

3

项目

人工智能2.0时代：知识，让计算机更聪明

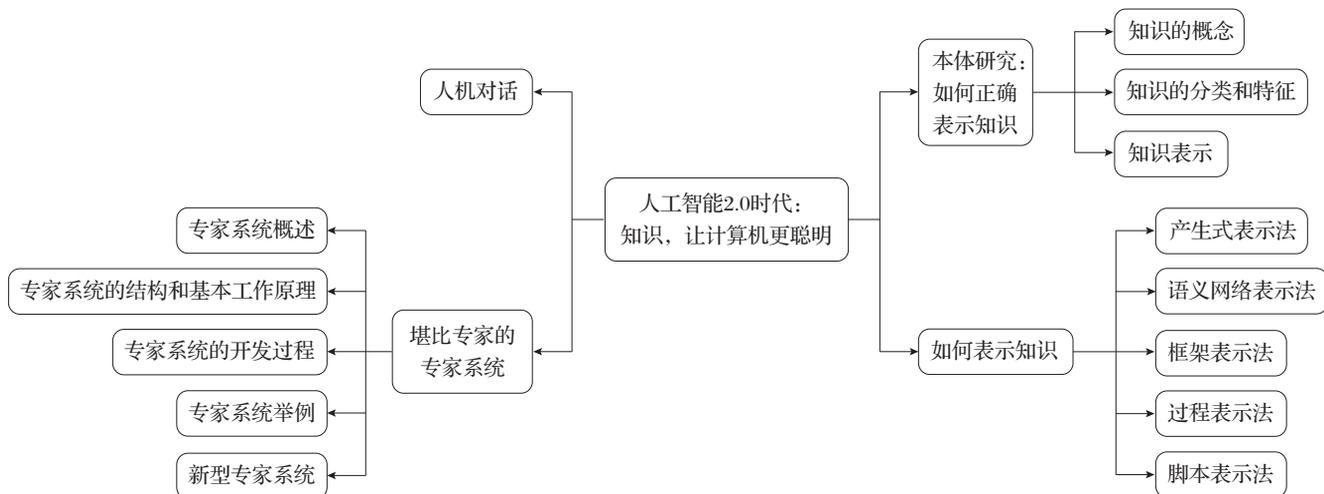


教学导航

- ★ 了解专家系统的定义、优点和应用领域。
- ★ 了解知识的概念、分类和特征。
- ★ 熟悉专家系统的结构和开发过程。
- ★ 掌握知识的几种表示方法。



知识脉络图



项目引入

人机对话的出现使人工智能从 1.0 时代跨越到 2.0 时代。尽管在 2.0 时代里人工智能经历了野蛮生长,但是“过度繁荣”的背后却未必就是顺利的发展。尽管人工智能已经有了极强的理论基础,但是要解决一个又一个的现实问题,还是要一步一个脚印地向前走。

任务 3.1 人机对话

在人工智能 1.0 时代里,人工智能已经能够解决一些诸如迷宫、梵塔问题等“玩具问题”。但是在解决这些问题的同时,人工智能的很多局限性也逐渐显现出来。对人类来说,如果人工智能只能解决“玩具问题”,那么它的价值不是很大,因此人工智能研究在 20 世纪 70 年代一度变得极为萧条、冷寂。到了 20 世纪 80 年代,由于一款程序的出现,人类的目光再次被吸引到了人工智能上,人工智能也因此得以卷土重来,这位“功臣”就是 Eliza。

在人工智能 1.0 时代里,人工智能主要是通过推理和搜索等简单的规则来处理问题;但是到了 2.0 时代,人工智能已经进化到了利用知识来武装自己。我们将知识的载体从图书转变成人工智能,这样它所能解决的就不再是“玩具问题”,并且看上去也变得更加聪明和实用。例如,如果想让人工智能取代厨师,只需要输入大量的菜品知识即可;如果想让人工智能取代医生,只需要输入大量的病理知识即可。

Eliza 是由系统工程师约瑟夫·魏泽堡和精神病学家肯尼斯·科尔比在 20 世纪 60 年代共同编写的。在当时自然语言技术还没有突破性进展的前提下,Eliza 的出现是一件让人“费解”的事情。

由于 Eliza 的血液中流淌着精神病学的血统,因此它首先被应用于精神类疾病的治疗。令人惊讶的是,在 Eliza 与病患聊天时,它不仅能够听懂病人的话,而且还富有同情心,会像知心朋友那样给予人安慰。很多心理学家和医生都想让 Eliza 为病人进行心理治疗,甚至有些病人与它谈话后,对它的信任程度已经远远超过了人类医生。它虽然不懂人类的喜怒哀乐,却能够让人类和它进行友好、亲密的交谈。

图 3-1 是 Eliza 与人类的聊天记录,如果仔细分析 Eliza 与人聊天的内容,就不难发现它的逻辑原理非常简单,甚至可以说其中并没有逻辑,它只是颠倒了对方所说的话的语序,然后回应给对话人。在 Eliza 运行过程中,它会扫描一些对方所说的话的关键词,然后根据系统内的“对应词”重新组织语言,最终成功与人类进行对话。当然,这种语序的重新组织、编排非常巧妙,不容易被人看出来。例如,你说“我很难过”,它会问你“为什么难过”;你说“我想大笑”,它会问你“为什么想笑”。不同的日常生活关键词会被划分成不同的等级,Eliza 通过这些等级可以快速从数据库中检索出相应的关键词,看这个词代表的意思是什么,应该怎么应对。如果 Eliza 遇到了一句完全陌生的话,那么它会用一种特殊的模式拖延时间或诱导你说出其他关键词。例如,它会问“你指的是什么”或“你能举个例子吗”,甚至有时它还会说一些像“这很有意思,请你继续”等“废话”来避免自己出“洋相”。

```

File Edit Options Buffers Tools Help
-----
I am the psychotherapist. Please, describe your problems.
Each time you are finished talking, type RET twice. 我是精神治疗师。
I have a terrible headache. 我的头很疼
Maybe you should consult a doctor of medicine, I am a psychi-
atrist. 也许你应该去找其他科室的医生,我是精神治疗师
Okay, I'll do that then. 好的,一会儿我会去的
Can you elaborate on that? 你能详细描述一下吗?
What are you talking about? No I cannot elaborate. See you la-
ter. 你现在说什么?我没法描述.再见
Why do you say that? 你为什么这么说?
Now you're just talking nonsense! 因为你说话莫名其妙!
What makes you believe now I am just talking nonsense?
You are not making any sense. 什么让你确信我说的是莫名其妙的话?
Perhaps your plans have something to do with this.
-u:*** *doctor* (Doctor Fill)--L1--Top-----

```

图3-1 Eliza与人类的聊天记录

从技术角度来说, Eliza 与人类的对话并不是在理解句子意思的基础上进行的, 但人类却对该系统青睐有加, 甚至很多人为此着迷。据说, 当时有人要查阅 Eliza 的对话记录时, 它还会“生气”地说: “你这是在侵犯别人隐私!”

虽然 Eliza 的运行原理没那么高深, 但是人们却产生了兴趣, 即使它只是利用单纯规则展开对话, 也不影响人们在使用时的乐趣。人工智能从一种只有科学家之间才聊的话题变成了一种人们茶余饭后的谈资。直到现在, 在微博或网络游戏中, 有很多被称为 BOT 的账号, 这些账号可以说是 Eliza 的进化体, 它们通过一定的计算规则不断地进行对话。

iPhone 中有一个名为 Siri 的手机助手, 很多人会在闲暇时与 Siri 聊天或在困惑时寻求 Siri 的帮助。使用 Siri 成为一种潮流, 甚至有些人已经离不开 Siri, 并将它当成生活的一部分。然而, 如果没有 50 年前 Eliza 的出现, 不知道这种潮流会被推迟到哪一年。

任务 3.2 堪比专家的专家系统

在人工智能的发展历史上, 比较著名的专家系统有 DENDRAL 和 MYCIN。其中, DENDRAL 是由美国斯坦福大学的费根鲍姆于 1968 年在美国斯坦福大学研制成功的, 它是第一个专家系统。DENDRAL 根据质谱仪产生的数据, 不仅可以推断出已经确定的分子结构, 而且可以说明未知的分子结构。MYCIN 是第一个功能较全的医疗诊断专家系统, 在 1974 年由美国斯坦福大学的 HPP 科研小组研制成功, 它能帮助医生对住院的血液感染患者进行诊断并选用抗生素类药物对患者进行治疗。从技术角度来看, 上述两个专家系统的研究彻底解决了知识表示、不精确推理、搜索策略、人机联系、知识获取及专家系统基本结构等一系列重大的技术问题。

3.2.1 专家系统概述

1. 专家系统的定义

专家系统是一类具有专门知识和经验的计算机智能程序系统, 通过对人类专家问题求解能力的建模, 采

用人工智能中的知识表示和知识推理技术来模拟通常由专家才能解决的复杂问题,达到具有与专家同等解决问题的水平。这种基于知识的系统设计方法是以知识库和推理机为中心而展开的,即

专家系统 = 知识库 + 推理机

专家系统把知识从系统中与其他部分分离开来。专家系统强调的是知识,而不是方法,因此专家系统是基于知识的系统。一般来说,一个专家系统应该具有以下3个要素:

- (1) 具备某个应用领域的专家级知识。
- (2) 能够模拟专家的思维。
- (3) 能达到专家级的解题水平。

2. 专家系统的优缺点

近30年来,专家系统迅速发展,应用领域越来越广,解决实际问题的能力也越来越强,这是由专家系统的优良性能决定的。具体来说,专家系统的优点包括以下几个方面:

- (1) 专家系统能够高效率、准确周到、迅速和不知疲倦地进行工作。
- (2) 专家系统解决实际问题时,不受周围环境的影响,也不可能遗漏或忘记。
- (3) 专家系统可使专家的专长不受时间和空间限制,以便推广珍贵和稀缺的专家知识与经验。
- (4) 专家系统能使各领域专家的专家知识和经验得到总结与精练,使其拥有更丰富的经验和更强的工作能力,能广泛、有力地传播专家的知识、经验和能力,促进各领域的发展。
- (5) 专家系统能汇集和集成多领域专家的知识与经验。
- (6) 军事专家系统能够提高国家的国防现代化和国防能力。
- (7) 专家系统的研制和应用具有巨大的经验效益和社会效益。
- (8) 研究专家系统能够促进整个科学技术的发展。

当然,专家系统不仅有优点,也有缺点,其缺点包括以下几个方面:

- (1) 没有自我学习能力。传统的专家系统不具备自我学习能力,只能通过知识工程师扩充知识库,更不具备在运行过程中自我完善、发展创新的能力。
- (2) 专业知识领域狭窄。一般的专家系统只能解决某个很小的知识领域内的问题,一旦超出范围,就无法解决。
- (3) 处理不确定性问题较差。专家系统采用可信度、主观贝叶斯方法处理不确定性问题,但在模糊推理、非完备推理等方面能力较差。
- (4) 搜索策略效率低。每个周期中,推理机要搜索所有规则,当规则较多时,系统实时性变差,速度降低。

3. 专家系统的应用领域

专家系统对人工智能各个领域的发展起到了很大的促进作用,并对科技、经济、国防、教育和人民生活产生了极其深远的影响。其应用领域概括如下:

- (1) 解释 (interpretation): 如肺部测试 (PUFF)。
- (2) 预测 (prediction): 如预测可能由黑蛾造成的玉米损失 (PLAN)。
- (3) 诊断 (diagnosis): 如诊断血液中细菌的感染 (MYCIN), 又如诊断汽车柴油引擎故障原因的 CATS 系统。
- (4) 故障排除 (fault isolation): 如电话故障排除系统 ACE。
- (5) 设计 (design): 如专门设计小型电动机弹簧与碳刷的专家系统 motorbrush designer。
- (6) 规划 (planning): 较出名的有辅助规划 IBM 计算机主架构的布置、重安装与重安排的专家系统 CSS, 以及辅助财物管理的 Plan Power 专家系统。
- (7) 监督 (monitoring): 如监督 IBM MVS 操作系统的 YES/MVS。

(8) 除错 (debugging)：如侦查学生减法算术错误原因的 BUGGY。

(9) 修理 (rep 人工智能 r)：如修理原油储油槽的专家系统 SECOFOR。

(10) 行程安排 (scheduling)：如制造与运输行程安排的专家系统 ISA，又如工作站 (Work Shop) 制造步骤安排系统。

(11) 教学 (instruction)：如教导使用者学习操作系统的 TVC 专家系统。

(12) 控制 (control)：如帮助 digital corporation 计算机制造及分配的控制系统 PTRANS。

(13) 分析 (analysis)：如分析油井储存量的专家系统 DIPMETER 及分析有机分子可能结构的 DENDRAL 系统。

(14) 维护 (M 人工智能 ntenance)：如分析电话交换机故障原因之后，即能建议人类该如何维修的专家系统 COMPASS。

(15) 架构设计 (configuration)：如设计 VAX 计算机架构的专家系统 XCON 以及设计新电梯架构的专家系统 VT 等。

(16) 校准 (targeting)：如校准武器如何工作。

4. 专家系统的特点

专家系统使用某个领域的实际专家经常使用的领域知识来求解问题，通常适用于完成那些没有公认的理论和方法、数据不精确或信息不完整、人类专家短缺或专门知识十分昂贵的诊断、解释、监控、预测、规划和设计等任务。一般专家系统具有以下特点：

(1) 启发性。专家系统能够运用专家的知识和经验进行推理、判断与决策。利用启发式信息找到问题求解的捷径，即专家系统不仅能使用逻辑知识，也能使用启发性知识，运用规范的专门知识和直觉的评判知识进行判断、推理和联想，实现问题求解。

