

摘 要

矿井通风是矿山生产的一个重要环节,一个安全、可靠、经济、实用的矿井通风系统,对保证井下安全生产有重要意义。目前,通风安全管理还不能及时准确地获得通风安全信息,而且还会延误各类重大决策。因此,矿井通风网络自动解算和在线监测异常数据分析专家系统软件的开发,可以大大提高通风管理水平,增强矿井的抗灾能力。

本文介绍了矿井通风网络解算的相关原理,研究了矿井通风网络在计算机中的存储和解算方法。介绍了专家系统的相关原理和它的特点与优点以及专家系统和常规系统的区别。深入研究了专家系统知识的表示方法和知识的检测及求精方法,主要对产生式表示方法进行了研究,阐述了产生式规则表示方法和它的基本要素,讨论了知识完整性和一致性的检测方法。描述了推理机的功能,研究了推理机的控制和冲突消解策略。结合矿井通风网络实例,在通风网络解算和知识库及推理机的研究基础上,应用 Visual C++ 语言开发了基于 Windows 的通风网络解算和诊断系统。

关键词: 矿井通风网络; 专家系统; 知识库; 推理机

Abstract

Mine ventilation is an important part of the production under a mine. A safe, reliable, economic and practical mine ventilation system is important in guaranteeing the safety in production under a mine. At present, the safety management of ventilation can't accurately get safe information of ventilation in time yet and will also delay all kinds of important decisions. So, the development of Computer-Aided Calculation of mine ventilation network and monitoring exceptional data analysis expert system online can improve the ventilation management level and strengthen the anti-disaster ability of mine.

The relative principles of mine ventilation network calculation are introduced and how to store mine ventilation network and calculate in computer is studied. The relative principles of expert system, its characteristics, advantages and the differences between expert system and routine system are also introduced. The knowledge expression method and the method of knowledge check and refinement are further studied. The expression method of production rule and its basic elements are expatiated and the knowledge integrality and consistency detection are discussed. The functions of the inference engine are described and the tactics of inference engine control and conflict clearing up are studied too. After the ventilation network calculation, knowledge base and inference engine are studied, the ventilation network calculation and diagnosis system based on Windows is developed using Visual C++ language and combining with an instance of mine ventilation network.

Keywords: mine ventilation network; expert system; knowledge base; inference engine

哈尔滨工程大学 学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：姜元宝

日期：2005年 2月 28日

第 1 章 绪论

1.1 课题的来源、研究意义及现状

1.1.1 题目的来源

煤矿矿井瓦斯爆炸事故综合预防及关键技术研究是黑龙江省 2004 年度重大科技攻关计划项目, 矿井通风网络自动解算和矿井通风系统在线监测异常数据分析专家系统是该项目的一个课题。

1.1.2 题目研究意义及现状

矿井通风是矿山生产的一个重要环节, 一个安全、可靠、经济、实用的矿井通风系统, 对保证井下安全生产有重要意义。煤矿生产过程中的瓦斯爆炸、煤尘爆炸、矿井火灾、有毒气体窒息等灾害的发生都与通风有直接的关系。可以说, 通风状况的好坏直接影响工人的安全、健康和劳动效率, 直接关系到煤矿的安全生产、经济效益和可持续发展。

因此, 随着计算机在煤矿的普及, 应用计算机建立智能化矿井通风系统, 实现通风网络解算自动化和风机、风门、风筒、瓦斯等故障诊断具有深远的现实意义。

煤矿通风安全对煤炭工业影响巨大, 通风安全工作在煤矿生产中占有重要地位, 其管理好坏直接关系到煤矿的安全生产和经济效益。目前, 国内外在通风技术和管理方面取得了很大成就, 通风系统中主通风机运行状态、局部通风机开停、风门关启、风速、风压、风流温度和湿度均实现了自动监测。

但目前我省通风安全管理还存在如下问题。

- (1) 风量调节凭经验, 时间长、效果差;
- (2) 通风系统安全状况评价凭经验;

第 1 章 绪论

1.1 课题的来源、研究意义及现状

1.1.1 题目的来源

煤矿矿井瓦斯爆炸事故综合预防及关键技术研究是黑龙江省 2004 年度重大科技攻关计划项目, 矿井通风网络自动解算和矿井通风系统在线监测异常数据分析专家系统是该项目的一个课题。

1.1.2 题目研究意义及现状

矿井通风是矿山生产的一个重要环节, 一个安全、可靠、经济、实用的矿井通风系统, 对保证井下安全生产有重要意义。煤矿生产过程中的瓦斯爆炸、煤尘爆炸、矿井火灾、有毒气体窒息等灾害的发生都与通风有直接的关系。可以说, 通风状况的好坏直接影响工人的安全、健康和劳动效率, 直接关系到煤矿的安全生产、经济效益和可持续发展。

因此, 随着计算机在煤矿的普及, 应用计算机建立智能化矿井通风系统, 实现通风网络解算自动化和风机、风门、风筒、瓦斯等故障诊断具有深远的现实意义。

煤矿通风安全对煤炭工业影响巨大, 通风安全工作在煤矿生产中占有重要地位, 其管理好坏直接关系到煤矿的安全生产和经济效益。目前, 国内外在通风技术和管理方面取得了很大成就, 通风系统中主通风机运行状态、局部通风机开停、风门关启、风速、风压、风流温度和湿度均实现了自动监测。

但目前我省通风安全管理还存在如下问题。

- (1) 风量调节凭经验, 时间长、效果差;
- (2) 通风系统安全状况评价凭经验;

(3) 通风网络解算、风量调节还没有实现自动化。

这些问题的存在, 不仅不能及时准确地获得通风安全信息, 而且还会延误各类重大决策, 导致重大安全事故的产生。

因此, 矿井通风网络自动解算和矿井通风系统在线监测异常数据分析专家系统软件的开发, 可以大大提高通风管理水平, 增强矿井的抗灾能力, 确保煤矿安全生产, 具有广阔的应用市场。

1.2 本文所做的工作

本文所做的工作主要有以下几点。

(1) 熟悉煤矿通风网络的各个组成部分。建造一个通风专家系统, 相当于学习一门新的专业, 只有熟悉煤矿通风网络, 才能对煤矿通风网络解算和诊断有一个全面的认识, 才能掌握通风网络解算和诊断的内容。

(2) 研究了网络解算的算法, 采用斯考德——恒斯雷算法和图论知识, 实现了矿井通风网络解算。

(3) 分析了该专家系统的知识表示方法, 根据客观情况和实际表示的因果关系, 选择用产生式规则的知识表示方法; 在通风专家指导下和参考矿井通风书籍后, 构建了系统的知识库, 是系统的重要组成部分。

(4) 完成推理机的设计和实现。系统的推理机在知识库的基础上构建。推理机是专家系统的组织控制结构, 它按一定的策略进行推理和运用知识库中的知识, 最终给出诊断结果。它是在产生式规则的基础上进行正向推理, 是该系统的另一重要部分。

1.3 本文的结构和组织

全文共分 4 章。第 1 章是绪论, 第 2 章是相关技术, 第 3 章是知识库与推理机的研究, 第 4 章是通风网络解算与故障诊断系统的设计与实现。具体的结构为:

第 1 章介绍课目来源、研究意义、现状、主要内容以及结构组织。

第 2 章阐述矿井通风网络中的一些基本理论和专家系统的基本原理及特

(3) 通风网络解算、风量调节还没有实现自动化。

这些问题的存在, 不仅不能及时准确地获得通风安全信息, 而且还会延误各类重大决策, 导致重大安全事故的产生。

因此, 矿井通风网络自动解算和矿井通风系统在线监测异常数据分析专家系统软件的开发, 可以大大提高通风管理水平, 增强矿井的抗灾能力, 确保煤矿安全生产, 具有广阔的应用市场。

1.2 本文所做的工作

本文所做的工作主要有以下几点。

(1) 熟悉煤矿通风网络的各个组成部分。建造一个通风专家系统, 相当于学习一门新的专业, 只有熟悉煤矿通风网络, 才能对煤矿通风网络解算和诊断有一个全面的认识, 才能掌握通风网络解算和诊断的内容。

(2) 研究了网络解算的算法, 采用斯考德——恒斯雷算法和图论知识, 实现了矿井通风网络解算。

(3) 分析了该专家系统的知识表示方法, 根据客观情况和实际表示的因果关系, 选择用产生式规则的知识表示方法; 在通风专家指导下和参考矿井通风书籍后, 构建了系统的知识库, 是系统的重要组成部分。

(4) 完成推理机的设计和实现。系统的推理机在知识库的基础上构建。推理机是专家系统的组织控制结构, 它按一定的策略进行推理和运用知识库中的知识, 最终给出诊断结果。它是在产生式规则的基础上进行正向推理, 是该系统的另一重要部分。

1.3 本文的结构和组织

全文共分 4 章。第 1 章是绪论, 第 2 章是相关技术, 第 3 章是知识库与推理机的研究, 第 4 章是通风网络解算与故障诊断系统的设计与实现。具体的结构为:

