 画

“清醒的”规则簇 sobriety rule strand, 158 (图10-11)
谓词 predicates, 40, 100
值～ value, 100
参数(属性)～ parameter(attribute), 103
函数～ functions of, 104
数值～ numeric, 100
培训人员 training, 30
辅助属性 dummy attributes, 154
控制层 control level, 27
～的应用 applications of, 30
推理 reasoning
～过程 process of, 82
～的例子 examples of, 87, 89
反向～ backward, 14, 88, 176
反向～过程 backward,cycle of, 89
正向～ forward, 14, 87, 175
正向～过程 forward,cycle of, 87
后件驱动 consequent driven, 91
前件驱动 antecedent driven, 91
数据驱动 data driven, 14
归纳推理 inductive, 16
循环～，循环论证 circular, 181

推理流程 inference flow, 61,66
推理流程与上下文树的综合视图 inference flow-context tree diagram,
66 (图5-7)
推理机 inference engine, 98
推理的解释 explanations, 19
逻辑 logic
~的模拟 modeling the, 82
~错误 fallacies in, 181-183
使用“为什么”检查~ using WHY to check, 132
使用“怎样推”检查~ using HOW to check, 134
使用“追踪”检查~ using TRACE to check, 132
追踪~流程 locating flows in, 132,134
检查~正确性 checking, 133
逻辑流程 logical strands,
追踪~ tracing the, 115
逻辑联结符 logical connective, 45
逻辑正确性 integrity, 128
规则的~ rule-logic, 129
属性的~ attribute, 129
逻辑错误 logical fallacies
不正确的类比 false analogy, 182
不成立的排中假设 false dilemma, 182
过份概括 overgeneralization, 182
循环论证 circular reasoning, 181
错误的推断 false extrapolation, 182
偶发相关 accidental correlation, 182
偶发相关 accidental correlation, 182
领域的有界性 domain bounded, 32
弹出式菜单 pop-up menus, 180
综合 synthesis, 25
经营管理中的~ business use of, 27

十二 图

联结符 connective
算术的~ arithmetic, 45
逻辑的~ logical, 45
斯坦福启发式程序开发项目 Stanford Heuristic Programming Project, 96
“期望值”命令 EXPECT command, 111

提问 questions,

~顺序不合理 incongruous order of, 150

不必要的~ unneeded, 153 (图10-8), 155 (图10-9)

不自然的~ awkward, 149

含混不清的~ unclear, 149

重复的~ repetitious, 150

提问次序的安排 Question sequencing

定义上下文 defining contexts with, 108

定义可观察事实 defining observable facts with, 110

定义可推演事实 defining inferable facts with, 110

“提问优先”特性 ASKFIRST, 110

“提示”命令 PROMPT command, 110

搜索 search, 24

黑箱 black box, 5

链接 chaining

反向~ backward, 14, 88, 175

正向~ forward, 14, 88, 176

属性 attributes, 41

~的正确性 integrity of,

~的要求 requirements of, 70

~的肯定描述 affirmative, 71

~表 worksheet for, 72

~隶属于合适的上下文 within appropriate context, 71

定义~的值 defining values of, 75

没有冗余~ non-redundant, 71

属性-值表 attribute-value worksheet, 78

十三 画

简写式规则语言 Abbreviated Rule Language (ARL)

~中开发者定义的元素 user-defined elements of, 189

~中用自然语言表述的规则 English rule in, 200

~中系统定义的元素 system-defined elements of, 186

~中的变量 variable, 189

~中的UNIQUE特性 UNIQUE, 190

~到自然语言(英语)的翻译 translation to English, 200

~的上下文 contexts in, 189

~的“同义词”特性 SYN, 190

~的可信因子 confidence facts in, 188

~的命令 commands in, 194

~的命令使用实例 examples of command use in, 197
~的函数 functions in, 188
~的规则 rules in, 193
~的“显示名”特性 PRINTID, 190
~的参数(属性) parameters(attributes) in, 191
~的谓词 predicates in, 186
~的“提示”特性 PROMPT, 191
~的算术运算符 arithmetic operators in, 186
数值属性 numeric attributes, 78
数据 data, 6
数据驱动的推理 data-driven reasoning, 14
概率演算 probability calculations, 178
错误的推断 false extrapolation, 182