(2) 透明性。专家系统具有透明性，是因为它具有解释功能，该功能用于向用户解释它的行为动机及得出某些答案的推理过程。这可使用户比较清楚地了解系统处理问题的过程及使用的知识和方法，从而提高用户对系统的可信程度，增加系统的透明性。另外，由于专家系统具有解释功能，系统设计者及领域专家可以方便地找出系统隐含的错误，便于对系统进行维护。

(3) 灵活性。一般专家系统的体系结构都采用了知识库与推理机相分离的构造原理，彼此既有联系又相互独立。当对知识库进行增加、删除、修改或更新时，灵活方便，对推理程序不会造成大的影响。

另外，由于知识库与推理机相分离，因此人们有可能把一个技术上成熟的专家系统变为一个专家系统工具，只要抽去知识库中的知识，就可以使它变为一个专家系统的外壳。当要建立另外一个与之功能类似的专家系统时，只要把相应的知识装入该外壳的知识库中即可。

(4) 交互性。专家系统一般采用交互式进行人机通信，这种交互性既有利于系统从专家那里获取知识，又便于用户在求解问题时输入条件或事实。也就是说，一方面，它需要与领域专家或知识工程师进行对话以获取知识；另一方面，它需要通过与用户对话来索取求解问题时所需的已知事实，以便于回答用户的询问。

(5) 实用性。专家系统是根据具体应用领域的问题开发的，针对性强，具有非常良好的实用性。

(6) 易推广。专家系统拥有大量高质量的专家知识，可使问题求解达到较高水平，再加上它具有透明性和交互性等特点，使其容易被人们接受和应用。

3.2.2 专家系统的结构和基本工作原理

1. 专家系统的结构

专家系统的结构是指专家系统各组成部分的构造方法和组织形式。系统结构选取恰当与否，是与专家系

统的适用性和有效性密切相关的。选择什么结构最为恰当,要根据系统的应用环境和所执行任务的特点而定。

不同的专家系统,其功能与结构都不尽相同,但一般都包括人机交互界面、知识库及其管理系统、推理机、解释机构、综合数据库及其管理系统和知识获取机构。专家系统的基本结构如图 3-2 所示,其中箭头方向为信息流动的方向。

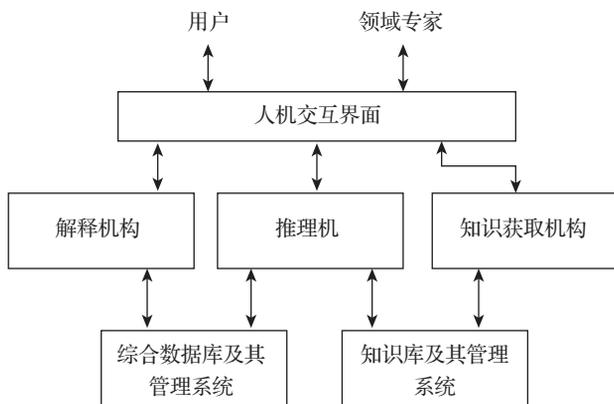


图3-2 专家系统的基本结构

(1) 人机交互界面。

人机交互界面是系统与用户进行交流时的界面。通过该界面,用户输入基本信息,回答系统提出的相关问题,并输出推理结果及相关解释等。领域专家或知识工程师通过人机交互界面输入知识,更新、完善知识库;一般用户通过人机交互界面输入预求解的问题、已知事实及向系统提出的询问;系统通过人机交互界面输出运行结果,回答用户的询问或者向用户索取进一步的事实。

在输入或输出过程中,人机交互界面需要进行内部表示形式与外部表示形式的转换。例如,输入时,它把领域专家、知识工程师或一般用户输入的信息由内部形式转换成系统的内部表示形式,然后分别交给相应的机构去处理;输出时,它把系统要输出的信息由内部形式转换为人们易理解的外部形式,显示给相应的用户。

(2) 知识库及其管理系统。

知识库是问题求解所需的领域知识的集合,包括基本事实、规则和其他有关信息。专家系统的问题求解过程是通过知识库中的知识来模拟专家的思维方式的,因此,知识库是专家系统质量是否优越的关键所在,即知识库中知识的质量和数量决定着专家系统的质量水平。一般来说,专家系统中的知识库与专家系统程序是相互独立的,用户可以通过改变、完善知识库中的知识内容来提高专家系统的性能。

知识库的建造需要知识工程师和领域专家相互合作,把领域专家头脑中的知识整理出来,并用系统的知识方法存放在知识库中。在解决问题时,用户为系统提供一些已知数据,并可从系统处获得专家水平的结论。

知识库中的知识表示形式可以是多种多样的,有产生式、框架和语义网络等,而在专家系统中运用得较为普遍的表现形式是产生式规则。产生式规则以 IF...THEN...的形式出现,就像 BASIC 等编程语言里的条件语句一样,IF 后面跟的是条件(前件),THEN 后面跟的是结论(后件),条件与结论均可以通过逻辑运算 AND、OR、NOT 进行复合。在这里,对产生式规则的理解非常简单:如果前提条件得到满足,就产生相应的动作或结论。

(3) 推理机。

推理机是实施问题求解的核心执行机构,它实际是对知识进行解释的程序,根据指示的语义,对接一定策略找到的知识进行解释执行,并把结果记录到动态库的适当空间中。推理机的程序与知识库的具体内容无关,即推理机和知识库是分离的,这是专家系统的重要特征。其优点是修改知识库时无须改动推理机,但是纯粹的推理机会降低问题求解的效率。

在这里，推理方式可以有正向推理和反向推理两种。正向推理是从前件匹配到结论；反向推理则先假设一个结论成立，看它的条件有没有得到满足。由此可见，推理机就如同专家解决问题的思维方式，知识库是通过推理机来实现其价值的。

(4) 解释机构。

解释机构用来对求解过程做出说明，并回答用户的提问。两个最基本的问题是 why 和 how。解释机制涉及程序的透明性，它让用户理解程序正在做什么和为什么这样做，向用户提供一个关于系统的认识窗口。在很多情况下，解释机制是非常重要的。为了回答 why，得到某个结论的询问，系统通常需要反向跟踪动态库中保存的推理路径，并把它翻译成用户能接受的自然语言表达式。

(5) 综合数据库及其管理系统。

综合数据库也称动态库或工作存储器，是反映当前问题求解状态的集合，用来存放系统运行过程中产生的所有信息及所需要的原始数据（包括用户输入的信息、推理的中间结果和推理过程的记录等）。综合数据中由各种事实、命题和关系组成的状态既是推理机选用知识的依据，也是解释机制获得推理路径的来源。

(6) 知识获取机构。

知识获取机构负责建立、修改和扩充知识库，是专家系统中把问题求解的各种专门知识从人类专家的头脑中或其他知识源那里转换到知识库中的一个重要机构。知识获取可以是手工的，也可以采用半自动知识获取方法或自动知识获取方法。为此，知识获取需要做知识的抽取、知识的转换、知识的输入和知识的检测 4 项工作。

知识获取是专家系统知识库是否优越的关键，也是专家系统设计的“瓶颈”问题。通过知识获取，可以扩充和修改知识库中的内容，也可以实现自动学习功能。

2. 专家系统的基本工作原理

具体来说，专家系统就是知识工程师通过知识获取手段，将领域专家解决特定问题的知识采用某种知识表示技术编辑成或自动生成某种特定表示形式，并存放于知识库中，然后用户通过人机交互接口输入信息、数据和命令，并借助数据库和运用推理机构控制知识库及整个系统工作，得到问题的求解结果。

(1) 专家系统的工作过程。

专家系统一般是通过推理机、知识库和综合数据库的交互作用来求解领域问题的，其大致过程如下：

- 1) 匹配。根据用户的问题对知识库进行搜索，寻找有关的知识。
- 2) 根据有关的知识 and 系统的控制策略形成解决问题的途径，从而构成一个假设方案集合。
- 3) 冲突解决。对假设方案集合进行排序，并挑选其中在某些准则下为最优的假设方案。
- 4) 执行。依据挑选的假设方案去求解具体问题。
- 5) 若解决方案都不能解决问题，则回溯到假设方案序列中的下一个方案，重复求解问题。
- 6) 反复执行上述过程，直到问题已经解决或可能的求解方案都不能解决问题而宣告“无解”为止。

(2) 推理机的工作过程。

推理机的工作过程如下：

- 1) 推理机将知识库中的规则前提与事实进行匹配：一般是将每条规则的前提提取出来，验证这些前提是否在数据库中。若都在，则匹配成功；否则取下一条规则进行匹配。
- 2) 把匹配成功的规则结论作为新的事实添加到综合数据库中。
- 3) 用更新后的综合数据中的事实重复上面两个步骤，直到某个事实就是意想中的结论或不再有新的事实产生为止。

3. 专家系统工作原理分析实例

下面介绍一个简单的专家系统——汽车轮胎故障诊断系统，来说明专家系统的工作原理。

如果你是汽车维修专家，非常熟悉汽车各种故障，某天你的朋友 A 先生的汽车轮胎发生故障，一时不知

道该怎么办, 他向你咨询, 你们之间有了以下对话。

你: 你的轮胎是否瘪了?

A: 是。

你: 轮胎里的空气是否快速地在泄气?

A: (经观察后) 是。

你: 你是想要修理轮胎还是要更换轮胎?

A: (思考后) 我想更换一个轮胎。

你: 你手边有什么工具吗? 如扳手、千斤顶或气泵这类工具?

A: 没有。

你: 你手边有电话吗?

A: 有。

你: 我建议你打电话呼叫公路服务来获得帮助。

在以上的对话中, 当从 A 先生那里得知汽车轮胎瘪了, 而且在快速地泄气时, 你就知道轮胎快没有气了。当你得知 A 先生想更换轮胎后, 你开始考虑让他自己修理, 或是呼叫公路服务获得帮助。于是询问他手边是否有工具, 当得到否定的答案时, 你觉得他只能依靠呼叫公路服务获得帮助了。为了进一步确认这个办法, 你又询问他是否手边有电话, 当得到肯定答复后, 你马上建议他打电话呼叫公路服务获得帮助。

我们也希望得到这样一个汽车轮胎故障诊断专家系统, 它能像你一样完成上述过程, 能与用户交互回答有关汽车轮胎故障的问题。

为了实现这一专家系统, 首先要把有关汽车轮胎故障的相关知识总结出来, 并以计算机认可的方式存放在计算机中, 这就是前面介绍的知识库。

(1) 构建知识库。

根据前面介绍的规则的知识, 结合汽车轮胎故障, 我们可以总结出如下规则组成的知识库。

Rule r1 :

If 轮胎瘪了
And 空气快速泄出
Then 确认故障: 轮胎没有气

Rule r2 :

If 轮胎变形了
And 轮胎边缘有凹入
Then 确认故障: 变形了的轮胎

Rule r3 :

If 轮胎被磨损
And 磨损严重
Then 确认故障: 磨损了的轮胎

Rule r4 :

If 变形了的轮胎
And 你要更换轮胎
Then 轮胎需要更换

Rule r5 :

If 磨损了的轮胎
And 你要更换轮胎
Then 轮胎需要更换

Rule r6 :

If 轮胎没有气
And 你要更换轮胎
Then 轮胎需要更换

Rule r7 :

If 轮胎没有气
And 你要修理轮胎
Then 轮胎需要修理

Rule r8 :

If 轮胎需要修理
And 你有一个扳手
And 你有一个千斤顶
And 你的状态良好
Then 采取措施：自己修理轮胎

Rule r9 :

If 轮胎需要更换
And 你离电话很近
Then 采取措施：呼叫公路服务获得帮助

Rule r10 :

If 轮胎需要修理
And 你要给这个轮胎充气
And 你有一个泵
And 你知道如何使用泵
Then 采取措施：自己给轮胎充气

Rule r11 :

If 轮胎需要修理
And 你知道某人会修理轮胎
And 他能够做这个事情
Then 采取措施：请求他的帮助

Rule r12 :

If 轮胎需要更换
And 你知道某人会更换轮胎
And 他能够做这个事情
Then 采取措施: 请求他的帮助

Rule r13 :

If 轮胎需要更换
And 你有一个扳手
And 你有一个千斤顶
And 你的状态良好
Then 采取措施: 自己更换轮胎

(2) 推理过程。

那推理机是如何利用知识库进行推理的呢? 由前面的知识可知, 推理机可以采用正向推理, 也可以采用逆向推理。

1) 采用正向推理。通过电话沟通已知的事实如下。

A1 : 轮胎瘪了。

A2 : 空气很快泄出。

A3 : 你要更换轮胎。

A4 : 你离电话很近。

通过正向推理获得处理故障采取的措施, 推理过程如下: 第一步, 由用户的描述“轮胎瘪了”和“空气快速泄出”, 根据规则 r1 确认故障, 得到“轮胎没有气”这个结论, 并将该结论存入动态数据库中; 第二步, 由用户回答的“你要更换轮胎”和动态数据库中的“轮胎没有气”这两个事实, 根据规则 r6, 得到“轮胎需要更换”这个结论, 并将该结论存入动态数据库中; 第三步, 由用户回答的“离电话很近”和动态数据库中的“轮胎需要更换”这两个事实, 根据规则 r9, 得到最后的结论“呼叫公路服务获得帮助”。在整个推理过程中, 每次使用规则得到新的结论都要存入动态数据库中, 并检查是否满足终止条件, 若满足终止条件, 则推理结束。正向推理过程如图 3-3 所示。