第 1 章介绍课目来源、研究意义、现状、主要内容以及结构组织。

第 2 章阐述矿井通风网络中的一些基本理论和专家系统的基本原理及特

(3) 通风网络解算、风量调节还没有实现自动化。

这些问题的存在, 不仅不能及时准确地获得通风安全信息, 而且还会延误各类重大决策, 导致重大安全事故的产生。

因此, 矿井通风网络自动解算和矿井通风系统在线监测异常数据分析专家系统软件的开发, 可以大大提高通风管理水平, 增强矿井的抗灾能力, 确保煤矿安全生产, 具有广阔的应用市场。

1.2 本文所做的工作

本文所做的工作主要有以下几点。

(1) 熟悉煤矿通风网络的各个组成部分。建造一个通风专家系统, 相当于学习一门新的专业, 只有熟悉煤矿通风网络, 才能对煤矿通风网络解算和诊断有一个全面的认识, 才能掌握通风网络解算和诊断的内容。

(2) 研究了网络解算的算法, 采用斯考德——恒斯雷算法和图论知识, 实现了矿井通风网络解算。

(3) 分析了该专家系统的知识表示方法, 根据客观情况和实际表示的因果关系, 选择用产生式规则的知识表示方法; 在通风专家指导下和参考矿井通风书籍后, 构建了系统的知识库, 是系统的重要组成部分。

(4) 完成推理机的设计和实现。系统的推理机在知识库的基础上构建。推理机是专家系统的组织控制结构, 它按一定的策略进行推理和运用知识库中的知识, 最终给出诊断结果。它是在产生式规则的基础上进行正向推理, 是该系统的另一重要部分。

1.3 本文的结构和组织

全文共分 4 章。第 1 章是绪论, 第 2 章是相关技术, 第 3 章是知识库与推理机的研究, 第 4 章是通风网络解算与故障诊断系统的设计与实现。具体的结构为:

第 1 章介绍课目来源、研究意义、现状、主要内容以及结构组织。

第 2 章阐述矿井通风网络中的一些基本理论和专家系统的基本原理及特

点，也对专家系统与常规系统的区别进行了介绍。

第 3 章介绍了专家系统中的知识库表示方法，比较了各种知识表示方法的优缺点，并对知识的检测与求精进行了研究；介绍了专家系统中推理机的功能和推理策略。

第 4 章给出了某矿井通风网路图，根据斯考德——恒斯雷法和图论知识，解算该矿井的网路，并且完成系统网络诊断的知识库和推理机的设计与实现。

第 2 章 相关技术

2.1 矿井通风网络解算基本理论

图论是应用非常广泛的一个数学分支，它是建立和研究离散数学模型的一个重要数学工具。图论作为一门科学是由 18 世纪的瑞士数学家欧拉创立的，在电子计算机问世以后，应用更加广泛。矿井通风网络的数学基础是图论。同样矿井通风网络解算理论与矿井通风本身的知识基础是不可分的^[1]。

本节主要为建立矿井通风网络数学模型的需要，简单地介绍图论中的基本知识和通风网络中风流流动的基本规律以及矿井通风网络解算的斯考德-恒斯雷法的数学模型等^[2-6]。

2.1.1 基本术语

图在各个领域的应用是十分广泛的，最典型的应用领域有电路分析、寻找最短路径、项目规划、鉴别化合物、统计力学、遗传学、控制论、语言学，以及一些社会科学。实际上在所有的数学结构中，图的应用是最广泛的。

图(Graph)是由顶点集合(Vertex)及顶点间的关系集合组成的一种结构：

$$G=(V, E) \quad (2-1)$$

其中，V 是顶点的有穷非空集合；E 是顶点之间关系的有穷集合，也叫边(Edge)集合。

在通风网络解析中，常应用以下几个术语：

- (1) 节点：三条或三条以上风路汇合或分岔的交点。
- (2) 分支风路：连接两个节点的任一风道。任何支路仅在它的端点才有节点。
- (3) 网络：由矿井各风路组成的网状系统，它是由通风系统转化而成的一个由节点和支路组成的线性图，它反映了原通风系统中节点与支路相联结

第 2 章 相关技术

2.1 矿井通风网络解算基本理论

图论是应用非常广泛的一个数学分支，它是建立和研究离散数学模型的一个重要数学工具。图论作为一门科学是由 18 世纪的瑞士数学家欧拉创立的，在电子计算机问世以后，应用更加广泛。矿井通风网络的数学基础是图论。同样矿井通风网络解算理论与矿井通风本身的知识基础是不可分的^[1]。

本节主要为建立矿井通风网络数学模型的需要，简单地介绍图论中的基本知识和通风网络中风流流动的基本规律以及矿井通风网络解算的斯考德-恒斯雷法的数学模型等^[2-6]。

2.1.1 基本术语

图在各个领域的应用是十分广泛的，最典型的应用领域有电路分析、寻找最短路径、项目规划、鉴别化合物、统计力学、遗传学、控制论、语言学，以及一些社会科学。实际上在所有的数学结构中，图的应用是最广泛的。

图(Graph)是由顶点集合(Vertex)及顶点间的关系集合组成的一种结构：

$$G=(V, E) \quad (2-1)$$

其中，V 是顶点的有穷非空集合；E 是顶点之间关系的有穷集合，也叫边(Edge)集合。

在通风网络解析中，常应用以下几个术语：

- (1) 节点：三条或三条以上风路汇合或分岔的交点。
- (2) 分支风路：连接两个节点的任一风道。任何支路仅在它的端点才有节点。
- (3) 网络：由矿井各风路组成的网状系统，它是由通风系统转化而成的一个由节点和支路组成的线性图，它反映了原通风系统中节点与支路相联结

的结构关系。在矿井通风网络中，流经其间的介质为空气。

(4) 网孔:在网络中由二条以上的支路于二个以上的不同节点所构成的闭合路径。在网孔中除首节点外，节点与支路皆不重复。

(5) 风向:表示风流在支路中的流动方向。分支的方向用箭头表示，与分支有关的两个端点中，箭头指向的为终结点、另一个为始节点。网孔的方向一般规定顺时针为正，逆时针为负。

(6) 网络通风参数:一般指的是构成网络的每一支路的风阻、风压和风量，以及整个网络中的总风阻、总风压和总风量，此外，还包括通风机的风量、风压和自然风压值。

(7) 有向图(Directed Graph)和无向图(Undirected Graph):在有向图中，顶点对 $\langle u, v \rangle$ 是有序的，它称为从顶点 u 到顶点 v 的一条有向边。因此， $\langle u, v \rangle$ 和 $\langle v, u \rangle$ 是不同的两条边。此时，顶点对 $\langle u, v \rangle$ 用一对尖括号括起来， u 是有向边的始点， v 是有向边的终点。在无向图中，顶点对 (u, v) 是无序的，它称为与顶点 u 和顶点 v 相关联的一条边，这条边没有特定的方向， (u, v) 与 (v, u) 是同一条边。一般为了有别于有向图，顶点对是用一对圆括号括起来。

(8) 路径(Path):在图 G 中若从顶点 V_i 出发，沿一些边经过一些顶点 $V_{p1}, V_{p2}, \dots, V_{pm}$ ，到达顶点 V_j 。则称顶点序列 $(V_i, V_{p1}, V_{p2}, \dots, V_{pm}, V_j)$ 为从顶点 V_i 到 V_j 的路径。

(9) 简单路径与回路(Cycle):若路径上各顶点 V_1, V_2, \dots, V_m 均不互相重复，则称这样的路径为简单路径。若路径上的第一个顶点和最后一个顶点重合，则称这样的路径为回路。

(10) 连通图(Connected graph)与连通分量(Connected Component):在无向图中，若从顶点 u 到 v 有路径，则称顶点 u 与 v 是连通的。如果图中任意一对顶点都是连通的，则称此图是连通图。非连通图的极大连通子图叫做连通分量。

(11) 强连通图与强连通分量(Strongly connected digraph and Strongly Connected Component):在有向图中，若对于每一对顶点 V_i 和 V_j 都存在一条从 V_i 到 V_j 和 V_j 到 V_i 的路径，则称此图是强连通图。非强连通图的极大强连通子图叫做强连通分量。

(12) 生成树 (Spanning Tree): 一个连通图的生成树是它的极小连通子图, 在 n 个顶点的情形下, 有 $n-1$ 条边。但有向图则可能得到它的由若干有向树组成的生成森林。

2.1.2 矿井通风网络和它在计算机中的存储

对任一矿井通风系统, 如果不考虑各风流交汇点和通风巷道的位置、长度、形状及断面大小等情况, 仅以单线表示各交汇点与风流的联结关系, 那么, 这种只用点的集合和线的集合来表示的通风系统图以及各线所联系得通风参数所组成的系统称为矿井通风网络。

矿井通风系统图是用点的集合和线的集合表示其图图形的, 计为 $G=(V, E)$ 。其中 V 为结点 (即分流交汇点) 的集合, 它所包含节点数为 $|V|=m$; 而 E 为边 (对应通风巷道) 的集合, 它所含的边数为 $|E|=n$ 。即为 $V=\{v_1, v_2, \dots, v_m\}$, $E=\{e_1, e_2, \dots, e_n\}$ 。若不考虑各节点的位置, 按通风巷道与各交汇点之间的联结关系, 连接边 $e_k=(v_i, v_j)$, 而且矿井通风系统是一个有向的连通体系, 所以图 $G=(V, E)$ 是一个有向连通图, 其中的各有向边称为弧。若以通风巷道中的有关通风参数如风阻、风量、阻力等对应的边赋权, 则图 $G=(V, E)$ 就是一个有向连通赋权图。