十四 画

算术联结符 arithmetic connective, 45

十五 画

德克萨斯仪器公司 Texas Instruments, 2,320,322

十八 画

翻译, 表述 translations, 148

Advisor专家系统开发工具 Advisor, 320
AutoCAD辅助绘图软件包 AutoCAD, 36
AutoES专家系统开发工具 AutoES, 323
California Intelligence公司 California Intelligence, 320
dBASE III数据库系统 dBASE III, 36
EMYCIN专家系统开发工具 EMYCIN, 96
ESP ADVISOR专家系统开发工具 ESP ADVISOR, 318
ExpertEase专家系统开发工具 ExpertEase, 318
ExpertEdge专家系统开发工具 ExpertEdge, 319
Expert Systems国际公司 Expert Systems International, 318
General Research公司 General Research Corp., 323
GURU事务处理专家系统工具 GURU, 321
Human Edge软件公司 Human Edge Software Corp., 318

- INSIGHT1专家系统开发工具 INSIGHT1, 323
INSIGHT2+专家系统开发工具 INSIGHT2+, 319
KDS公司 KDS Corp., 320
KDS2专家系统开发工具 KDS2, 319
KES专家系统开发工具 KES, 322
Level 5研究公司 Level 5 Research, 319
LISP语言 LISP
 上下文 contexts in, 207
 开发者定义的元素 developer defined elements of, 207
 可信因子OF confidence factors in, 207
 用户定义的函数 user defined function 213
 动作子句 action clauses in, 211
 命令实例 command examples in, 208
 规则 rules in, 207
 前提子句 premise clauses in, 209
 函数 functions in, 206
 变量 variables in, 207
 系统定义的元素 system defined elements of, 204
 值谓词 value predicates, 205
 谓词 predicates in, 204
 参数(属性)谓词 parameters(attributes)in, 205
 数值谓词 numeric predicates, 204, 208
 算术运算符 arithmetic operators in, 204
 算术表达式 arithmetic expressions 208
- Lotus 1-2-3组合软件 Lotus 1-2-3, 36.
- M.I专家系统开发工具 M.I, 319
- McGraw-Hill图书出版公司 McGraw-Hill Book Co., 321
- Micro-Data Base Systems公司 Micro-Data Base Systems Inc., 322
- Micro Expert专家系统开发工具 Micro Expert, 321
- MYCIN细菌感染咨询专家系统 MYCIN, 25, 98
- Neuron Data公司 Neuron Data Corp., 322
- NEXPERT专家系统开发工具 NEXPERT, 322
- Personal Consultant 专家系统开发工具(电脑顾问) Personal Consultant, 320
- Personal Consultant Plus专家系统开发工具(增加型电脑顾问) Personal Consultant Plus, 322
- Programs in Motion公司 Programs in Motion, 320
- PROSPECTOR(勘探者)专家系统 PROSPECTOR, 25
- SACON咨询专家系统 SACON, 36, 37

SAGE专家系统开发工具 SAGE, 322
SAS统计软件包 SAS, 36
Software Architecture and Engineering公司 Software Architecture and Engineering 322,
Systems Designers Software公司 Systems Designers Software Inc., 323
Teknowledge公司 Teknowledge Inc., 319
TJMM-PC专家系统开发工具 TJMM-PC, 323
Ultimate Media公司 Ultimate Media Inc., 321
VAX-11计算机系统 VAX-11, 26
XCON/R1专家系统 XCON/R1, 28
XSEL专家系统 XSEL, 26
XSEL/R1的配置规则 XSEL/R1 configuration rule, 26 (图2-4)
XSYS专家系统开发工具 XSYS, 320
1ST CLASS专家系统开发工具 1ST CLASS, 39