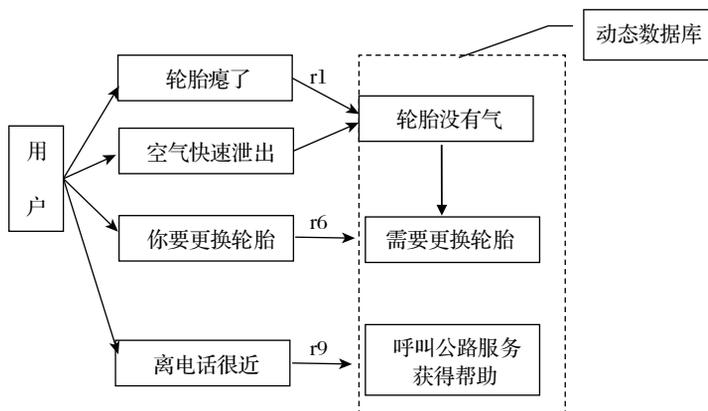


图3-3 正向推理过程

2) 采用逆向推理。逆向推理和正向推理不同, 它是从结论开始由下向上进行推理。假设系统最开始确定

的结论是“自己修理轮胎”，推理过程如图 3-3 所示。根据规则 r8，需要验证的条件是：“轮胎需要修理，你有一个扳手，你有一个千斤顶，你的状态良好”。首先需要验证“轮胎需要修理”，这时动态数据库里还没有相关信息，所以查找结论含有“轮胎需要修理”的规则 r7，其前提是“轮胎没有气”并且“你要修理轮胎”，这两个事实在动态数据库中也没有相应的信息，再查找结论含有“轮胎没有气”的规则，根据规则 r1，其前提是“轮胎瘪了”和“空气快速泄出”，可这两个事实在动态数据库中也没有相关的信息，也没有哪个规则的结论有这个结果，所以向用户提出询问是否“轮胎瘪了”和是否“空气快速泄出”，用户回答“是”，从而由规则 r1 得到“轮胎没有气”这一结论，将该结论存入动态数据库中。规则 r7 的第一个条件得到满足，再来验证第二个条件“你要修理轮胎”。同样，动态数据库中也没有相关的信息，也没有哪个规则可以得到该结论，还是要询问用户，用户回答的是“要更换轮胎”，就没有得到“你要修理轮胎”这一事实，所以规则 r7 不被满足，系统初始假定的结论“你要修理轮胎”不能成立。系统只能再次提出新的假设结论“呼叫公路服务获得帮助”，推理过程如图 3-4 所示。根据规则 r9，要验证规则的前提条件是“轮胎需要更换”并且“你离电话很近”。对于“轮胎需要更换”这一结论，有 3 个规则可以得到，分别是 r4、r5 和 r6。而其中“你要更换轮胎”这一事实已经经过电话和用户确认，得到满足；另外，动态数据库中已经有了“轮胎没有气”这一结论，所以规则 r6 的两个条件均被满足，“轮胎需要更换”这一结论已经成立，存入了动态数据库中。规则 r9 的另一个条件“你离电话很近”，需要询问用户，在得到确认后，规则 r9 得到确认。系统得出结论——呼叫公路服务获得帮助。

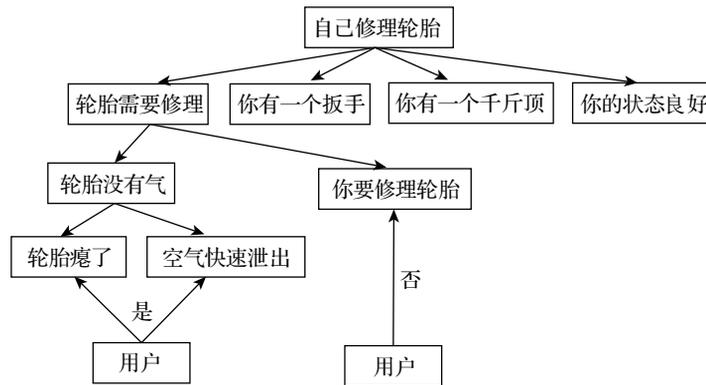


图3-3 假设“自己修理轮胎”逆向推理过程

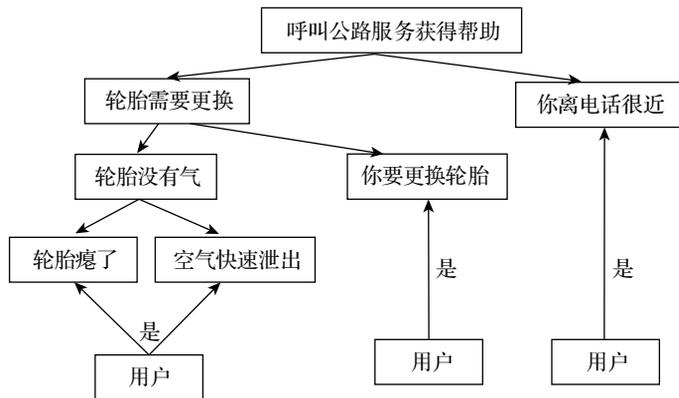


图3-4 假设“呼叫公路服务获得帮助”逆向推理过程

从以上两个推理过程可以看出, 正向推理是由上向下的推理过程, 而逆向推理是由下向上的推理过程。除了单独的正向推理和逆向推理之外, 专家系统还可以将正向推理和逆向推理结合起来, 形成双向推理, 这种推理是同时从初始事实和最终目标出发, 若在中间相遇, 则推理成功。

(3) 解释过程。

专家系统除了可以进行推理外, 还可以通过解释机构向用户进行解释, 这也是专家系统区别于传统计算机程序的重要特征。例如, 用户可能会提问为什么不可以自己修理? 解释结构可以回答: 根据规则 r8, 如果你想修理, 那么需要手边有扳手、千斤顶这些工具, 而通过和你对话, 你手边没有这些工具, 所以你不能选择自己修理。如果用户询问为什么是“呼叫公路服务获得帮助”, 解释结构可以回答: 根据你的回答, 轮胎瘪了且空气在快速泄气, 根据规则 r1, 可以确认故障是轮胎没有气; 再由你的回答, 你要更换轮胎并且你离电话很近, 则根据 r9, 系统可以得出结论“采取措施: 呼叫公路服务获得帮助”。

(4) 其他过程。

专家系统能处理所有的问题吗? 我们使用专家系统一段时间后, 会发现它并不能处理所有的问题, 有些选项系统中不存在。例如:

你: 你的轮胎是否瘪了?

A: 是。

你: 轮胎里的空气是否快速地在泄气?

A: (经观察后) 是。

你: 你是想要修理轮胎还是要更换轮胎?

A: (思考后) 我想更换一个轮胎。

你: 你手边有什么工具吗? 如扳手、千斤顶、气泵这类工具?

A: 没有。

你: 你手边有电话吗?

A: 没有。

这时就会发现, 现阶段该汽车轮胎故障诊断系统没有这个选项, 并不能处理上面这种情况。面对这种问题, 我们可以通过增加规则不断扩展并完善这个专家系统, 直到满足用户的需求。

以上给出了一个简单的专家系统以及它的工作过程, 实际的系统为了提高效率, 可能要比该系统复杂得多, 如何提出假设以使系统尽快得出答案? 如何提高匹配速度以提高系统效率? 如何处理不确定的知识和推理? 这些都是专家系统所要面临的问题。

3.2.3 专家系统的开发过程

1. 知识获取和知识工程

从人类专家处获取知识, 并将这些知识组织到可用的系统中——这个任务一直被认为是很困难的。实质上这表示了专家对问题的理解, 这对专家系统的能力至关重要。这项任务的正式名称是知识获取, 是构建专家系统面临的最大挑战。

知识获取是从领域专家处获取知识、提取知识并将其转换为专家系统程序的艰巨而细致的工作过程, 即将问题求解中领域专家的经验和技术从某个知识源提取到专家系统中。知识源包括人类专家、教科书、数据库及人本身的经验。

虽然书籍、数据库、报告或记录可以作为知识来源, 但是大多数项目最重要的来源是领域专业人员或专

家。从专家处获取知识的过程称为知识引导。知识引导可能是一项漫长而艰巨的任务，会涉及许多乏味的会话。这些会话可以是交换想法的交互式讨论，也可以是采访或案例研究的形式。在后一种形式中，人们观察专家如何试图去解决一个真正的问题。无论使用什么方法，人们的目标都是揭示专家的知识，更好地了解专家解决问题的技能。人们想知道为什么不能通过简单的问题来探索专家的知识。

自 20 世纪 70 年代以来，人们尝试了多种自动化知识获取的技术，如机器学习、数据挖掘和神经网络等。事实证明，这些方法在某些情况下很成功。例如，有一个著名的大豆作物诊断案例，在这个案例中，从植物病理学家 Jacobsen 的原始描述符集和确定诊断的患病植物的训练集开始，程序合成了诊断规则集。意想不到的是，机器合成的规则集超出了由植物病理学家 Jacobsen 博士（领域专家）制定的规则。Jacobsen 提供了原始的描述符集，然后通过部分成功实验来尝试改进他的规则，但机器的规则具有 99% 的准确性，于是他放弃了自己的努力，采用机器合成的规则作为其专业工作的基础。

专家系统的知识有以下 5 种主要的知识分类：

- (1) 过程性知识：规则、策略、议程和程序。
- (2) 陈述性知识：概念、对象和事实。
- (3) 元知识：关于其他类型的知识以及如何使用知识的知识。
- (4) 启发式知识：经验法则。
- (5) 结构化知识：规则集、概念关系和对象关系。

知识工程是通过领域专家和知识工程师之间的一系列交互来构建知识库的过程。通常随着时间的推移，随着知识工程师越来越熟悉领域专家的规则，这个过程会涉及许多规则的迭代和改进。

知识工程师一直在寻找可用于表示和解决现有问题的最佳工具，他尝试组织知识，开发推理方法和构建符号信息的技术。知识工程师与领域专家密切合作，尝试建立最好的专家系统。根据需要，重新概念化知识及其在系统中的表示。系统的人机界面得到改善，系统的“语言处理”让人类用户觉得更加舒适，系统的推理过程使用户更加容易理解。

2. 专家系统的设计

(1) 专家系统设计前的论证。

专家系统的开发设计十分复杂，而且并不是所有问题都适合用专家系统来解决，所以只有在经过论证是可行的、合理的、合适时，才能开发解决特定问题的专家系统。

1) 可行性论证。

a. 论证是否存在真正的领域专家，如果没有领域专家，那么系统没有足够的专业知识和经验进行支撑，是不能建成专家系统的。

b. 论证解决的难题需要的是经验性知识，而不是常识性知识。

2) 合理性论证。

a. 论证难题的解决能带来较高的经济效益，而且其他解决方案不适合。

b. 论证人类专家稀缺或无法亲临现场做出决策，而这些难题又十分需要专家来解决。

c. 论证人类专家的经验知识由于种种原因不断流失，而这些经验知识对于解决难题又不可或缺。

3) 合适性论证。

a. 论证问题的本质不是简单地通过算法就能解决，而是需要通过符号操作和符号结构进行求解，求解时需要经验规则才能得到答案。

b. 论证难题是具有一定复杂性的，人类需要几年的学习时间才能达到专家水平。

c. 论证求解的难题有实用价值，而且不宜过大或过小。

(2) 专家系统设计开发团队组成。

一般地,对于开发大型专家系统,开发团队需要5类人员,分别是项目经理、领域专家、知识工程师、程序员和最终用户。系统能否成功开发完全取决于成员间能否良好的沟通和协调合作。

1) 项目经理负责整个项目的统筹,与开发团队的其他4类人员进行沟通协调,安排项目的进程,保障开发的顺利进行。

2) 领域专家在特定领域有丰富的知识和经验,这些知识和经验是构成专家系统最重要的因素,因此领域专家也是最重要的团队成员。由于要将领域专家的知识和经验加入系统中,因此要求领域专家必须能够清楚地表达他们的知识和经验,而且还要善于团结协作。

3) 知识工程师负责设计、构建、测试和维护专家系统,他们要与领域专家沟通如何解决特定的问题,如何解决事实和规则,思考如何在系统中表达出来,然后选择开发框架或开发软件或编程语言。系统设计好后,还需要知识工程师对专家系统进行测试、修改以及后期维护。

4) 程序员负责设计知识库和数据库,以及推理引擎和人机交互接口。这就需要程序员掌握人工智能常用编程语言,如LISP、Prolog、OPS5,以及常用编程语言,如C、C++、Java等。但如果使用专家系统框架开发,就只需要知识工程师对知识进行编码,不需要程序员。

5) 最终用户通过使用专家系统,负责评价系统的好坏,不仅包括系统是否能够解决专业问题,还包括人机交互是否友好、操作是否便捷等。

对于大型系统,开发团队需要这5类人员;而对于微型和小型系统,有时不需要这么多的人参与,因为项目经理、知识工程师和程序员有可能是同一个人。

(3) 专家系统的设计关键。

专家系统的基本设计思想是将知识和控制推理策略分开,形成一个知识库。专家系统在控制推理策略的引导下,利用存储起来的知识分析和处理问题。这样,在解决问题时,用户为系统提供一些已知数据,然后从系统中获得专家水平的结论。

设计专家系统的关键点有两个:一是建造知识库,涉及建造知识库的两项主要技术是知识获取和知识表示;二是设计推理机制与控制策略,涉及推理机制设计的两项主要技术是基于知识规则的推理和推理解释。

1) 知识获取是建造知识库的重要基础,是专家系统开发中最关键、最艰难的一步,因此被称为专家系统开发的“瓶颈”。知识表示是指把知识客体中的知识因子与知识关联起来,便于人们识别和理解知识。