图有两种基本的存储方式即邻接矩阵和邻接表, 本系统网络生成模块采用的是邻接矩阵存储通风网络图。邻接矩阵是表示顶点之间相邻关系的矩阵。设 $G=(V, E)$ 是具有 m 个顶点的网络图, 顶点序号依次为 $1, 2, \dots, m$, 则 G 的邻接矩阵是具有如下定义的 m 阶方阵 A :

$$A[i, j]=\begin{cases} w_{i,j} & \text{若 } (v_i, v_j) \in E(G) \\ 0 \text{ 或 } \infty & \text{其它} \end{cases} \quad (2-2)$$

2.1.3 矿井通风网络计算方法

多年以来, 国内外学者对矿井通风网络的自然分风计算, 提出了各种不同的方法^{[9][10]}如斯考德——恒斯雷法、牛顿法、节点压力法等。用代数法解算复杂网路的原理是依据风量平衡定律、风压平衡定律、风阻平衡定律、阻力

(12) 生成树 (Spanning Tree): 一个连通图的生成树是它的极小连通子图, 在 n 个顶点的情形下, 有 $n-1$ 条边。但有向图则可能得到它的由若干有向树组成的生成森林。

2.1.2 矿井通风网络和它在计算机中的存储

对任一矿井通风系统, 如果不考虑各风流交汇点和通风巷道的位置、长度、形状及断面大小等情况, 仅以单线表示各交汇点与风流的联结关系, 那么, 这种只用点的集合和线的集合来表示的通风系统图以及各线所联系得通风参数所组成的系统称为矿井通风网络。

矿井通风系统图是用点的集合和线的集合表示其图图形的, 计为 $G=(V, E)$ 。其中 V 为结点 (即分流交汇点) 的集合, 它所包含节点数为 $|V|=m$; 而 E 为边 (对应通风巷道) 的集合, 它所含的边数为 $|E|=n$ 。即为 $V=\{v_1, v_2, \dots, v_m\}$, $E=\{e_1, e_2, \dots, e_n\}$ 。若不考虑各节点的位置, 按通风巷道与各交汇点之间的联结关系, 连接边 $e_k=(v_i, v_j)$, 而且矿井通风系统是一个有向的连通体系, 所以图 $G=(V, E)$ 是一个有向连通图, 其中的各有向边称为弧。若以通风巷道中的有关通风参数如风阻、风量、阻力等对应的边赋权, 则图 $G=(V, E)$ 就是一个有向连通赋权图。

图有两种基本的存储方式即邻接矩阵和邻接表, 本系统网络生成模块采用的是邻接矩阵存储通风网络图。邻接矩阵是表示顶点之间相邻关系的矩阵。设 $G=(V, E)$ 是具有 m 个顶点的网络图, 顶点序号依次为 $1, 2, \dots, m$, 则 G 的邻接矩阵是具有如下定义的 m 阶方阵 A :

$$A[i, j]=\begin{cases} w_{i,j} & \text{若 } (v_i, v_j) \in E(G) \\ 0 \text{ 或 } \infty & \text{其它} \end{cases} \quad (2-2)$$

2.1.3 矿井通风网络计算方法

多年以来, 国内外学者对矿井通风网络的自然分风计算, 提出了各种不同的方法^{[9][10]}如斯考德——恒斯雷法、牛顿法、节点压力法等。用代数法解算复杂网路的原理是依据风量平衡定律、风压平衡定律、风阻平衡定律、阻力

(12) 生成树 (Spanning Tree): 一个连通图的生成树是它的极小连通子图, 在 n 个顶点的情形下, 有 $n-1$ 条边。但有向图则可能得到它的由若干有向树组成的生成森林。

2.1.2 矿井通风网络和它在计算机中的存储

对任一矿井通风系统, 如果不考虑各风流交汇点和通风巷道的位置、长度、形状及断面大小等情况, 仅以单线表示各交汇点与风流的联结关系, 那么, 这种只用点的集合和线的集合来表示的通风系统图以及各线所联系得通风参数所组成的系统称为矿井通风网络。

矿井通风系统图是用点的集合和线的集合表示其图图形的, 计为 $G=(V, E)$ 。其中 V 为结点 (即分流交汇点) 的集合, 它所包含节点数为 $|V|=m$; 而 E 为边 (对应通风巷道) 的集合, 它所含的边数为 $|E|=n$ 。即为 $V=\{v_1, v_2, \dots, v_m\}$, $E=\{e_1, e_2, \dots, e_n\}$ 。若不考虑各节点的位置, 按通风巷道与各交汇点之间的联结关系, 连接边 $e_k=(v_i, v_j)$, 而且矿井通风系统是一个有向的连通体系, 所以图 $G=(V, E)$ 是一个有向连通图, 其中的各有向边称为弧。若以通风巷道中的有关通风参数如风阻、风量、阻力等对应的边赋权, 则图 $G=(V, E)$ 就是一个有向连通赋权图。

图有两种基本的存储方式即邻接矩阵和邻接表, 本系统网络生成模块采用的是邻接矩阵存储通风网络图。邻接矩阵是表示顶点之间相邻关系的矩阵。设 $G=(V, E)$ 是具有 m 个顶点的网络图, 顶点序号依次为 $1, 2, \dots, m$, 则 G 的邻接矩阵是具有如下定义的 m 阶方阵 A :

$$A[i, j]=\begin{cases} w_{i,j} & \text{若 } (v_i, v_j) \in E(G) \\ 0 \text{ 或 } \infty & \text{其它} \end{cases} \quad (2-2)$$

2.1.3 矿井通风网络计算方法

多年以来, 国内外学者对矿井通风网络的自然分风计算, 提出了各种不同的方法^{[9][10]}如斯考德——恒斯雷法、牛顿法、节点压力法等。用代数法解算复杂网路的原理是依据风量平衡定律、风压平衡定律、风阻平衡定律、阻力

定律及已知的参数列出方程组，然后求解。由于未知数的个数众多，阻力定律又是二次方程，求解非常困难。使用计算机迭代试算法可以简化求解难度，解算复杂通风网路的迭代试算法可分为两大类：一类是回路法，即由假定回路内分支风向和风量开始，逐步修正，使之满足风压平衡定律；一类是节点法，由假定风流节点的压力值开始，逐步修正压力分布值，使之满足风量平衡定律。目前广泛应用的是回路法，特别是斯考德——恒斯雷法。

下面给出斯考德——恒斯雷法的数学模型描述。

设复杂风网的分支数为 N ，节点数为 J ，按图论有关树的理论，该风网的独立回路数为 $M=N-J+1$ 。在已知风网各分支的风阻，风机特性和自然风压的条件下，网络各分支的自然分配风量可由以下三个方程来决定：

(1) 节点风量平衡方程

在通风网络中，当风流为不可压缩流时，流入任一节点的风量等于流出的风量，

即：

$$\sum_{i=1}^n Q_i = 0 \quad (2-3)$$

式中： Q_i ——第 i 条支路的风量，流入节点的风量取正，流出的取负；

n ——相应节点所汇集的支路数。

上式表示流入及流出某节点（或闭合回路）的各分支风量的代数和等于零。

(2) 回路风压平衡方程

$$\sum_{j=1}^N a_{ij} R_j |Q_j| Q_j - P_i - F_i(Q_i) = 0 \quad (i=1, 2, 3 \dots M) \quad (2-4)$$

式中： Q_j ， R_j — j 分支的风量和风阻， m^3/s ， Ns^2/m^5 ；

P_i — i 回路的自然风压，Pa；

$F_i(Q_i)$ — i 号风机的风压，Pa；

a_{ij} ——风流方向的符号函数；

$a_{ij}=1$ ， j 分支包括在 i 回路内并与回路方向相同，

$a_{ij}=-1$ ， j 分支包括在 i 回路内并与回路方向相反，

$a_{ij}=0$ ， j 分支不包括在 i 回路内。

(3) 通风阻力特征方程

$$h_i = R_i Q_i^2 \quad (2-5)$$

式中: h_i —风网中的 i 条风路的风压或阻力, Pa;

R_i —第 i 条风路的风阻, Ns^2/m^8 ;

Q_i —第 i 条风路的风量, m^3/s 。

由于一个风网共有 M 个独立回路, 故有 M 个回路方程, 与节点方程一起共有 $(N-J+1) + (J-1) = N$ 个方程。可解出 N 个分支的风量, 有定解。对于大型网络, 这 N 个方程组成的方程组是个大型的非线性方程组, 一般可用线性化方法按泰勒公式展开并略去高阶项, 再经化简整理, 最后可得出:

$$\Delta Q_i = -\frac{\sum_{j=1}^N a_{ij} R_j |Q_j| Q_j - P_i - F_i}{2 \sum_{j=1}^N a_{ij}^2 R_j |Q_j| - F_i'} \quad (i=1, 2, 3 \dots M) \quad (2-6)$$

式中: F_i —风机风压, Pa;