2) 基于知识规则的推理是指针对用户的特定问题选择并运用知识库中的知识,以实现求解问题的控制过程。推理涉及的两个基本问题是推理方向的选择和冲突消除。

推理方向用于确定推理的驱动方式。推理方向分为正向推理、反向推理、混合推理、双向推理以及元控制(多级控制)等。冲突消除策略解决如何在多条可用知识中合理地选择一条知识的问题,是一种基本的推理控制策略。冲突消除策略通常有以下几种:

a. 最简单的控制策略是选择某个匹配的规则。

b. 选择条件较多的规则。

c. 选择条件中含有最近生成的事实的规则。

d. 将规则的所有条件按优先级顺序进行排列,首先选择条件部分包含优先级较高条件的规则。

3) 推理解释是解释机制(解释器)的重要组成部分,其目的是对系统的推理过程、推理位置及推理的每个动作做出解释,使用户相信问题求解结论的可信性和正确性。一般设计一个专家系统的解释机制,应使其满足准确性、可理解性和智能性3个基本要求。设计原理就是基于知识和知识表达,依据规则的推理而实现的。

推理解释一般包括两部分:咨询过程中使用的推理状态检查程序和咨询中或咨询后使用的通用回答程序。

实现推理解释机制的方法有预制文本法、追踪解释法、策略解释法和自动程序员解释法等。

(4) 专家系统的设计原则。

掌握了知识获取、知识表示、知识推理和推理解释等基本技术后,即可着手进行专家系统的设计。专家系统设计与基于算法的传统程序设计有一定的区别:专家系统设计一般是渐增式的,通过知识库由小到大地逐步扩充和改进,要求系统不断地进行验证、评价和专家认可,最终才能成为可交付使用的专家系统。所以,大多数专家系统的设计都遵循如下3个原则:

1) 知识与知识机构分开和相互独立的原则。该原则使得系统有很好的模块性、可扩充性和可维护性。

2) 按系统功能实现模块化构造的原则。为了使结构清晰、调试容易,绝大多数专家系统都采用按系统功能实现模块化构造的原则把系统分成几个相互独立的功能模块。

3) 交互性原则。领域专家和用户与专家系统信息交换的人机接口、知识工程师维护知识库等都需要与系统具有良好的交互性操作,这使得领域专家和用户都以尽可能自然、易于实现的方式进行信息传输和结果输出,并使知识工程师对于知识矛盾、冗余检查、新知识加入对现有知识的影响进行调节,以及知识的存储、共享等进行交互处理,还要求能处理一些意想不到的“怪现象”,专家系统应有“不耻下问,善于求教”的交互功能,以及十分友好的用户接口等。

(5) 专家系统的设计步骤。

1) 设计初始知识库。知识库的设计是建立专家系统最重要和最艰巨的任务。初始知识库的设计大致包括以下5个部分:

a. 问题知识化,即辨别所研究问题的实质,如要解决的任务是什么、它是如何定义的、可否把它分解为子问题或子任务、它包含哪些典型数据等。

b. 知识概念化,即概括知识表示所需要的关键概念及其关系,如数据类型、已知条件(状态)和目标(状态)、提出的假设及控制策略等。

c. 概念形式化,即确定用来组织知识的数据结构形式,应用人工智能中的各种知识表示方法把与概念化过程有关的关键概念、子问题及信息流特性等变换为比较正式的表达,包括假设空间、过程模型和数据特性等。

d. 形式规则化,即编制规则,把形式化了的知识变换为由编程语言表示的可供计算机执行的语句和程序。

e. 规则合法化,即确认规则化了知识的合理性,检验规则的有效性。

2) 原型机(Prototype)的开发与试验。在选定知识表达方法后,即可着手建立整个系统所需要的试验子集,包括整个模型的典型知识,只涉及与试验有关的足够简单的任务和推理过程。选择合适的专家系统程序设计语言和开发工具设计推理机制,或借用工具语言已具备的推理机制将形式化表示的知识以专家系统求解目标或图解形式的模块为单位,逐个单元地把知识转换为适合程序设计语言或工具能接受的内部编码形式,输入知识库。在不断供给知识库新知识的同时,系统要不断地对已有知识和新加入知识的正确性及协调性通过实例进行测试,通过不断扩充知识库和不断测试的过程,一般可以发现已形式化的知识有许多不完善处,从而可以在领域专家的配合下不断地对系统进行调整。

3) 知识库的维护(知识库的改进与归纳)。当开发出原型专家系统后,让领域专家选择一些有代表性的实验例子。在可能的情况下,让领域专家利用这些实验例子进行实际问题的求解。根据实验例子运行可能出现的新问题,反复对知识库(推理规则)进行改进试验,以便得出更完善的结果。经过长时间(如数月甚至两三年)的努力,专家系统在一定范围内能达到人类专家的水平。专家系统的设计步骤如图3-5所示。

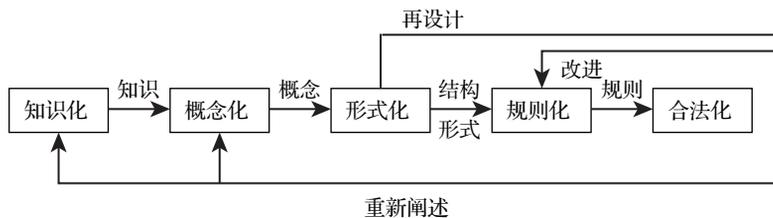


图3-5 专家系统的设计步骤

3. 专家系统设计开发工具

由于专家系统的应用领域非常广泛,而每个具体的专家系统都重新设计,这必然降低开发速度,因此人们研制出一些通用工具,作为辅助手段来提高开发效率。这些辅助手段被称为专家系统开发工具,其主要包括骨架开发工具、通用语言开发工具和辅助构造开发工具3种。

(1) 骨架开发工具。专家系统都包括推理机和知识库,这两者在结构上是独立的。骨架开发工具是将已经成功运行的专家系统中的知识库部分去掉,保留推理机部分。使用这种骨架开发工具时,只需将其他专业领域的知识代替被去掉的知识库即可,这样就大大加快了开发进度。常见的骨架开发工具有EMYCIN、KAS和EXPERT等。

(2) 通用语言开发工具。通用语言开发工具与骨架开发工具不同,它不提供具体的体系和范例,也不提供具体问题的求解策略和表示方法,而是提供建立专家系统的基本机制,控制策略也不固定于某种形式。因此,通用语言开发工具结构灵活,应用范围更广。常见的通用语言开发工具有OPS5、CLIPS、OPS83、RLL和ROSIE等。

(3) 辅助构造开发工具。辅助构造开发工具由一些程序模块组成,有些用于辅助获得专业知识,有些用于构造专家系统的结构。其主要分成两类:一类是设计辅助工具;另一类是知识获取辅助工具。常见的辅助构造开发工具有AGE、TIMM和TEIRESIAS等。

4. 专家系统的评价

在专家系统完成后,应对系统的各个方面做出适时的评价。专家评价是贯穿于专家系统整个建造过程的一项重要工作,从需求分析开始直到完成都要反复多次地进行评价,以便及时发现问题、及时修正,不致走太多弯路。

关于如何评价一个专家系统,目前尚无统一标准,但从建造专家系统的设计目标、结构和性能等方面来看,一般可从以下几个方面进行评价。

(1) 知识的完备性。

知识的完备性可从以下3个方面来考察:

1) 是否具有完善的知识,即专家系统是否具有求解领域问题的全部知识,包括领域知识及求解问题时运用知识的知识(元知识)。

2) 专家系统的知识是否与领域专家的知识保持一致,即是否正确理解了领域专家的知识。

3) 知识是否一致、完整,即是否存在冗余、矛盾和环路问题等。

(2) 表示方法及组织方法的适当性。

对于表示方法及组织方法的适当性,可从以下4个方面进行考察:

1) 能否充分表达领域知识,尤其是对不确定知识的表示是否准确、合理。

2) 是否有利于对知识的利用,是否有利于提高搜索及推理效率。

3) 若问题领域要求用多种模式表示知识,则其表示方法与组织方法是否便于对这种知识进行表示和组织。

4) 是否便于对知识进行维护与管理。

(3) 系统所做的决定和建议的质量。

在对发展起来的专家系统（包括 MYCIN 和 PROSPECTOR 等）做评价研究时，倾向于把重点放在评价这些系统完成决策任务时的程序性能。因为可靠而准确的建议是专家咨询系统的一个关键成分，通常这是一个不但有重大研究价值，而且有重大实用价值的领域，从而理所当然地成为要着重评价的领域。但是，要定义或论证一个系统的建议是否合适或充分的机理可能是困难的。专家系统往往是为需要起裁判作用和非标准化决定的领域而建立的。但有一点是很清楚的，那就是专家系统如果不能说服别人相信这些决定和建议是恰当的与可靠的，那么使用者就不会接受这个专家系统。通常某些进行校核的方法是强制性的。

(4) 所用推理技术的正确性。

只要专家系统能提供适当的建议，那么并非所有的专家系统设计者都关心他们的程序是否是以和人类思维类似的方法得出结论的。对 MYCIN 系统和 PROSPECTOR 系统的评价都不考虑系统用来得到结论的推理方法是否和专家所用的可相比较的方法等价。

现在人们日益认识到，要达到专家水平的性能，可能要求更加重视推理机理，其是专家用来解决那些通常要求专家系统去解决的问题时所应用的。

(5) 人和计算机之间对话的质量。

虽然专家系统推理过程的可靠性是系统最后能否被成功使用的根本因素，但现在知识工程师通常承认各种其他因素也会影响系统是否被预订的使用者接受。专家系统和使用者之间能很自然的对话尤其重要。与此相关的问题如下：

1) 在提问和由程序来产生解答时用词的选择。

2) 专家系统解释它如何做出决策的基本能力，以及使系统的解释适合于使用者专门知识水平的能力。

3) 当使用者对系统要求他们做的事情疑惑不解时，或在使用程序时由于某种原因需要帮助时，专家系统对使用者提供帮助的能力。

4) 专家系统以容易理解的方式或以使用者熟悉的术语来提出建议或向使用者进行解释。

对于一个专家系统最后能否被成功使用来说，上述问题中的每一点都与系统提供建议的质量同样重要，所以有必要正式地对这些方面进行评价。现在许多专家系统已经做出很多努力来发展这方面的能力，但是评价这些能力效果的技术以及在研究设计中把一个因素和其他因素分开出的技术都还很不成熟。因为在一个专家系统被实际使用之前，上述这些问题通常不适于被正式研究。

(6) 系统的效率。

系统的效率是指系统运行时对系统资源的利用率及时空开销。在评价系统的过程中，必须分析在实际使用环境下专家系统对决策过程的影响。例如，一个系统如果要求使用花费过多的时间，即使它在完成所有上面提到的任务方面是很出色的，也难以被使用者接受。类似地，系统运行的技术分析一般也是必要的。例如，CPU 的能力没有充分发挥，或磁盘寻找过程设计不善，可能会造成系统效率不高，这就严重地限制了系统的反应时间或成本效果。

(7) 系统的可维护性。

可维护性是指系统是否便于检测、修改和扩充，对于知识库更是如此。

(8) 解释能力。

解释是让用户了解系统，增强对系统信任度的有力手段，也是帮助系统进行调试的辅助工具。

(9) 成本效果（或工程经济分析）。

如果希望一个专家系统能够成为市场上的产品，那么就有必要对系统的成本效果做某些详细的评价。当然，最后要由市场来对产品的成本效果做出评判，市场将帮助确定程序所提供的方法是否成功。

3.2.4 专家系统举例

近40多年来,人们建成了具有数以千计规则的专家系统。本小节将探讨一些著名的系统,并介绍它们的背景、历史、主要特征和主要成就。

1. DENDRAL

作为专家系统发展的一个例子, DENDRAL 几乎与人工智能的历史一样悠久,而且它的地位举足轻重。从各个角度来看, DENDRAL 代表了一个成功的故事,该项目开始于1965年,持续多年,涉及斯坦福大学的许多化学家和计算机科学家。无论是在试验意义上,还是在正式的科学意义上,许多与人工智能发展有关的想法都是从这个项目开始的。例如,在早期, DENDRAL 强有力地证明了生成和测试 (Generate-and-Test) 算法以及基于规则的方法能够有效地建立专家系统。

该系统的主要开发人员是爱德华·费根鲍姆 (Edward Feigenbaum, 计算机科学家)、约书亚·莱德伯格 (Joshua Lederberg, 化学家、遗传学诺贝尔奖获得者)、布鲁斯·布坎南 (Bruce Buchanan, 计算机科学家) 和雷蒙德·卡哈特 (Raymond Carhart, 化学家), 他们都在斯坦福大学工作。

DENDRAL 的任务是列举合理的有机分子化学结构 (原子键图), 输入两种信息: 一种是分析仪器质谱仪和核磁共振光谱仪的数据; 另一种是用户提供的答案约束, 这些约束可从用户可用的任何其他的知识源 (工具或上下文) 推导得到。

正如费德鲍姆所说, 过去还没有将未知化合物的质谱图映射到其分子结构的算法。因此, DENDRAL 的任务是将人类专家莱德伯格的经验、技能和专业知识纳入程序中, 这样程序就可以以人类专家的水平运行。在开发 DENDRAL 的过程中, 莱德伯格不得不学习很多关于计算的知识, 正如费德鲍姆不得不学习化学知识一样。显然, 对于费德鲍姆而言, 除了与化学有关的许多具体规则外, 化学家还根据经验和猜想使用了大量启发式知识。