F_i' —风机特性曲线的斜率。

对于上述方程, 假如 Q_j ($j=1, 2, 3 \dots$) 已赋某一初值, 则可满足方程(2-3), 但一般不能满足方程(2-4), 需要用 $a_{ij} \Delta Q_i$ 对 Q_j 进行反复修正, 使回路方程也逐渐满足。当相邻两次修正所得的风量的差值满足精度要求时, 就认为整个网络达到平衡。

为了提高迭代过程的收敛速度, 在每次迭代过程中不是等到把所有回路风量修正值全部求出之后, 再逐个修正各分支的风量, 而是求出一个回路的风量修正值后, 立即对构成本回路的分支的风量及时地给以修正, 并在计算后面回路风量修正值时, 均采用已经修正过的风量^[6]。

2.2 专家系统

专家系统是人工智能的一个重要分支。自 1968 年费根鲍姆等人研制成功第一个专家系统 DENDRAL 以来, 专家系统技术已经获得了迅速发展, 广泛应用于医疗诊断、图像处理、石油化工、地质勘探、金融决策、实时监控、

(3) 通风阻力特征方程

$$h_i = R_i Q_i^2 \quad (2-5)$$

式中: h_i —风网中的 i 条风路的风压或阻力, Pa;

R_i —第 i 条风路的风阻, Ns^2/m^8 ;

Q_i —第 i 条风路的风量, m^3/s 。

由于一个风网共有 M 个独立回路, 故有 M 个回路方程, 与节点方程一起共有 $(N-J+1) + (J-1) = N$ 个方程。可解出 N 个分支的风量, 有定解。对于大型网络, 这 N 个方程组成的方程组是个大型的非线性方程组, 一般可用线性化方法按泰勒公式展开并略去高阶项, 再经化简整理, 最后可得出:

$$\Delta Q_i = -\frac{\sum_{j=1}^N a_{ij} R_j |Q_j| Q_j - P_i - F_i}{2 \sum_{j=1}^N a_{ij}^2 R_j |Q_j| - F_i'} \quad (i=1, 2, 3 \dots M) \quad (2-6)$$

式中: F_i —风机风压, Pa;

F_i' —风机特性曲线的斜率。

对于上述方程, 假如 Q_j ($j=1, 2, 3 \dots$) 已赋某一初值, 则可满足方程(2-3), 但一般不能满足方程(2-4), 需要用 $a_{ij} \Delta Q_i$ 对 Q_j 进行反复修正, 使回路方程也逐渐满足。当相邻两次修正所得的风量的差值满足精度要求时, 就认为整个网络达到平衡。

为了提高迭代过程的收敛速度, 在每次迭代过程中不是等到把所有回路风量修正值全部求出之后, 再逐个修正各分支的风量, 而是求出一个回路的风量修正值后, 立即对构成本回路的分支的风量及时地给以修正, 并在计算后面回路风量修正值时, 均采用已经修正过的风量^[6]。

2.2 专家系统

专家系统是人工智能的一个重要分支。自 1968 年费根鲍姆等人研制成功第一个专家系统 DENDRAL 以来, 专家系统技术已经获得了迅速发展, 广泛应用于医疗诊断、图像处理、石油化工、地质勘探、金融决策、实时监控、

分子遗传工程、教学、军事等多种领域中，产生了巨大的社会效益及经济效益，同时也促进了人工智能基本理论和基本技术的研究与发展。目前，它已成为人工智能中一个最活跃且最有成效的研究领域^{[8][9]}。

2.2.1 专家系统基本原理

1. 专家系统

迄今为止，关于专家系统还没有一个公认严格定义，一般认为：

- (1) 它是一个智能程序系统；
- (2) 它具有相关领域内大量的专家知识；
- (3) 它能应用人工智能技术模拟人类专家求解问题的思维过程进行推理，解决相关领域内的困难问题，并且达到领域专家的水平。

将以上几点概括起来，可以说专家系统就是一种在相关领域中具有专家水平解决问题能力的智能程序系统，它能运用领域专家多年积累的经验与专门知识，模拟人类专家的思维过程，求解需要专家才能解决的困难问题^[8]。

2. 专家系统的一般结构

专家系统一般都包括人机接口、推理机、知识库及其管理系统、数据库及其管理系统、知识获取机构和解释机构这六个部分^[9]，如图 2.1。

分子遗传工程、教学、军事等多种领域中，产生了巨大的社会效益及经济效益，同时也促进了人工智能基本理论和基本技术的研究与发展。目前，它已成为人工智能中一个最活跃且最有成效的研究领域^{[8][9]}。

2.2.1 专家系统基本原理

1. 专家系统

迄今为止，关于专家系统还没有一个公认严格定义，一般认为：

- (1) 它是一个智能程序系统；
- (2) 它具有相关领域内大量的专家知识；
- (3) 它能应用人工智能技术模拟人类专家求解问题的思维过程进行推理，解决相关领域内的困难问题，并且达到领域专家的水平。

将以上几点概括起来，可以说专家系统就是一种在相关领域中具有专家水平解决问题能力的智能程序系统，它能运用领域专家多年积累的经验与专门知识，模拟人类专家的思维过程，求解需要专家才能解决的困难问题^[8]。

2. 专家系统的一般结构

专家系统一般都包括人机接口、推理机、知识库及其管理系统、数据库及其管理系统、知识获取机构和解释机构这六个部分^[9]，如图 2.1。

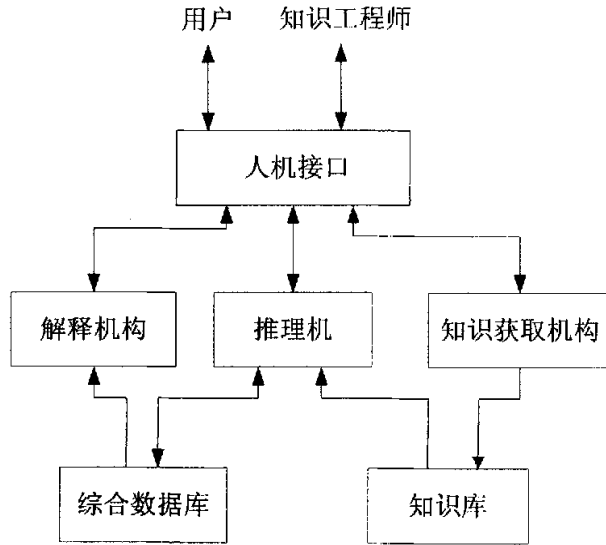


图 2.1 专家系统的结构

专家系统的各部分模块的阐述如下。

知识库:在这里也就是规则库,是专家系统的核心,负责存储和管理专家系统中的知识,包括相关领域中的公认知识、领域专家的经验知识和策略性或控制性知识。

推理机:推理就是一组计算机程序,主要功能是控制整个系统,决定如何选用知识库中的知识,对用户提供的证据进行推理,最终对用户提出的问题做出回答或完成特定的动作。

综合数据库:综合数据库存放专家系统中反映系统当前状态的事实数据,包括输入的已知事实以及推理过程中得到的中间结果等,它们是系统操作的对象。综合数据库中数据的表示和组织,应与知识库中知识的表示和组织具有相容性,以使推理机能方便地使用知识库中的知识和综合数据库中的事实对问题求解。

人机界面:人机界面是人机交互程序。人机界面负责把用户输入的信息转化成系统内规范化的表示形式,并把这些内部表示交给相应的模块去处理。

系统的输出信息也由人机界面转换成用户易于理解的外部表示形式显示给用户。

知识获取模块:该模块负责将知识转化为计算机可利用的形式送入知识库。知识获取模块使领域专家或用户可以修改知识库而不必了解知识库的组织结构、知识的具体表示方法等实现上的细节问题,提高了系统的可扩充性和灵活性。

解释模块:能够对自己的行为做出解释,由一组程序组成,它能跟踪并记录推理全过程。

2.2.2 专家系统的一般特点与优点

1. 专家系统的特点

人们对专家系统具有什么样的本质特征,有着各种各样的说法,存在着争议。不列颠计算机学会的学者提出:专家系统是被认为在计算机内根据专家的经验建立的基于知识的系统,这个系统提出智能的建议或对处理功能做出智能的决策。一个附加的特性是,系统在受到要求时有能力给咨询者能直接理解的方式证明系统的推理过程。一般说来,一个专家系统具有以下几个显著特点^{[9][10]}:

(1) 专家系统被局限于特定的专家知识的范围内。

(2) 启发性:专家系统能运用专家的知识与经验进行推理、判断和决策。世界上的大部分工作和知识都是非数学性的,只有一小部分人类活动是以数学公式为核心的(约占8%)。即使是化学和物理学科,大部分也是靠推理进行思考的;对于生物学、大部分医学和全部法律,情况也是这样。企业管理的思考几乎全靠符号推理,而不是数值计算。

(3) 专家系统可对不确定的数据进行推理。

(4) 透明性:专家系统能够解释本身的推理过程和回答用户提出的问题,以便让用户了解推理过程,提高对专家系统的信赖度;例如,一个医疗诊断专家系统诊断某病人患有肺炎,而且必需用某种抗生素治疗,那么,这一专家系统将会向病人解释为什么他患有肺炎,而且必须用某种抗生素治疗,

系统的输出信息也由人机界面转换成用户易于理解的外部表示形式显示给用户。

知识获取模块:该模块负责将知识转化为计算机可利用的形式送入知识库。知识获取模块使领域专家或用户可以修改知识库而不必了解知识库的组织结构、知识的具体表示方法等实现上的细节问题,提高了系统的可扩充性和灵活性。