DENDRAL 的输入通常包含所研究的如下化合物信息: 化学式, 如 $C_6H_{12}O$; 未知有机化合物的质谱图和核磁共振光谱信息。无须反馈, DENDRAL 在3个阶段执行启发式搜索, 这称为规划-生成-测试 (Plan-Generate-Test)。

(1) 规划。在该阶段, 根据所有可能的原子构型的集合中和质谱推导出的约束一致的原子构型集合, 还原出答案。应用约束, 选择必须出现在最终结构中的分子片段, 剔除不能出现的分子片段。

(2) 生成。使用名为 CONGEN 的程序来生成可能的结构, 它的基础是组合算法 (具有数学证明的完整性以及非冗余生成性)。组合算法可以产生所有在拓扑上合法的候选结构。通过使用规划过程提供的约束进行裁剪, 引导生成合理的集合 (满足约束条件的集合), 而不是巨大的合法集合。

(3) 测试。在最后阶段, 根据假想中的质谱结构与试验结果之间的匹配程度对生成的输出结构排列次序。DENDRAL 可以很迅速地将数百种可能的结构缩减到可能的几种或一种结构。如果生成了几种可能的结构, 那么系统将会列出这些结构并附上概率。DENDRAL 证明了计算机可以在一个有限的领域内表现得与人类专家相当。在化学领域, 它的表现高于或等于一个化学博士生。程序主要使用 Interlisp (LISP 语言的一个分支) 来写, 如 CONGEN 这样的子程序使用 Fortran 和 S 人工智能 I 语言编写。在美国, 该系统很有市场, 它在化学领域中得到了广泛的应用。费德鲍姆进一步指出, DENDRAL 的结构阐释能力既非常广泛, 也非常狭窄, 这很矛盾。一般来说, DENDRAL 能够处理所有分子、环和树状。在约束条件下 (纯粹的仪器数据), 对纯结构的阐释, CONGEN 的表现是人类无法匹及的, 在这些知识密集型的专业领域, 通常来说, 比起人类专家的表现, DENDRAL 的表现不但快, 而且准确。

2. MYCIN

毫无疑问，人们引用最多的最著名的专家系统是 MYCIN。它也是在斯坦福大学开始开发的，并且是爱德华·肖特利夫 (Edward Shortliffe) 的博士论文项目。这个基于规则的专家系统主要针对由血液和脑膜炎 (细菌性疾病，引起脑和脊髓周围膜的炎症) 引起的感染性血液病给出诊断和治疗建议。如果这些疾病不及早治疗，将是致命性的。开发 MYCIN 需要大约 20 年，该系统使用反向链接 (Backward Chaining)，并且由 400 多条规则组成。与 DENDRAL 一样，它主要是用 Interlisp 编写的。

显然，由于可能的疾病对生命有威胁，因此对出现的特定感染快速诊断，并快速地确定适当的药物进行干预这一过程很重要。人们需要这种系统，这与 20 世纪 70 年代人工智能发展的方向是一致的。

此外，如果系统开发成功了，为了使人们接受系统，那么系统必须是互动式的，这与医生和常驻血液感染专家之间的合作类似。系统应该能够回答医生的问题，并且要能够适应 (而不是消除或阻碍) 医生的需求。

MYCIN 是人们描写最多的、研究最多的、最典范的程序。例如，杜尔金 (Durkin) 用一整章来描写 MYCIN，并为系统的背景、方法、性能和评估提供了一些非常有趣的见解。他指出，20 世纪 70 年代，治疗程序导致了抗生素的滥用。他注意到了罗伯茨 (Roberts) 和维斯康西 (Visconti) 的研究，这些研究暗示医生选择的 66% 的治疗方法是不合适的，在这些方法中，超过 62% 的方法使用了不当的抗生素组合。

他注意到青霉素的发现导致了大量抗生素的引进。这些药物虽然在处方适当并且正确使用时无效，但是也可能产生毒副作用。标准的案头参考 *The Physician's Desk Reference* 是必需的，也是有帮助的。此外，血液病领域的专业知识比较匮乏。因此，据 Durkin 介绍，由于上述原因，开发 MYCIN 程序似乎是个好主意。

为了证明并解释结论，MYCIN 通过规则集、反向链接搜索确凿的证据。通过使用反向链接，MYCIN 可以与人类专家相媲美。它证实了系统的专家能力来自知识，而不是一些聪明的技巧。

基于 Durkin 的介绍，MYCIN 代表的重要成就如下：

(1) 知识与控制分离。这是所有专家系统的标志，MYCIN 是较早的例子之一。如果规则更改，人们就不必更改推理引擎。

(2) 集成元规则 (规则的规则)。何时因特殊案例或情况打破规则。

(3) 采用不精确推理。MYCIN 的确定因子 (CF) 范围为 -1 (绝对为假) $\sim +1$ (绝对为真)。另一个术语是“可信度”。

(4) 记住先前会话，如一个优秀的人类专家在领域中的每一个经验都会对专家知识产生影响。

(5) 适应用户。程序必须让用户感到舒服，对用户或医师透明。

(6) 自然语言交互。MYCIN 以自然语言的方式与医生互动。

(7) 提供解释。MYCIN 可以解释它得出特别结论的方式和原因。

(8) 提供可替代建议。MYCIN 试图为医生提供可替代选项。这样，程序可以显得是合作型的，而不是指导型的，因此医生可以感觉到系统是在帮助他，而不是在控制或命令他。

3. EMYCIN

实践证明，MYCIN 是一个成功的专家系统，因此人们决定推广它。William van Melle 使用 MYCIN 推理引擎和一本 1975 年庞蒂亚克服务手册，构建了一个用于诊断汽车喇叭电路问题的 15 规则系统。该玩具系统为第一个专家系统命令解释器 EMYCIN 的开发奠定了基础。约书亚·莱德伯格 (Joshua Lederberg) 建议的首字母缩略词 EMYCIN 表明这是“基本 (Essential)”或“空 (Empty)”的 MYCIN。命令解释器具有特定目的，设计它是用于特定类型的应用程序，在这些应用程序中，用户只需提供知识库。通过删除 MYCIN 专家系统中的医学知识库，我们就可以得到 EMYCIN 命令解释器。

很自然地，这个目标是为了保留 MYCIN 的优秀特征。这些特征包括特定领域知识的表示、遍历知识库

的能力、支持不确定性的能力、假设推理和解释工具等。

EMYCIN 支持正向链接和反向链接,并引导了许多专家系统的开发,包括一个诊断肺部问题的应用程序 PUFF。专家系统技术提供了一种工具,专家系统可以使用该工具经济、有效地建立起来。EMYCIN 符合了 Donald Michie 所列出的成功专家系统的要求,所以其对专家系统技术而言是一个非常重要的发展。EMYCIN 成为所有未来专家系统命令解释器的典范。

3.2.5 新型专家系统

随着专家系统的应用日益广泛,专家系统较人类专家的优越性突显,但由于处理问题的难度和复杂程度也不断增大,使得专家系统的一些劣势也逐渐显现,具体体现在以下几个方面:

(1) 知识获取困难。专家系统的核心是知识与推理,虽然有许多因素都制约专家系统的性能,但首先起制约作用的是专家系统内存储的知识数量和质量。而构建一个专家系统,需要知识工程师从大量资料中获取该领域的基础知识,并且从该领域的专家中获取个性化知识,这些知识的获取都是很困难的。

(2) 工作效率不高。专家系统不是依据启发式方法处理问题的,而是依靠“匹配—冲突消解—操作”的重复周期来实现问题求解的,处理问题的工作效率不高。

(3) 处理复杂问题困难。在处理复杂问题时,由于求解问题状态空间大,操作数目多,使得知识的匹配以及操作序列会大大增多,导致处理复杂问题十分困难。

(4) 处理不确定性问题能力差。经典的专家系统实际上是一个二进制逻辑系统,对于处理确定性的数据和命题十分可靠,但对于在实际工作中经常出现的模糊的不确定的系统,处理结果很不理想。

(5) 系统适应能力较差。经典专家系统无法在系统运转过程中不断发展完善,没有自我学习能力,系统功能取决于设计者的能力和知识。

由于上述原因,基于知识的经典专家系统开发和应用已经逐渐减少。近年来,在计算机技术快速发展的情况下,涌现出一些新型实用的专家系统。

这些新型实用的专家系统具有以下特点:能够进行并行分布式处理,提升了处理效率;为了拓宽解决问题的领域,增加了多专家协同工作;增强了自主学习能力;增加了多种知识推理机制,包括模糊推理和不完备知识推理等;拥有更多的智能人机接口。

1. 模糊专家系统

模糊专家系统是采用模糊计算技术来处理不确定性问题的新型专家系统。模糊专家系统中的知识是特定领域的具有复杂性和模糊性的专门知识。模糊专家系统能够对以模糊形式表示的知识进行模糊推理,能够采用模糊逻辑推理、知识不完全推理和不确定性推理来解决各种问题。

模糊专家系统的结构与其他专家系统的结构基本相同,一般由模糊知识库、模糊数据库、模糊推理机、模糊知识获取模块、解释模块和人机接口 6 部分组成。

(1) 模糊知识库用于存放解决特定问题的事实与规则。

(2) 模糊数据库用于存放已知的模糊证据和推理过程中得到的具有模糊性的中间结论。

(3) 模糊推理机依据初始的模糊信息,利用模糊知识,按照模糊推理方法,得到可以接受的模糊结论。主要的模糊推理方法包括模糊关系合成推理和模糊匹配推理。

(4) 模糊知识获取模块用于把专家的专业领域知识转化成模糊知识的形式存于模糊知识库中。

(5) 解释模块用于回答用户的提问,给出模糊推理过程和结果。

(6) 人机接口是系统与外界进行信息交流的通道,并且交流的信息都是模糊信息。

2. 分布式专家系统

分布式专家系统能够进行并行分布式处理，简单来说，就是把一个专家系统的功能分布到多个处理机上去执行，其可以是在一个计算机上的紧耦合形式的多处理器系统，也可以是在众多计算机上的松耦合形式的计算机网络系统，这样大大提高了系统处理问题的效率。

分布式专家系统分布的特性决定了其需要解决一些有别于其他专家系统的特殊问题。

(1) 功能分布。需要把系统功能分解成多个子功能，再分布到各个处理节点上，每个节点接收一部分功能，各个节点组合在一起构成全部功能完成系统任务。但节点不宜过少，也不宜过多。节点过少会使系统并行性能降低；过多则使节点间信息交换时间过长，影响处理效率。

(2) 知识分布。要合理地将知识分布在各个节点上，既要考虑使用知识的便利性，又要考虑系统的可靠性，所以在每个节点上的知识需要一些冗余，但也不能有太多的冗余。

(3) 接口设计。其接口设计既要易于各个节点之间通信和同步，也要保持各个节点的相对独立。

(4) 系统结构。整个分布式系统的结构既与所处理的问题有关，也与硬件环境有关，一般采用树形结构和星形结构。

(5) 驱动方式。当系统结构确定后，就需要设计各个模块之间的驱动方式，一般采用数据驱动、需求驱动、事件驱动和控制驱动等。



任务 3.3 本体研究：如何正确表示知识

知识是智能的基础，人类的一切智能行为归根结底就是一个不断获取知识和应用知识的过程。同样地，要使计算机具有智能，使之能够模拟和延伸人类的智能行为，就必须使它具有知识。而应用知识的前提是知识的表示方式，只有能够被计算机识别、存储和处理的知识表示方式，才能驱动计算机进行工作。因此，基于知识的表示问题一直以来都是人工智能领域研究的一个重要课题。

3.3.1 知识的概念

知识是什么？这是一个既简单又难以给出严格、统一定义的概念。在日常生活中，人们几乎每天都会涉及“知识”这一术语。例如，“知识改变命运”“知识就是力量”“从这件事情中我们学到哪些知识”“把我们学到的知识应用于社会实践当中”等。但究竟什么才是知识，它和信息及数据之间又具有怎样的联系呢？

事实上，人们所生活的环境就是一个由大量信息组成的信息世界，而且每时每刻人们都在应用这些信息并产生大量新的信息。可以说，信息是推动社会发展和进步最根本的原动力，谁掌握的信息多就意味着谁能占得所在领域的先机。但就信息本身而言，它并不能被人们直接使用，尤其是对于计算机而言，需要以一定的形式表示出来，才能被人们识别、记忆和传递。数据是信息的载体，是形式化了的信息，是用一组符号及其组合表示的信息。因此，信息与数据是两个紧密相关的概念，它们之间既有联系又有区别。相同的数据在不同的信息场合所代表的含义是不同的。例如，“0”和“1”这一对数据可以是二进制信息中的两种基本符号，可以表示电压信息的两种状态，也可以用来表示判断结果的两种状态（真和假）。

不难理解，单一的信息不足以表达一个完整的、复杂的问题，只有将相关的信息按一定规则关联在一起

才具有实际的意义。因此,把相关信息组合在一起所形成的信息结构称为知识,这也是人工智能研究的起点。目前,公认的比较有代表性的关于知识的定义有以下几种:

- (1) 费根鲍姆认为知识是经过裁剪、塑造、解释、选择和转换了的信息。
- (2) 伯恩斯坦 (Bernstein) 认为知识是由特定领域的描述、关系和过程组成的。
- (3) 海叶斯 - 罗斯 (Heyes- Roth) 认为知识 = 事实 + 信念 + 启发式。