解释模块:能够对自己的行为做出解释,由一组程序组成,它能跟踪并记录推理全过程。

2.2.2 专家系统的一般特点与优点

1. 专家系统的特点

人们对专家系统具有什么样的本质特征,有着各种各样的说法,存在着争议。不列颠计算机学会的学者提出:专家系统是被认为在计算机内根据专家的经验建立的基于知识的系统,这个系统提出智能的建议或对处理功能做出智能的决策。一个附加的特性是,系统在受到要求时有能力给咨询者能直接理解的方式证明系统的推理过程。一般说来,一个专家系统具有以下几个显著特点^{[9][10]}:

(1) 专家系统被局限于特定的专家知识的范围内。

(2) 启发性:专家系统能运用专家的知识与经验进行推理、判断和决策。世界上的大部分工作和知识都是非数学性的,只有一小部分人类活动是以数学公式为核心的(约占8%)。即使是化学和物理学科,大部分也是靠推理进行思考的;对于生物学、大部分医学和全部法律,情况也是这样。企业管理的思考几乎全靠符号推理,而不是数值计算。

(3) 专家系统可对不确定的数据进行推理。

(4) 透明性:专家系统能够解释本身的推理过程和回答用户提出的问题,以便让用户了解推理过程,提高对专家系统的信赖度;例如,一个医疗诊断专家系统诊断某病人患有肺炎,而且必需用某种抗生素治疗,那么,这一专家系统将会向病人解释为什么他患有肺炎,而且必须用某种抗生素治疗,

像一位医疗专家对病人详细解释病情一样。

(5) 事实与推理机构是完全分开的(知识没有固定的编码在推理过程中)。

(6) 灵活性:专家系统能不断的增长知识,修改原有知识,不断更新专家系统。

2. 专家系统的优点

专家系统除了与一般程序相比具有优越性外,专家系统也具有一般系统及人类专家无法比拟的优点^{[9][11]}:

(1) 专家系统具有很多人类专家无法比拟的优点,它能高效率、准确、周到、迅速和不知疲倦的工作;

(2) 专家系统解决问题时不受周围环境的影响,也不会遗漏忘记;

(3) 专家系统解决问题不受时间和空间的限制,以便能推广珍贵和稀缺的专家知识和经验;

(4) 专家系统能够汇集多领域专家的知识和经验以使他们协作解决重大问题的能力,它拥有更渊博的知识、更丰富的经验、更强的工作能力;

(5) 专家系统的研制与应用,具有巨大的经济效益和社会效益。

2.2.3 专家系统与常规系统的区别

(1) 常规的计算机程序是对数据结构以及作用于数据结构的确定性算法的表述,

常规程序=数据结构+算法

(2) 而专家系统是通过运用知识进行推理,力求在问题领域内推导出满意的解答,

专家系统=知识+推理

(3) 常规程序将关于问题求解的知识隐含于程序中,而专家系统则将应用领域中关于问题求解的知识单独地组成一个知识库。也就是说,常规程序将其知识组织为两级,即数据级和程序级,而专家系统则将其知识组织成三级,即数据级、知识库级和控制级。

像一位医疗专家对病人详细解释病情一样。

(5) 事实与推理机构是完全分开的(知识没有固定的编码在推理过程中)。

(6) 灵活性:专家系统能不断的增长知识,修改原有知识,不断更新专家系统。

2. 专家系统的优点

专家系统除了与一般程序相比具有优越性外,专家系统也具有一般系统及人类专家无法比拟的优点^{[9][11]}:

(1) 专家系统具有很多人类专家无法比拟的优点,它能高效率、准确、周到、迅速和不知疲倦的工作;

(2) 专家系统解决问题时不受周围环境的影响,也不会遗漏忘记;

(3) 专家系统解决问题不受时间和空间的限制,以便能推广珍贵和稀缺的专家知识和经验;

(4) 专家系统能够汇集多领域专家的知识和经验以使他们协作解决重大问题的能力,它拥有更渊博的知识、更丰富的经验、更强的工作能力;

(5) 专家系统的研制与应用,具有巨大的经济效益和社会效益。

2.2.3 专家系统与常规系统的区别

(1) 常规的计算机程序是对数据结构以及作用于数据结构的确定性算法的表述,

常规程序=数据结构+算法

(2) 而专家系统是通过运用知识进行推理,力求在问题领域内推导出满意的解答,

专家系统=知识+推理

(3) 常规程序将关于问题求解的知识隐含于程序中,而专家系统则将应用领域中关于问题求解的知识单独地组成一个知识库。也就是说,常规程序将其知识组织为两级,即数据级和程序级,而专家系统则将其知识组织成三级,即数据级、知识库级和控制级。

(4) 常规程序一般是通过查找或计算来求取问题的答案，基本上是面向数值计算和数据处理的，而且在问题的求解过程中先做什么及后做什么都是由程序规定的；而专家系统是通过推理来求取问题的答案或证明某个假设，本质上是面向符号处理的，其推理过程随着情况的变化而变化，具有不确定性及灵活性。

(5) 常规程序处理的数据多是精确的，对数据的检索是基于模式的布尔匹配；而专家系统处理的数据及知识大多数是不精确的、模糊的，知识的模式匹配也大多是不精确的。

(6) 常规程序一般不具有解释功能，而专家系统一般具有解释功能，可对自己的行为做出解释。

(7) 在常规程序结构中，输入信息经过特定领域代码的处理直接得到输出结果，而在专家系统程序结构中，对特定问题数据的处理必须得到知识库中相关知识的支持才能推出结论，常规程序结构与专家系统结构的对比见图 2.2 所示。

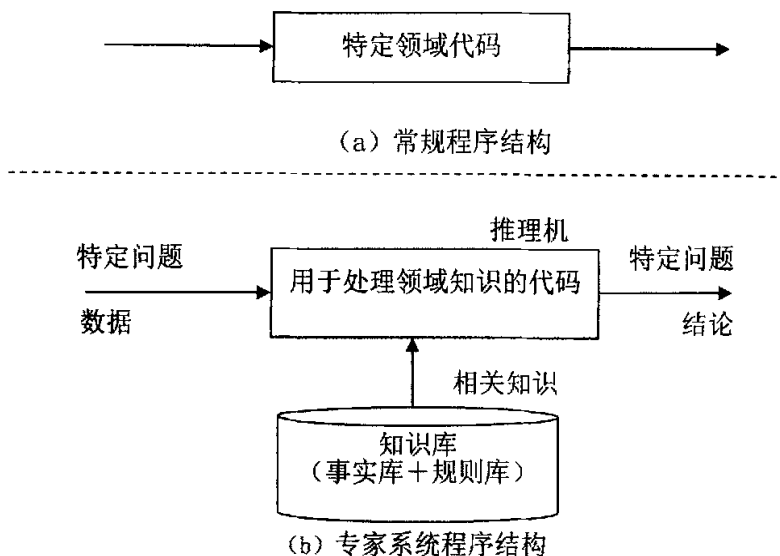


图 2.2 常规程序结构与专家系统结构的对比

2.3 本章小结

本章简单介绍了矿井通风网络解算的数学基础——图论，介绍了矿井通风网络中风流流动的基本规律和通风网络在计算机的存储，以及矿井通风网络解算的斯考德——恒斯雷法的数学模型。

专家系统的基本要求就是希望系统像专家一样的工作，完成专家才能完成的任务。专家系统相对于一般系统来说具有它自己的特点，并且和传统程序存在着明显的区别，这在本章都作了介绍。同时，还主要介绍了专家系统的组成以及各部分的功能、专家系统的优点。

第 3 章 知识库与推理机的研究

3.1 概述

知识表示是整个专家系统的首要问题，知识库的构建与推理机的设计和知识表示方法密切相关。专家系统的研究致力于在具体的专门领域内建立高性能的程序，其实质就是将与领域问题求解有关的知识有机地结合到程序设计中，使程序能够像人类专家一样进行推理、学习、解释、实现问题求解。专家系统的研究和设计着重于知识处理，包括知识的获取、表示和运用三个核心环节。知识表示主要研究用什么样的方法将求解问题所需要的知识存储在计算机中，开发灵活操作这些知识的推理机，使知识的表示和运用知识的推理控制相融合，便于计算机处理^[9]。

推理是根据一定的原则(公理或规则)从已知的事实(或判断)推出新的事实(或另外的判断)的思维过程，其中推理所依据的事实叫做前提(或条件)，前提所推出的新事实叫做结论。推理是人们无时无刻都在使用的方法，不论是学习和科学研究，还是日常生活都在运用着它。专家系统求解问题的基本原理在于将问题求解转化为状态空间的搜索问题，用求解问题的已知事实构成初始状态，问题的最终解答构成目标状态，初始状态与目标状态之间充满不同层次的子目标状态。在专家系统中，推理以知识库中的已有知识为基础，因此又叫基于知识的推理^[12]。所谓基于知识的推理，是指在计算机或智能机中，在知识表达的基础上，进行机器思维，求解问题，实现知识推理的智能操作过程。知识的选择称为控制策略，它是控制推理的智能操作过程，控制推理如何进行下去以及在何种情况下采用何种推理方式的一套控制方法。