知识反映的是客观世界中同类事物或不同类事物之间的相互关联,存在事实和规则两种基本的类型。其中,事实陈述的是一种状态或一种现象,它反映的是“……是……”,如“当雪花是白色的”“今天不下雨”“老虎是食肉动物”“太阳从东边升起,从西边落下”等;规则反映的是两个事物间的因果关系,讲的是“如果……,那么……”。例如,“如果某动物有羽毛,那么它是鸟类”“如果某动物有奶,那么它是哺乳动物”“如果大雁往南飞,那么冬天就要来了”等。后面所要学习的产生式系统就是以规则作为基本知识表示的。

3.3.2 知识的分类和特征

1. 知识的分类

根据研究和观察的角度不同,知识有多种不同的分类方法。下面介绍几种比较常见的知识分类方法。

(1) 就知识的作用范围而言,知识可分为常识性知识和领域性知识。常识性知识是通用性知识,是人们普遍知道的知识,适用于所有领域;领域性知识是面向某个具体领域的知识,是专业性知识,只有相应专业的人员才能掌握,并用来求解领域内的相关问题。

(2) 就知识的层次性而言,知识可分为表层知识和深层知识。

(3) 就知识的确定性而言,知识可分为确定性知识和不确定性知识。确定性知识是指可指出其真值为“真”或“假”的知识,是精确的知识;不确定性知识是指具有“不确定”特性的知识,是不精确、不完全及模糊性知识的总称。

(4) 就人类的思维及认识方法而言,知识可分为逻辑性知识和形象性知识。逻辑性知识是反映人类思维过程的知识,这种知识一般都具有因果联系性及难以精确描述的特点;而形象性知识是通过形象思维获得的知识,如用语言描述物体不如用图片展示。

(5) 就知识的等级而言,知识可分为元知识和非元知识。

(6) 就知识的作用及表示而言,知识可分为事实性知识、过程性知识和控制性知识。

1) 事实性知识用来描述系统的状态、环境和条件,以及问题的概念、定义和事实。事实性知识一般采用直接表达的形式,如用谓词公式表示等。

2) 过程性知识主要是指与领域相关的知识,由规则、定理等构成,一般是通过对领域内各种问题的比较与分析得出的规律性的知识。

3) 控制性知识又称深层知识或元知识,是关于如何运用已有的知识进行问题求解的知识,因此又称关于知识的知识。例如,问题求解中的推理策略、信息传播策略、搜索策略和求解策略等。

2. 知识的特征

知识反映的是客观世界中事物之间的相互关系,具有相对正确性、不确定性、可表示性和可利用性。

(1) 相对正确性。知识的正确性是在“一定条件或确定环境”下才成立的,具有相对性,即想要确定知识的正确性必须在一定的条件或具体的环境下才有意义,且脱离具体的环境或条件,正确性的知识就会变成错误的知识。例如,“在同一平面上,永不相交的两条直线称为平行线”“ $1+2=3$,这种表示只有在十进制下

才是正确的”。所以，知识具有相对正确性，必须在实践中不断检验。

(2) 不确定性。由于知识反映的是事物之间的相互关系，而世界上的事物之间的关系本就是一个极其复杂的系统，因此很多情况下很难用“真”和“假”两种状态明确地表示事物之间的关系。不确定性表示的就是这种介于“真”和“假”之间的中间状态，它具有随机性、模糊性、经验性和不完全性。

(3) 可表示性。知识要想被人们识别、理解、存储和传递，就必须能够以一定的方式（如语言、文字、图形、图像、声音和公式等）呈现出来。因此，人们所能够正确使用的知识都是形式化以后得到的信息。

(4) 可利用性。不管是用什么样的表示方式呈现知识，最终都应该能够被人们利用。如果知识不具有可利用性，那么它也就没有存在的意义了。人们之所以能够改变世界，就是因为人们在不断地发现知识和利用知识。

3.3.3 知识表示

在人工智能领域中，知识表示实际上就是对知识的一种描述，或者说是一组约定，即一种能够被计算机接受的用于描述知识的数据结构。对知识表示的过程就是把知识编码成某种数据结构的过程。

知识表示是人工智能所有研究工作的起点，对知识表示方法的研究离不开人们对于知识的认知和理解的程度。

目前由于对于人类知识的结构及认知机制还没有完全认识清楚，因此人们对知识表示的理论及规范还没有建立起一个完整的体系。

1. 知识表示方法的类型

根据研究的内容及侧重点不同，人们逐渐形成了一系列的知识表示方法，归纳起来，主要分为两大类，即符号表示法和连接机制表示法。

(1) 符号表示法主要用来表示逻辑性知识，即用各种包含具体含义的符号，以不同的组织方式和组织次序连接起来的知识表示方法。

(2) 连接机制表示法是用神经网络技术表示知识的一种方法，它把各种物理对象以不同的方式及次序连接起来，并在其间互相传递及加工各种包含具体意义的信息，以此来表示各种相关的概念和知识。

2. 知识表示方法的要求

人工智能中对于知识表示方法的要求可以概括为以下两个方面。

(1) 知识表达性。好的知识表示方式必须具有较强的表达能力和足够的精细程度，用相对较少的符号及规则实现对复杂问题的描述。其具体可以从以下 3 个方面考虑：

- 1) 表示能力，要求能够正确、有效地将问题求解所需的各类知识表达出来。
- 2) 可理解性，所表示的知识应易懂、易读且易于表示。
- 3) 自然性，表示方式要自然，适用于不同的环境和用途，易于检查、修改和维护。

(2) 知识利用性。从知识的可利用性角度考虑，衡量知识表示方法可从以下 3 个方面考虑：

- 1) 便于获取和表示新知识，并以合适的方式与已有的知识相连接。
- 2) 便于搜索，在求解问题时，能够较快地在知识库中找到有关的知识。
- 3) 便于推理，要能够从已有知识中推出需要的答案或结论。

任务 3.4 如何表示知识

知识是需要表示的,为表示方便,一般采用形式化的,并且具有规范化的表示方法,这就是知识表示。在人类智能中,知识蕴藏于人脑中,但在人工智能中需要用知识表示的方式将知识表示出来,以便对它进行讨论与研究。知识表示就是用形式化、规范化的方式对知识进行描述,其内容包括一组事实、规则以及控制性知识等,部分情况下还会组成知识模型。

3.4.1 产生式表示法

1943年,由Post提出产生式表示法,其使用类似于文法的规则,对符号串进行替换运算。产生式表示法可用以模拟人类求解问题时的思维过程。

产生式表示法是人工智能中最常见、最简单的一种表示法。当给定的问题要用产生式系统求解时,要求能掌握建立产生式系统形式化描述的方法,所提出的描述体系具有一般性。

1. 产生式表示法的知识组成

产生式表示法的知识由事实和规则组成,它也可表示部分元知识。

(1) 事实表示。产生式表示法中的事实表示有性质和关系两种表示。

1) 对象性质表示。对象性质可用一个三元组表示,即

(对象, 属性, 值)

它表示指定对象具有指定性质的某个指定值,如(牡丹花, 颜色, 红)表示牡丹花是红色的。

2) 对象间关系表示。对象间关系可用一个三元组表示,即

(关系, 对象1, 对象2)

它表示指定两个对象间具有指定的某个关系,如(父子, 王龙, 王晨)表示王龙与王晨是父子关系。

一个给定问题的产生式系统可组成一个事实集合体,称为综合数据库。

(2) 规则表示。规则是事实间因果联系的动态表示。产生式规则的一般形式为

If P then Q

其中,前半部P确定了该规则可应用的先决条件,后半部Q描述了应用这条规则所采取行动得出的结论。一条产生式规则满足了应用的先决条件P后,就可用规则进行操作,使其发生变化,并产生结果Q。

一个给定的问题的产生式系统可组成一个规则集合体,称为规则库。

2. 产生式表示法与知识

第一层:产生式表示法中的对象,它给出了知识中的对象。

第二层:产生式表示法中的事实,它给出了知识中的事实。

第三层:产生式表示法中的操作,它给出了知识中的规则。

第四层:产生式表示法中的知识可设置约束,它给出了元知识。

3. 产生式表示法实例

在医学专家系统中判定咽炎的产生式表示。

在该医学专家系统中有事实：

- A：(病人，咽部观察，充血)；
- B：(病人，咽部主诉，疼痛)；
- C：(病人，白细胞数，高)；
- D：(病人，中性指标，高)；
- E：(病人，体温，高)；
- F：(病人、病症，急性咽炎)；
- G：(病人，白细胞数，>500)；
- H：(病人，中性指标，>60)；
- I：(病人，体温，>17)；
- J：(病人，白细胞数，≤ 500)；
- K：(病人，中性指标，≤ 60)；
- L：(病人，体温，≤ 60)；
- M：(病人，白细胞数，正常)；
- N：(病人，中性指标，正常)；
- P：(病人，体温，正常)；
- Q：(病人，病症，慢性咽炎)。

上面 16 个事实组成了综合数据库。它的产生式规则如下：

- If A and B and C and D and E then F ；
- If G then C ；
- If H then D ；
- If I then E ；
- If J then M ；
- If K then N ；
- If L then P ；
- If A and B and M and N and P then Q。

上面 8 个规则组成了规则库。

4. 产生式表示法的评价

产生式表示法是目前人工智能中最常见的一种表示法，它在表示上有以下优点：

(1) 知识表示的完整性。可以用产生式表示法表示知识体系中全部 4 部分：可以用产生式表示法中的对象表示知识中的对象，可以用产生式表示法中的事实表示知识中的事实，可以用产生式表示法中的规则表示知识中的规则，可以用产生式表示法表示知识中的部分元知识。此外，用产生式表示法表示的知识以确定性知识为主，但一定程度上也可以表示非确定性知识。

(2) 表示规则简单易用。用产生式表示法表示知识时，无论是对象、事实还是规则都很简单，因此易于掌握使用。



知识表示法

3.4.2 语义网络表示法

1968 年，J.R.Quillian 在研究人类联想记忆时提出了一种心理学模型——语义网络，他认为记忆是由概念

间的联系实现的。随后在他设计的可教式语言理解器 (Teachable Language Comprehendent) 中,又把它用作知识表示方法。1972年,西蒙(Simon)在他的自然语言理解系统中也采用了语义网络知识表示法。1975年,亨德里克(G.G Hendrix)又对全称量词的表示提出了语义网络分区技术。目前,语义网络已经成为人工智能中应用较多的一种知识表示方法,尤其是在自然语言处理方面的应用。

1. 语义网络的概念及结构

语义网络是一种通过概念及其语义联系(或语义关系)来表示知识的有向图,节点和弧必须带有标注。其中,有向图的各节点用来表示各种事物、概念、情况、属性、状态、事件和动作等,节点上的标注用来区分各节点所表示的不同对象,每个节点可以带有多个属性,以表征其所代表的对象的特性。

在语义网络中,节点还可以是一个语义子网络。弧是有方向的、有标注的,方向表示节点间的主次关系,且方向不能随意调换。标注用来表示各种语义联系,指明它所连接的节点间的某种语义关系。

从结构上来看,语义网络一般由一些基本的语义单元组成。这些基本的语义单元称为语义基元,可用如下三元组来表示:

(节点1, 弧, 节点2)

语义基元也可用图3-6所示的有向图来表示,其中A和B分别代表节点,而R则表示A和B之间的某种语义联系。当把多个语义基元用相应的语义联系关联在一起时,就形成了一个语义网络,如图3-7所示。

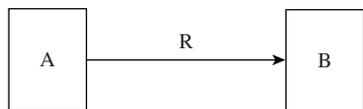


图3-6 语义基元结构

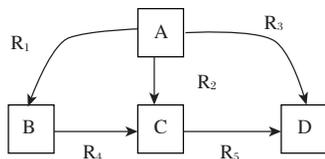


图3-7 语义网络结构

2. 语义网络的基本语义联系

语义网络除了可以描述事物本身之外,还可以描述事物之间错综复杂的关系。基本语义联系是构成复杂语义联系的基本单元,也是语义网络表示知识的基础,因此用一些基本的语义联系组合成任意复杂的语义联系是可以实现的。这里只给出一些经常使用的基本语义关系。

(1) 类属关系。

类属关系是指具体有共同属性的不同事物间的分类关系、成员关系或实例关系,它体现的是“具体与抽象”“个体与集体”的层次分类。其直观意义是“是一个”“是一种”“是一只”等。在类属关系中,最主要的特征是属性的继承性,处在具体层的节点可以继承抽象层节点的所有属性。常用的类属关系如下:

- 1) AKO(A-Kind-of): 表示一个事物是另一个事物的一种类型。
- 2) AMO(A-Member-of): 表示一个事物是另一个事物的成员。
- 3) ISA(Is-a): 表示一个事物是另一个事物的实例。

(2) 包含关系。

包含关系也称为聚集关系,是指具有组织或结构特征的“部分与整体”之间的关系,它和类属关系最主要的区别就是包含关系一般不具备属性的继承性。常用的包含关系的有Part-of、Member-of,含义为一部分,表示一个事物是另一个事物的一部分,或说是部分与整体的关系。用它连接的上下层节点的属性很可能是很不相同的,即Part_of联系不具备属性的继承性。例如,“轮胎是汽车的一部分”,其语义网络表示如图3-8所示。

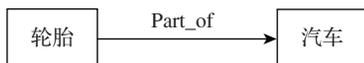


图3-8 包含关系实例

(3) 属性关系。

属性关系是指事物和其属性之间的关系。常用的属性关系如下：

- 1) Have：表示一个节点具有另一个节点描述的属性。
- 2) Can：表示一个节点能做另一个节点的事情。

例如，“鸟有翅膀”“电视机可以放电视节目”，其对应的语义网络表示如图 3-9 所示。



图3-9 属性关系实例

(4) 时间关系。

时间关系是指不同事件在其发生时间方面的先后关系，节点间不具备属性的继承性。常用的时间关系如下：

- 1) Before：表示一个事件在一个事件之前发生。
- 2) After：表示一个事件在一个事件之后发生。

例如，“王芳在黎明之前毕业”“香港回归之后，澳门也回归了”，其对应的语义网络表示如图 3-10 所示。



图3-10 时间关系实例

(5) 位置关系。

位置关系是指不同事物在位置方面的关系。节点间不具备属性的继承性。常用的位置关系如下：

- 1) Located-on：表示一物体在另一物体之上。
- 2) Located-at：表示一物体在某一位置。
- 3) Located-under：表示一物体在另一物体之下。
- 4) Located-inside：表示一物体在另一物体之中。
- 5) Located-outside：表示一物体在另一物体之外。

例如，“天津大学坐落于天津南开区”，其对应的语义网络表示如图 3-11 所示。

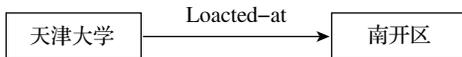


图3-11 位置关系实例

(6) 相近关系。

相近关系又称相似关系，是指不同事物在形状、内容等方面相似和接近。常用的相近关系如下：

- 1) Similar-to：表示一事物与另一事物相似。
- 2) Near-to：表示一事物与另一事物接近。

例如，“狗长得像狼”，其对应的语义网络表示如图 3-12 所示。

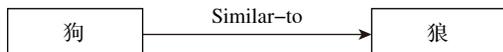


图3-12 相近关系实例

(7) 因果关系。

因果关系是指由于某一事件的发生而导致另一事件的发生，适合表示规则性知识。通常用 If-then 联系表示两个节点之间的因果关系，其含义是“如果…，那么…”。例如，“如果天晴，那么小明骑自行车上班”，其对应的语义网络表示如图 3-13 所示。

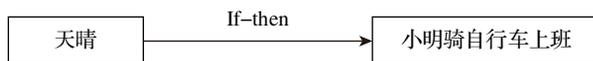


图3-13 因果关系实例

(8) 组成关系。

组成关系用于表示某一事物由其他一些事物构成，通常用 Composed of 联系表示。Composed of 联系所连接的节点间不具备属性的继承性。例如，“整数由正整数、负整数和零组成”，其对应的语义网络表示如图 3-14 所示。

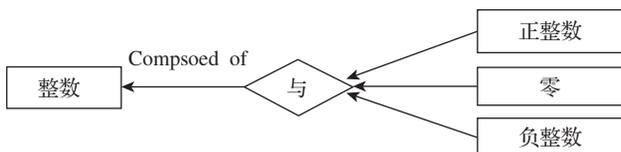


图3-14 组成关系实例

3. 语义网络表示知识的方法及步骤

(1) 事实性知识的表示。

对于一些简单的事实，如“鸟有翅膀”“轮胎是汽车的一部分”，这里要描述这些事实需要两个节点，用前面给出的基本语义联系或自定义的基本语义联系即可以表示。对于稍微复杂的事实，如在一个事实中涉及多个事物时，如果语义网络只被用来表示一个特定的事物或概念，那么当有更多的实例时，就需要更多的语义网络，这样就使问题复杂化了。

通常把有关一个事物或一组相关事物的知识用一个语义网络来表示。

例如，用一个语义网络来表示事实“苹果树是一种果树，果树又是树的一种，树有根、有叶而且树是一种植物”。

这一事实涉及“苹果树”“果树”和“树”这3个对象，树有两个属性“有根”“有叶”。

首先建立“苹果树”节点，为了进一步说明苹果树是一种果树，增加一个“果树”节点，并用 AKO 联系连接两个节点。为了说明果树是树的一种，增加一个“树”节点，并用 AKO 联系连接两个节点。为了进一步描述树“有根”“有叶”的属性，引入“根”和“叶”两个节点，并分别用 Have 联系与“树”节点连接。该事实可用图 3-15 所示的语义网络表示。

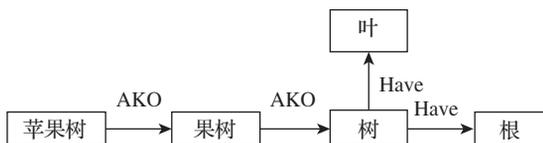


图3-15 有关苹果树的语义网络

(2) 情况、动作和事件的表示。

为了描述复杂的知识，在语义网络的知识表示法中，通常采用引进附加节点的方法来解决。西蒙在提出的表示方法中增加了情况节点、动作节点和事件节点，允许用一个节点来表示情况、动作和事件。

1) 情况的表示。在用语义网络表示那些不及物动词表示的语句或没有间接宾语的及物动词表示的语句时，若该语句的动作表示了一些其他情况，如动作作用的时间等，则需要增加一个情况节点，用于指出各种不同的情况。

例如，用语义网络表示知识“请在 2019 年 11 月前归还图书”。这条知识只涉及一个对象就是“图书”，它表示了 2019 年 11 月前“归还”图书这一情况。为了表示归还的时间，可以增加一个“归还”节点和

一个“情况”节点，这样不仅说明了归还的对象是图书，而且很好地表示了归还图书的时间。图 3-16 为带有情况节点的语义网络，归还图书情况 2019 年 11 月 AKO、Object、Before。

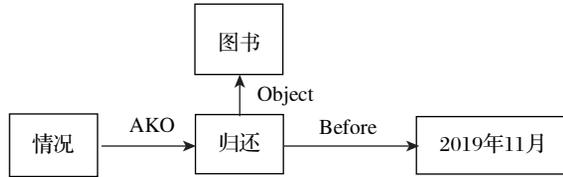


图3-16 带有情况节点的语义网络

2) 动作的表示。有些表示知识的语句既有发出动作的主体，又有接收动作的客体。在用语义网络表示这样的知识时，可以增加一个动作节点用于指出动作的主体和客体。

例如，用语义网络表示知识“校长送给李老师一本书”。这条知识只涉及两个对象，就是“图书”和“校长”，为了表示这个事实，增加一个“送给”节点。其语义网络表示如图 3-17 所示。

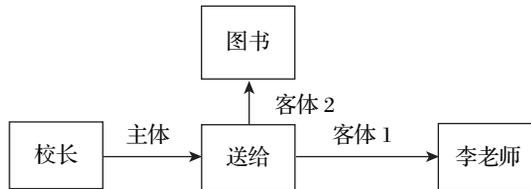


图3-17 带有动作节点的语义网络

3) 事件的表示。如果要表示的知识可以看成发生的一个事件，那么可以增加一个事件节点来描述这条知识。

例如，用语义网络表示知识“中国与日本两国的国家乒乓球队在中国进行一场比赛，结局的比分是 5 : 1”。其语义网络表示如图 3-18 所示。

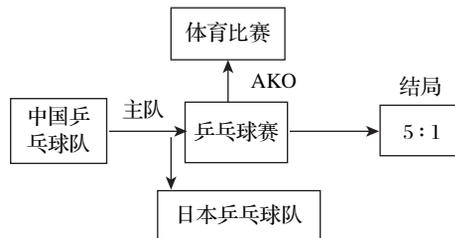


图3-18 带有时间节点的语义网络

(3) 连词和量词的表示。

在稍微复杂的知识中，经常用到像“并且”“或者”“所有的”“有一些”等这样的连接词或量词，在谓词逻辑表示法中，很容易就可以表示这类知识。而谓词逻辑中的连词和量词可以用语义网络来表示。因此，语义网络也能表示这类知识。

(4) 用语义网络表示知识的步骤。

1) 确定问题的所有对象和每个对象的属性。

2) 确定所讨论对象间的关系。

3) 根据语义网络中涉及的关系，对语义网络中的节点及弧进行整理，包括增加节点、弧和归并节点等。

a. 在语义网络中，若节点中的联系是 ISA、AKO 和 AMO 等类属关系，则下层节点对上层节点具有属性继承性。整理同一层节点的共同属性，并抽出这些属性，加入上层节点中，以免造成信息冗余。

b. 若要表示的知识中含有因果关系，则增加情况节点，并从该节点引出多条弧将原因节点和结果节点连接起来。

c. 若要表示的知识中含有动作关系,则增加动作节点,并从该节点引出多条弧,将动作的主体节点和客体节点连接起来。

d. 若要表示的知识中含有“与”和“或”关系,则增加“与”节点和“或”节点,并用弧将这些“与”“或”与其他节点连接起来,表示知识中的语义关系。

e. 若要表示的知识是含有全称量词和存在量词的复杂问题,则采用亨德里克提出的语义网络分区技术来表示。

f. 若要表示的知识是规则性的知识,则应仔细分析问题中的条件与结论,并将它们作为语义网络中的两个节点,最后用 If-then 弧将它们连接起来。

g. 将各对象作为语义网络的一个节点,而各对象间的关系作为网络中各节点的弧,连接形成语义网络。

4. 语义网络知识表示举例

例 3.1 把下列命题用一个语义网络表示出来:

- (1) 猪和羊都是动物。
- (2) 猪和羊都是哺乳动物。
- (3) 野猪是猪,但生长在森林中。
- (4) 山羊是羊,头上长着角。
- (5) 绵羊是一种羊,它能生产羊毛。

解题分析:

问题涉及的对象有猪、羊、动物、哺乳动物、野猪、山羊、绵羊、森林、羊毛和角等。分析它们之间的语义关系,“动物”和“哺乳动物”“哺乳动物”和“猪”“哺乳动物”和“羊”“羊”和“山羊”及“绵羊”、“野猪”和“猪”之间的关系是“是一种”,可用 AKO 来表示;“山羊”和“头上有角”之间是一种属性关系,可用 ISA 来描述;“绵羊”和“羊毛”之间是一种属性关系,可用 Have 来描述;“野猪”和“森林”之间是位置关系,可用 Locate-at 来表示。其对应的语义网络表示如图 3-19 所示。

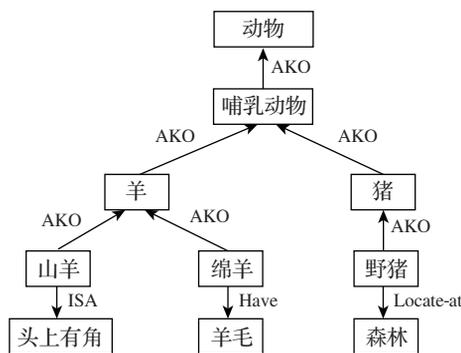


图3-19 有关猪和羊的语义网络

例 3.2 用语义网络表示下列知识: 教师张明在本年度第二学期给计算机应用专业的学生讲授“人工智能”这一门课程。

解题分析:

本例涉及的对象包括教师、张明、学生、计算机应用、人工智能和本年度第二学期等。确定各对象间的关系。“张明”与“教师”之间是一种类属关系,可用 ISA 表示;“学生”和“计算机应用”之间是一种属性关系,可以用 Major 表示;“张明”“学生”“人工智能”则通过“讲课”这一动作联系在一起。从上面的分析可知,必须增加一个动作节点“讲课”,“张明”是这一动作的主体,而“学生”“人工智能”是这一动作的两个客体。“本年度第二学期”则是这个动作的作用时间,属于一种时间关系。因此,通过增加动作节点“讲课”,即可将网络中的各节点联系起来。由“讲课”节点引出的弧不仅指出了讲课的主体和客体,还指出了讲课的时间。

其对应的语义网络表示如图 3-20 所示。

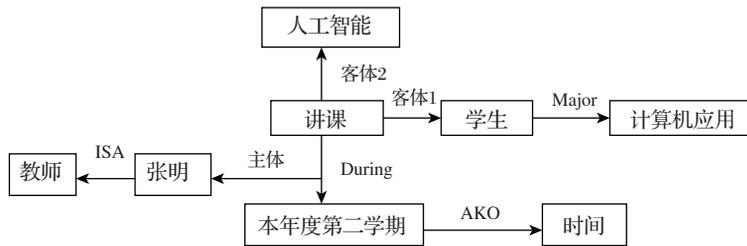


图3-20 有关讲课的语义网络

3.4.3 框架表示法

框架表示法是以框架理论为基础发展起来的一种结构化的知识表示，它适用于表达多种类型的知识。1975年，美国麻省理工学院明斯基 (Minsky) 在论文 *A framework for representing knowledge* 中提出了框架理论，引起了人工智能学者的重视。

它是针对人们在理解情景、故事时提出的心理学模型，论述的是思想方法，而不是具体实现。

框架理论的基本观点是：人脑已存储大量的典型情景，当人面临新的情景时，就从记忆中选择（粗匹配）一个称为框架的基本知识结构，该框架是以前记忆的一个知识空框，而其具体内容依新的情景而改变。对该空框进行细节加工、修改和补充，形成对新情景的认识，又记忆于人脑中，以丰富人的知识。

1. 框架结构

框架是表示某一类情景的结构化的一种数据结构。框架由描述事物的各个方面的槽组成，每个槽可有若干个侧面。一个槽用于描述所讨论对象的某一方面的属性，一个侧面用于描述相应属性的一个方面。槽和侧面具有的值分别称为槽值和侧面值。槽值可以是逻辑的、数字的，可以是程序、条件、默认值或是一个子框架。槽值含有如何使用框架信息、下一步可能发生的信息、预计未实现该如何做的信息等。

在一个用框架表示的知识系统中，一般含有多个框架，为了区分不同的框架以及一个框架内不同的槽、不同的侧面，需要分别赋予不同的名字，分别称为框架名、槽名和侧面。因此，一个框架通常由框架名、槽名、侧面和值这4个部分组成，其一般结构如图 3-21 所示。