在专家系统中，推理以知识库中的已有知识为根据，是一种基于知识的推理，基于知识推理的计算机实现就构成了推理机。知识库和推理机是专家系统的核心部分。

3.2 知识的表示

知识表示方法既是一个独立的课题，又与推理的方法有着密切的关系。主要有下面几种主要的方法：

(1) 逻辑表示法:表示一个知识库的主要方法是采用一阶谓词逻辑。此时的知识库可以看成是一组逻辑公式的集合。逻辑表示模式的优点是记号简单，描述易于理解；具有很好地理解形式语言的能力；从已知事实推导出新事实的过程能机械地使用自动定理证明技术。缺点是难以表示过程性和启发式知识；由于缺少结构上统一的规则，致使大型知识库难以管理；对证明过程进行操作的能力差，当事实的数量很大时，鉴于证明过程的每一步都需使用规则，而有可能导致组合爆炸^[9]。

(2) 语义网络:语义网络是通过包含结点和边(连结结点之间的弧)的有向图来描述对象、事件、概念以及动作的方法。语义网络可用在很复杂的分类学推理领域以及在那些需要表示对象、事件、场景及行为的特性及它们之间关系的领域中。它的优点是能简明准确地表示出重要的关系；有关对象或概念的相关事实可由它们直接相连的结点推导出来，而不必遍历一个庞大的数据库；容易做出有关继承分层的演绎推论。缺点是通过网络进行操作得到的推理并非都能有效；没有标准术语或约定解释语义，只能通过语义网的程序慢慢推导出；需要强有力的组织原则指导搜索，否则将陷入无穷支路；很难表示非物理连接的布尔关系^{[9][13]}。

(3) 框架:框架是一个复杂的数据结构，用以表示重复出现的原型事件。框架为每个对象和关系提供槽(或空位)，连接到每个框架的信息有:如何使用该框架；发生意外情况怎么办；缺省槽的值；附加必要的过程^{[10][14]}。此时的知识库是一个有组织的框架的集合。框架易于进行预期驱动进程的处理；在一个给定的环境下能决定本身的应用能力，或选择其它适当的框架；以一种集中注意力并且易于回顾、易于推理的方法组织知识。其缺点是许多真实情况在相当程度上脱离原型而且不容易归纳新的情况。

(4) 产生式规则:“产生式”这一术语是由美国数学家波斯特(E. Post)在1943年首先提出来的，他根据串替代规则提出了一种称为波斯特机的计算模型，模型中的每一条规则称为一个产生式^[11]。在此之后，几经修改与充实，

②一个综合数据库。它是一个用于存放问题求解过程中各种当前信息的数据结构，例如问题的初始状态、推理的中间结果及最终结论。当规则库中某条产生式的前提可与综合数据库中的某些已知事实匹配时，该产生式就被激活，并将用它推出的结论放入综合数据库中，作为后面推理的已知事实。显然，综合数据库的内容是在不断变化的，是动态的。

③一个控制系统。控制系统又称为推理机制，由一组程序组成，负责整个产生式系统的运行，实现对问题的求解。其功能主要有：第一，按一定的策略从规则库中选择规则与综合数据库的已知事实进行匹配。所谓匹配是指将规则的前提条件与综合数据库中的已知事实进行比较，如果两者一致，或者近似一致且满足预先规定的条件，则称匹配成功，相应的规则可以被使用；否则称为匹配不成功，相应规则不可用于当前的推理。第二，匹配成功的规则可能不止一条，这称为发生了冲突。此时，推理机必须调用相应的解决冲突策略进行消解，以便从中选出一条执行。第三，在执行某一条规则时，如果该规则的右部是一个或多个结论，则将这些结论加入到综合数据库中；如果规则的右部是一个或多个操作，则执行这些操作。

图 3.1 中的三个部分与产生式系统三要素相对应：(a)数据：数据的对象是实例。这些实例既可以描述与应用领域有关的物理对象或事实，又可以描述与问题解决有关的概念对象。数据存储于数据库中。(b)规则：规则用来描述应用领域人类专家的知识。这是知识系统的主体。规则存储在规则库中。(c)推理机：推理机实质上是一个规则执行器。它决定哪些规则与数据库中的数据匹配，以选择一个规则去执行。

(2) 产生式规则表示法

①产生式规则描述

产生式规则常用于表示具有因果关系的知识，其基本形式是

$$\text{IF } P \text{ THEN } Q$$

其中，P 代表一组前提或状态，Q 代表若干结论或动作，其含义是如果前提 P 得以满足，即为“真”，则可得出结论 Q 或 Q 所规定的动作。

②产生式规则的特点

- 自然性

产生式表示法用“如果……，则……”的形式表示知识，这种表示形式

与人类的判断性知识基本一致，既直观自然，又便于进行推理。

- 模块性

产生式规则是规则库中最基本的知识单元，各规则之间只能通过综合数据库发生联系，而不能相互调用，从而增加了规则的模块性，有利于知识的增加、删除、修改和扩充。

- 有效性

产生式知识表示法既可以表示确定性知识，又可以表示不确定知识，既有利于表示启发性知识，又有利于表示过程性知识。

- 一致性

产生式系统中所有的规则都具有相同的格式，并且综合数据库可被所有的规则访问，不存在只属于某些规则的局部数据库，因此规则库中的规则可以统一处理。

③产生式规则的主要缺点

- 效率较低

在产生式表示中，各规则之间的联系必须以综合数据库为媒介，并且其求解过程是一种反复进行的“匹配——冲突消解——执行”的过程。即：先用规则前提与综合数据库中的已知事实进行匹配，再从规则中选择可用规则，当可用规则是多条时还需要按一定策略进行“冲突消解”，然后才能执行选中的规则。这样的执行方式将导致执行的低效率。

- 不能表示结构性知识

由于产生式表示中知识具有一致格式，且规则之间不能相互调用，因此那种具有结构关系或层次关系的知识用产生式是很难将其以自然的方式来表示的。实际上，产生式系统除了作为一种独立的知识表示方式以外，还常常需要和其它方法配合起来使用。

3.3 知识的检测与求精

知识的一致性和完整性是影响专家系统性能的重要因素。在文献^[9]中对其概念、检测及处理方法进行了讨论。

与人类的判断性知识基本一致，既直观自然，又便于进行推理。

- 模块性

产生式规则是规则库中最基本的知识单元，各规则之间只能通过综合数据库发生联系，而不能相互调用，从而增加了规则的模块性，有利于知识的增加、删除、修改和扩充。

- 有效性

产生式知识表示法既可以表示确定性知识，又可以表示不确定知识，既有利于表示启发性知识，又有利于表示过程性知识。

- 一致性

产生式系统中所有的规则都具有相同的格式，并且综合数据库可被所有的规则访问，不存在只属于某些规则的局部数据库，因此规则库中的规则可以统一处理。

③产生式规则的主要缺点

- 效率较低

在产生式表示中，各规则之间的联系必须以综合数据库为媒介，并且其求解过程是一种反复进行的“匹配——冲突消解——执行”的过程。即：先用规则前提与综合数据库中的已知事实进行匹配，再从规则中选择可用规则，当可用规则是多条时还需要按一定策略进行“冲突消解”，然后才能执行选中的规则。这样的执行方式将导致执行的低效率。

- 不能表示结构性知识

由于产生式表示中知识具有一致格式，且规则之间不能相互调用，因此那种具有结构关系或层次关系的知识用产生式是很难将其以自然的方式来表示的。实际上，产生式系统除了作为一种独立的知识表示方式以外，还常常需要和其它方法配合起来使用。

3.3 知识的检测与求精

知识的一致性和完整性是影响专家系统性能的重要因素。在文献^[9]中对其概念、检测及处理方法进行了讨论。

3.3.1 知识的一致性和完整性

知识库的建立过程是知识经过一系列变换进入计算机系统的过程,在这个过程中存在着各种各样导致知识不健全的因素。例如,领域专家提供的知识中存在某些不一致、不完整,甚至错误的知识。由于专家系统是以专家知识为基础,因而专家知识的任何不一致、不完整必然影响到知识库的一致性和完整性。

(1) 知识工程师未能准确、全面地理解领域专家的意图,使得所形成的知识条款隐含着种种错误,影响到知识的一致性及完整性。

(2) 采用的知识表示模式不适当,不能把领域知识准确地表示出来。

(3) 对知识库进行增、删、改时没有充分考虑到可能产生的影响,以致在进行了这些操作之后使得知识库出现了不完备的情况。特别是在知识库建立之后,由于知识间存在着千丝万缕的复杂联系,因而对它的任何改动都可能产生意想不到的后果。

3.3.2 知识不一致和不完整的表现方面

由于上面讲到的原因,知识库中经常会出现这样或那样的问题,主要表现在知识冗余、矛盾、从属、环路、不完整等方面。下面是以产生式表示法为例说明各种问题的表现形式^[9]。

(1) 知识冗余

所谓知识冗余是指知识库中存在多余的知识或者存在多余的约束条件。分为等价规则、冗余规则链、冗余条件三种。

(2) 矛盾

如果两条产生式规则或规则链在相同的条件下得到的结论是互斥的,或者它们虽有相同的结论,但规则的强度不同,则称它们是矛盾的。对于矛盾规则或矛盾规则链,不能让它们共处于同一知识库中,必须从中舍弃一个。至于舍弃哪一个,需征求领域专家的意见。