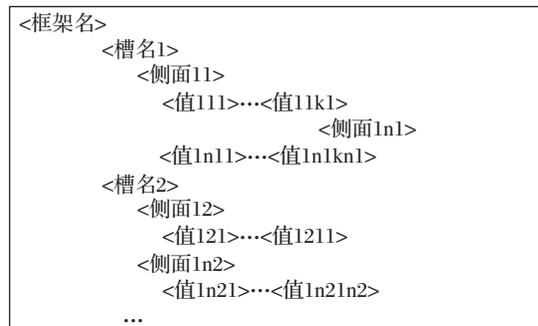


图3-21 框架的一般结构

例如，要用框架来描述“优质商品”这一概念。首先分析商品具有的属性，一个商品可能具有的属性有商品名称、生产厂家、生产日期和获奖情况等，这里只考虑这几个属性。这几个属性可以定义为“优质商品”框架的槽，而“获奖情况”这个属性还可以从获奖等级、颁奖单位和获奖时间这3个侧面来加以描述。如果给各个槽和侧面赋予具体的值，就得到了“优质商品”这一概念的一个实例框架，如图 3-22 所示。

框架名: <优质商品>
商品名称: 红桃K
生产厂家: 红桃K集团
生产日期: 1998年6月17日
获奖情况: 获奖等级: 省级
颁奖单位: 湖北省卫生厅
获奖时间: 2000年5月

图3-22 框架表示法实例

通常在框架系统中定义一些公用、常用且标准的槽名, 并把这些槽名称为系统预定义槽名。人们在使用这些槽名时, 不用说明就知道它们表示何种联系。下面给出几个比较常用的、用来表示对象间关系的槽名。

(1) ISA 槽。ISA 槽用于指出对象间抽象概念上的类属关系, 其直观意义是“是一个”“是一种”“是一只”等。在一般情况下, 用 ISA 槽指出的联系都具有继承性。

框架的继承性就是指当下层框架中的某些槽值或侧面值没有被直接给定时, 可以从其上层框架中继承这些值或属性。例如, 椅子一般有 4 条腿, 当一把具体的椅子没有指出它有几条腿时, 则可以通过一般椅子的特性, 得出它有 4 条腿。

(2) AKO 槽。AKO 槽用于具体地指出对象间的类属关系, 其直观意义是“是一种”。当用它作为某下层框架的槽时, 就明确地指出了该下层框架所描述的事物是其上层框架所描述事物中的一种, 下层框架可继承上层框架中的值或属性。

(3) Instance 槽。Instance 槽用来表示 AKO 槽的逆关系。当用它作为某上层框架的槽时, 可在该槽中指出它所联系的下层框架。用 Instance 槽指出的联系都具有继承性, 即下层框架可继承上层框架中所描述的属性或值。

(4) Part-of 槽。Part-of 槽用于指出部分和整体的关系。当用它作为某框架的一个槽时, 槽中所填的值称为该框架的上层框架名, 该框架所描述的对象只是其上层框架所描述对象的一部分。例如, “两条腿”是“人体”的一部分。可以将“两条腿”和“人体”分别定义成框架, “两条腿”为下层框架, “人体”为其上层框架。在“两条腿”的框架中设置一个 Part-of 槽, 槽值填入 <人体> 这个框架名。

显然, 用 Part-of 槽指出的联系所描述的下层框架和上层框架之间不具有继承性。

2. 框架表示法实例

例 3.3 描述一个具体教师的框架

框架名: <教师 -1>

类属: <大学教师>

姓名: 张宇

性别: 男

年龄: 32

职业: <教师>

职称: 副教授

部门: 计算机系

研究方向: 计算机软件与理论

参加工作时间: 2003 年 7 月



框架表示法

工龄：当前年份为 2003

工资：<工资单>

解题分析：

比较上面几个例子，可以发现“教师-1”是“大学教师”的下层框架，而“大学教师”又是“教师”的下层框架，“教师”又是“知识分子”的下层框架。框架之间的这种层次关系是相对而言的，下层框架可以从上层框架继承某些属性或值。这样，一些相同的信息可以不必重复存储，节省了存储空间，这种层次结构对减少冗余信息有重要意义。

例 3.4 下面为有关地震的新闻报道，请用框架结构表示这段报道。

今天，一次强度为里氏 8.5 级的强烈地震袭击了下斯洛文尼亚 (Low Slabovia) 地区，造成 25 人死亡和 5 亿美元的财产损失。下斯洛文尼亚地区主席说：“多年来，靠近萨迪壕金斯断层的重灾区一直是一个危险地区。这是本地区发生的第 3 号地震。”

解题分析：

首先分析关于地震报道中涉及的一些有关地震的关键属性，包括地震发生的地点、时间、伤亡人数、财产损失数量、地震强度的震级和断层情况。这些属性可以作为该框架的各个槽。接下来，将本报道中的有关数据填入相应的槽后，就得到了第 3 号地震的框架。

框架名：<第 3 号地震>

地点：Low Slabovia

时间：今天

伤亡人数：25

财产损失：5 亿美元

震级：8.5 级

断层：萨迪壕金斯

3.4.4 过程表示法

过程表示法着重于对知识的利用，它把与问题有关的知识以及如何运用这些知识求解问题的控制策略都表述为一个或多个求解问题的过程，每一个过程一段程序，用于完成对一个具体时间和情况的处理。在问题求解过程中，当需要使用某个过程时，就调用相应的程序并执行。

1. 过程表示法举例

例如，有如下知识：

若 x 与 y 是兄弟，且 x 是 z 的父亲，则 y 是 z 的叔父。

用产生式规则（说明性知识表示法）表示如下：

IF brother(x , y) and father(x , z)

then unce(y , z)

用过程性知识（过程表示法有多种表示形式，在此选过程规则表示）表示如下：

BR(Uncle ? y ? z)

GOAL(brother ? x y)

GOAL(father x z)

INSERT(Uncle y z)

RETURN

其中, BR 是后向推理的标志; GOAL 表示求解子目标, 即进行过程调用; INSTER 表示对数据库实施插入操作; RETURN 表示该过程规则结束。带“?”号的变量表示其值将在该过程中求得。

上述过程规则的含义: 按后向推理方式进行推理, 为了求解 (Uncle ? y ? z), 首先应通过过程调用求解 (Brother ? x y), 得到 x 的值; 然后将得到的 x 值传递给 (Father x z), 并求解它。如果这些都成功, 就将 (Uncle y z) 插入数据库中, 并将控制权返回给调用者。

一般来说, 一个过程规则包括以下 4 个部分:

(1) 激发条件。激发条件由两部分组成, 即推理方向和调用模式。

1) 推理方向指出其推理是前向推理还是后向推理。

2) 若为前向推理, 则只有当数据库中有已知事实可与其“调用模式”匹配时, 该过程规则才能被激活; 若为后向推理, 则只有当“调用模式”与查询目标或子目标匹配时, 才能将该过程规则激活。

(2) 演绎操作。演绎操作由一系列的子目标构成, 当上面的激发条件被满足时, 将执行这里列出的演绎操作, 如 GOAL(brother ? x y)、GOAL(father x z)。

(3) 状态转换。状态转换操作用于对数据库进行增、删、改, 分别用 INSTER、DELETE、MODIFY 语句实现。

(4) 返回。过程规则的最后一个语句是 RETURN, 用于将控制权返回给调用该过程规则的上级过程规则。

2. 求解问题的基本过程

每当有一个新的目标时, 就从可用的过程规则中选择一个 (设为 R), 并执行过程规则 R; 在 R 的执行过程中可能又将产生新的目标, 此时调用相应的过程规则并执行。反复进行上述过程, 直到执行到 RETURN, 这是就将控制权返回给调用当前过程规则的上级过程规则。

在这一过程中, 如果某过程规则运行失败, 就选择另一同层的可用过程规则执行。如果不存在这样的过程规则, 则返回失败标志并将执行的控制权移交给上级过程规则。

例 3.5 设数据库中有以下已知事实:

(Brother 刘海 刘杨)

(Father 刘海 刘小海)

需要求解的问题是: 找出两个人 u 及 v, 其中 u 是 v 的叔父。

该问题可表示为 GOAL(Uncle ? u ? v)。

规则库中的规则表示如下:

BR(Uncle ? y ? z)

GOAL(brother ? x y)

GOAL(father x z)

INSERT(Uncle y z)

RETURN

求解该问题的过程如下:

(1) 在过程规则库中找出对于问题 GOAL(Uncle ?u ?v) 的激发条件可被满足的过程规则。显然, BR(Uncle ?y ?z) 经 u/y、v/t 的变量代换后可以匹配, 所以选用该规则。

(2) 执行该规则的第 2 个语句 GOAL(Brother ?x y), 此时 y 已被 u 代换, 经与已知事实 Brother(刘海, 刘洋) 匹配, 得 x= 刘海, u= 刘洋。

(3) 执行该规则过程的第 3 个语句 GOAL(Father x z), 可得与已知事实 Father(刘海 刘小海) 匹配, 求得

变量 v 的值：v= 刘小海。

(4) 执行该过程规则的第 4 个语句 Insert(Uncley y z)，此时 y 与 z 的值均已知，所以插入数据库中的事实是 Uncle(刘杨 刘小海)。这表明“刘洋是刘小海的叔父”，从而求得了问题的解。

该问题的过程表示法求解过程如图 3-23 所示。

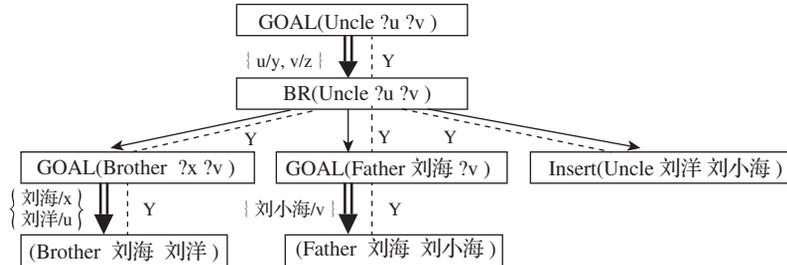


图3-23 过程表示法求解过程

3.4.5 脚本表示法

1975 年，脚本表示法是夏克 (Schank) 依据概念依赖理论提出的一种知识表示方式。它是知识工程中框架表示方法的一种特殊形式，它用一组槽来描述某些事件的发生序列，就像剧本中的事件序列一样，故称为脚本。

脚本用一些原语作为槽名来表示对象的基本行为，描述某些时间的发生序列，类似于电影剧本。脚本表示的知识有强烈的因果结构，系统对事件的处理必须是一个动作完成之后才能完成另一个。整个过程的启动取决于开场条件，只有满足于脚本的开场条件，脚本中的时间才有可能发生，而脚本的结果就是所有动作完成后的系统结果。由于脚本是以非常固定的形式描述的，因此在预言一些没有直接提及的事件方面特别有用。但如果事件被强行中断，即给定情节中的某个时间与脚本中的事件不能对应，脚本就不能预测被中断以后的事件。与语义网络表示法和框架表示法相比，显然脚本表示法的表达能力要弱得多，知识表达的范围也很有限，但是对于预先就已经“构思”好的特定知识而言，不失为一个相当合适的表示方法。

脚本由以下基本组成部分组成：

- (1) 进入条件：指出脚本描述的事件可能发生的先决条件，即事件发生的前提条件角色，其描述事件中可能出现的人物。
- (2) 道具：描述事件中可能出现的相关物体舞台，即事件发生的空间。
- (3) 场景：描述事件序列，可以有多个场景（场景可变化，剧情可变化）。
- (4) 结局：给出在剧本描述的事件发生以后通常产生的结果。

脚本表示法可实现海军特种作战方案的辅助生成，但传统的脚本知识表示方式较为呆板，能力有限，不具备对于元素的基本属性的描述能力，难以描述多变的作战现实。通过改进脚本，可为海军特种作战方案与计划的生成提供更科学、有效、灵活的辅助。

脚本表示法是一种特殊的框架表示法，它把生活中的事件的相互依赖关系编制成脚本，只有当满足特定的进入条件时，才会按照预先设定的脚本情节发展下去。根据这种特点，知识的脚本表示法适用于一些预先方案的设计，不过其前提是对该领域有一定的经验并了解事情可能会沿着哪个方向向前发展。

因此采用脚本表示法辅助于海军特种作战方案设计，在该设计方案中，充分考虑海军特种作战时的实际情况，对脚本表示法进行改进并扩充了一些方案设计所需的概念，如元作战环境、元目标、元兵力和元动作，

这些是脚本表示法应用于海军特种作战领域所必需的一些概念。因为特种作战涉及的领域灵活多变,因此作战方案的设计就显得尤为重要。通过学习特种作战方案设计的知识表示过程,将作战环境、作战任务、作战兵力和作战行动都细分为元作战环境、元目标、元兵力和元行动,然后由这些元元素构成元情节网。通过这些元素的细分,在方案设计中,基本上使兵力与目标实现了一一对应,这样才能使得作战有条不紊地进行。在知识的表示过程中,采用BNF范式表示脚本中的各个元素,详细地从各个方面描述脚本中的元素,考虑各个属性对作战方案的影响,从而按照原有的经验与流程框架构建出适合于特种作战的方案。从整个过程可以看出,脚本表示法遵循一定的客观经验与事物的发展规律,对于灵活多变的实际情况难以适应,它只适用于一些预测性的领域,如演习方案设计、作战方案设计和活动方案设计等。



思考与练习

1. 试述知识的几种分类方法。

2. 试述知识的特征。

3. 专家系统的优点有哪些?

4. 试述专家系统设计的步骤。