(3) 从属

如果规则 r_1 与 r_2 有相同的结论,但 r_1 比 r_2 要求更多的约束条件,则

3.3.1 知识的一致性和完整性

知识库的建立过程是知识经过一系列变换进入计算机系统的过程,在这个过程中存在着各种各样导致知识不健全的因素。例如,领域专家提供的知识中存在某些不一致、不完整,甚至错误的知识。由于专家系统是以专家知识为基础,因而专家知识的任何不一致、不完整必然影响到知识库的一致性和完整性。

(1) 知识工程师未能准确、全面地理解领域专家的意图,使得所形成的知识条款隐含着种种错误,影响到知识的一致性及完整性。

(2) 采用的知识表示模式不适当,不能把领域知识准确地表示出来。

(3) 对知识库进行增、删、改时没有充分考虑到可能产生的影响,以致在进行了这些操作之后使得知识库出现了不完备的情况。特别是在知识库建立之后,由于知识间存在着千丝万缕的复杂联系,因而对它的任何改动都可能产生意想不到的后果。

3.3.2 知识不一致和不完整的表现方面

由于上面讲到的原因,知识库中经常会出现这样或那样的问题,主要表现在知识冗余、矛盾、从属、环路、不完整等方面。下面是以产生式表示法为例说明各种问题的表现形式^[9]。

(1) 知识冗余

所谓知识冗余是指知识库中存在多余的知识或者存在多余的约束条件。分为等价规则、冗余规则链、冗余条件三种。

(2) 矛盾

如果两条产生式规则或规则链在相同的条件下得到的结论是互斥的,或者它们虽有相同的结论,但规则的强度不同,则称它们是矛盾的。对于矛盾规则或矛盾规则链,不能让它们共处于同一知识库中,必须从中舍弃一个。至于舍弃哪一个,需征求领域专家的意见。

(3) 从属

如果规则 r_1 与 r_2 有相同的结论,但 r_1 比 r_2 要求更多的约束条件,则

称 r1 是 r2 的从属规则。

(4) 环路

当一条规则形成一条循环链时，称它们构成了一个环路。例如，设有如下的一组产生式规则：

r1: IF P THEN Q

r2: IF Q THEN R

r3: IF R THEN S

r4: IF S THEN P

对这四条规则无论先执行哪一条，最终都回到了出发点，即它们之间出现了环路。环路有可能使推理陷入死循环，应引起足够的重视。

当知识库中出现环路时，应征求领域专家的意见，修改或舍弃其中的一条规则，破坏形成环路的条件。

(5) 不完整

所谓不完整是指知识库中的知识不完全，不能满足预想定义的约束条件，即当存在应该推出某一结论时，却推不出这一结论，不能形成产生这一结论的推理链；或者虽能推出结论，但却是错误的。

当知识库的知识不完整时，需要通过知识求精不断改进、完善，使其能满足问题求解的需要。

3.3.3 基于经典逻辑的检测方法

为了保证知识库的正确性，需要做好对知识的检测，检测分为静态检测和动态检测两种。静态检测是指在知识输入之前由领域专家及知识工程师所做的检查工作；动态检测是指在输入过程中以及对知识库进行增、删、改时由系统所进行的检查。在系统运行过程中出现错误时也需要对知识库进行动态检测。

基于经典逻辑的检测方法在文献^[10]对它进行了理论上的探讨。

由于在这个系统中，只对逻辑表达式等价和循环进行算法的研究和设计，因此对它做比较详细的介绍，并介绍了一下其它与此相关的方法。如冗余、矛盾、从属规则的检测。

称 r_1 是 r_2 的从属规则。

(4) 环路

当一条规则形成一条循环链时，称它们构成了一个环路。例如，设有如下的一组产生式规则：

$r_1: \text{IF } P \text{ THEN } Q$

$r_2: \text{IF } Q \text{ THEN } R$

$r_3: \text{IF } R \text{ THEN } S$

$r_4: \text{IF } S \text{ THEN } P$

对这四条规则无论先执行哪一条，最终都回到了出发点，即它们之间出现了环路。环路有可能使推理陷入死循环，应引起足够的重视。

当知识库中出现环路时，应征求领域专家的意见，修改或舍弃其中的一条规则，破坏形成环路的条件。

(5) 不完整

所谓不完整是指知识库中的知识不完全，不能满足预想定义的约束条件，即当存在应该推出某一结论时，却推不出这一结论，不能形成产生这一结论的推理链；或者虽能推出结论，但却是错误的。

当知识库的知识不完整时，需要通过知识求精不断改进、完善，使其能满足问题求解的需要。

3.3.3 基于经典逻辑的检测方法

为了保证知识库的正确性，需要做好对知识的检测，检测分为静态检测和动态检测两种。静态检测是指在知识输入之前由领域专家及知识工程师所做的检查工作；动态检测是指在输入过程中以及对知识库进行增、删、改时由系统所进行的检查。在系统运行过程中出现错误时也需要对知识库进行动态检测。

基于经典逻辑的检测方法在文献^[10]对它进行了理论上的探讨。

由于在这个系统中，只对逻辑表达式等价和循环进行算法的研究和设计，因此对它做比较详细的介绍，并介绍了一下其它与此相关的方法。如冗余、矛盾、从属规则的检测。

对知识冗余、矛盾等的检测实际上是通过知识的相应部分进行比较实现的。例如对两条产生式规则的条件部分进行比较，看其是否等价，再对它们的结论部分进行比较，看它们是否一致等，因此，检测中的基本环节是检查两个逻辑表达式的等价性。下面首先讨论检测两个逻辑表达式等价性的方法，然后再在此基础上讨论冗余、矛盾等的检测。

(1) 逻辑表达式等价性的检测

产生式规则的条件部分是由原子命题 AND, OR, NOT 连接而成的逻辑表达式，但由于不同的表示形式可能会有相同的逻辑结果，这就使两个逻辑表达式等价性的检测增加了复杂性。例如对如下两个逻辑表达式：

$$\neg(P \vee Q) \quad (3-1)$$

$$\neg P \wedge \neg Q \quad (3-2)$$

它们虽然有不同的表示形式，但却是等价的。为解决这一问题，可在对它们进行检测之前做些变换，使它们都统一地变换为合取式，然后在再进行比较，例如对如下逻辑表达式：

$$P \vee (Q \wedge R) \quad (3-3)$$

把它变换为合取式后成为：

$$(P \vee Q) \wedge (P \vee R) \quad (3-4)$$

这样就可用它与另一个逻辑表达式的合取式进行比较。

设有如下两个合取式：

$$(P1 \vee Q1) \wedge (R1 \vee S1) \quad (3-5)$$

$$(P2 \vee Q2) \wedge (R2 \vee S2) \quad (3-6)$$

进行比较时，先用 $P1 \vee Q1$ 与 $P2 \vee Q2$ 进行比较，若两者等价，接着就用 $R1 \vee S1$ 与 $R2 \vee S2$ 进行比较，如果也等价，则上述两个合取式等价；如果 $P1 \vee Q1$ 与 $P2 \vee Q2$ 不等价，则还需要用 $P1 \vee Q1$ 与 $R2 \vee S2$ 进行比较，用 $R1 \vee S1$ 与 $P2 \vee Q2$ 进行比较。如果 $P1 \vee Q1$ 或者 $R1 \vee S1$ 在第二个合取式中找不到可与它等价的合取项，则上述两个合取式就不等价。

(2) 冗余的检测

运用逻辑表达式等价性的检测方法，就可方便地进行冗余的检测。它的算法虽然简单，但种类复杂，在文献^[9]中对它的方法有介绍。

(3) 矛盾规则及矛盾规则链的检测

矛盾规则链的检测与冗余规则链的检测方法类似。首先根据 IF-IF 表找出两条其条件部分等价的规则，并分别为它们建立一个推理图，然后遍历这两个推理图，如果发现有一对节点的结论部分是矛盾的，则从这一对节点分别到其根节点所形成的两条推理链是矛盾的。至于矛盾规则，它只是矛盾规则链的一种简单情况，即每条规则链中只要一条规则。

对于由规则强度不同所引起的矛盾，需要先检查两条规则是否等价，如果等价，再检查它们的规则强度是否相同，若不同，表明这两条规则是矛盾的。

(4) 从属规则的检测

为发现从属规则，首先要检查两条规则的结论是否等价，这可从 THEN-THEN 表得到。若两条规则的结论等价，再检查一条规则的条件部分是否为另一条规则条件部分的一部分，若是，表明存在一条从属规则，并且找到了它。

(5) 环路的检测

为了检测知识间是否存在环路，需要找到这样的规则，即其结论可与其它规则的条件等价。为此，可建立一张名为 IF-THEN 的二维表，表中存放每一条规则的条件与其它规则的结论相比较的结果。检测时搜索这张表，找出其结论可与其它规则的条件部分等价的规则，它有可能就是环路中的一条规则。然后从这条规则开始，根据 IF-THEN 表沿着规则链进行查找，直到出现如下两种情况之一时为止：

①在链中找到了一条规则，其结论与前面某条规则的条件等价，这说明找到了环路。

②沿规则链找不到其结论与前面某规则的条件等价的规则，且规则链结束，这说明该规则链没有形成环路。

若对每条规则链都进行上述检测，就可把知识库中的环路都找出来，并做相应的处理^[8]。

本系统在知识的获取模块设计了知识库的检测与求精算法，保证了知识库完整性和一致性。

3.4 推理机的功能描述

一般来说推理机包括调度器, 执行器和一致性协调器三部分。调度器控制整个推理流程, 使得推理可以按照一定的流程进行下去; 执行器执行调度器所选定的动作, 并且负责读取知识库中的知识和全局黑板中的信息; 一致性协调器保持推理中间结果的一致性, 在反向推理中用的最多。现今的大多数专家系统中都采用产生式规则集的知识表示, 基于这种知识表示的推理的工作流程如图 3.2 所示。

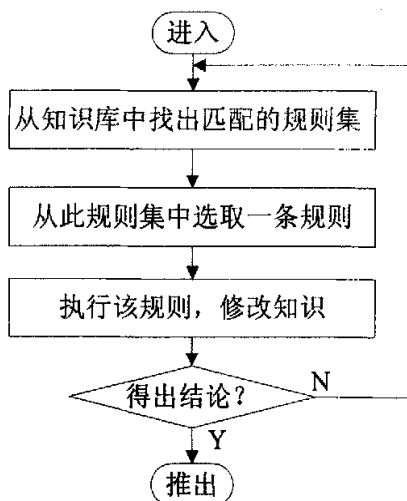


图 3.2 基于规则的推理机流程

3.5 推理的控制策略

推理过程是一个思维过程, 即求解问题的过程。问题求解的质量与效率不仅依赖于所采用的求解方法(如匹配方法、不确定的传递算法等), 而且还依赖于求解问题的策略, 即推理的控制策略。

据文献^[9]中的介绍, 推理的控制策略主要包括推理方向、搜索策略、冲突消解策略、求解策略及限制策略等。推理的方向用于确定推理的驱动方式, 分为正向推理、逆向推理、混合推理三种。无论按哪种方向推理, 一般都要

3.4 推理机的功能描述

一般来说推理机包括调度器, 执行器和一致性协调器三部分。调度器控制整个推理流程, 使得推理可以按照一定的流程进行下去; 执行器执行调度器所选定的动作, 并且负责读取知识库中的知识和全局黑板中的信息; 一致性协调器保持推理中间结果的一致性, 在反向推理中用的最多。现今的大多数专家系统中都采用产生式规则集的知识表示, 基于这种知识表示的推理的工作流程如图 3.2 所示。

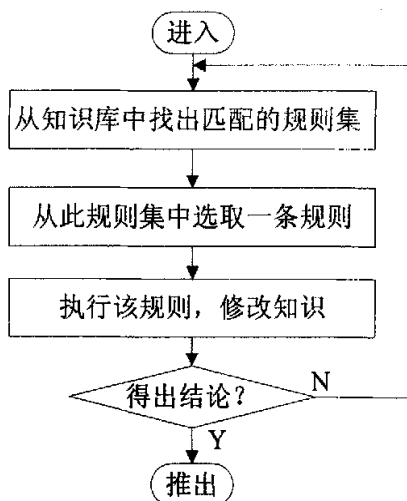


图 3.2 基于规则的推理机流程

3.5 推理的控制策略

推理过程是一个思维过程, 即求解问题的过程。问题求解的质量与效率不仅依赖于所采用的求解方法(如匹配方法、不确定的传递算法等), 而且还依赖于求解问题的策略, 即推理的控制策略。

据文献^[9]中的介绍, 推理的控制策略主要包括推理方向、搜索策略、冲突消解策略、求解策略及限制策略等。推理的方向用于确定推理的驱动方式, 分为正向推理、逆向推理、混合推理三种。无论按哪种方向推理, 一般都要

求系统具有一个存放知识的知识库，一个存放初始已知事实及问题状态的数据库和一个用于推理的推理机。

由于本系统使用的是正向推理方法，因此这里重点介绍正向推理，并对其它的控制策略只进行了简略的介绍。

(1) 正向推理

正向推理的主要思想是从问题所有可能的初始证据(事实)开始，通过匹配每条知识的前提识别出所有可用知识，形成一个可用知识集(也称为冲突集)，然后以某种冲突消解方式(有时也称知识排序方式)在冲突集中选取一条知识使用，这条知识的使用会得出新的事实。新的事实和原有事实又引起知识库中新的知识的匹配，从而继续问题的求解。直到求解达到某一状态，要么问题的结论以包含在所产生的事实中(有解)，要么冲突求解得不出可以使用的知识(无解或所有解已经给出不再有新的解)。所以正向推理算法思想的原理是以不断的“匹配——冲突消解——执行”的三阶段周期进行运转。基本的正向推理的过程可以用图 3.3 来说明。

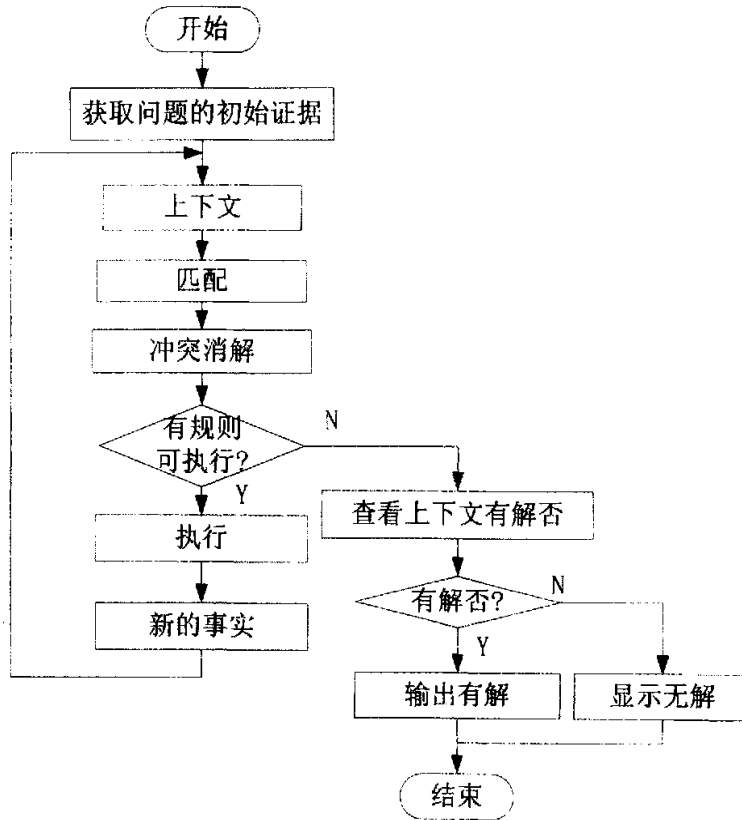


图 3.3 正向推理过程

算法 3-1:正向推理算法

开始:

```

do (//由前提链表中的事实, 判断哪些规则适用
    规则=规则链表头节点;
    while(规则!=null)
    {
        修改标志=false;
        if(该规则被检测过)
        {
    
```

取下一条规则;

```
continue;
```

```
}
```

根据取得的未被检测过的规则, 得到其结果;

将该规则设置为检测过;

```
if(该结果不在前提队列中)
```

```
{将结果插入前提队列尾部;
```

```
  修改标志=true;
```

```
  break;
```

```
}//end of while
```

```
//如果 while 循环结束且修改标志为 false,
```

```
//则所有规则都被检测过, 不再有新结论
```

```
if(修改标志==false)
```

```
{
```

```
  return 搜索结果;
```

```
}
```

```
}while (true);
```

用户输入当前可以满足的前提组, 从总规则集中寻找其前提部分可以被用户的前提满足的规则, 依次检测这些规则, 冲突消解策略如上所述, 将这些规则产生的结论并入前提组, 重新判断总规则中有哪些新规则可以得到满足, 如此循环, 直到不再有新的结论并入到前提组中为止, 这时, 最后得到的一个结论必定是推论的结果。

(2) 反向推理

反向推理是正向推理的逆过程, 它的基本实现是: 首先假设问题的结论为推理网络中的某个顶层节点, 放入假设堆栈, 然后从知识库中找出其结论部分能与假设相匹配的所有知识, 得出一个可用知识集, 从可用知识集中选取一条知识验证其前提部分^[16]。如果该条知识中的所有前提条件均为叶节点且能被用户证实或被上下文匹配, 则该知识验证成功, 从假设堆栈中弹出该知识的结论并放入上下文; 如果该知识中的某个前提条件被用户或上下文否定, 则该知识的结论部分不能被这条知识所验证, 可以返回可用知识集重新选择

一条可用知识；当所有可用知识都不能验证成功时，该结论从假设堆栈中删除；如果一条选择的可用知识中某些前提不是叶节点且又不能被上下文肯定或否定，则将这些前提压入假设堆栈，以上述相同的步骤验证这些前提。直到上下文中含有推理网络中顶层节点(有解)或所有顶层节点都测试过以后上下文中仍没有顶层节点(无解)。它使用的较多，主要特点是目标明确，推理快，算法实现如下：

算法 3-2:反向推理算法

开始:从规则队列中寻找出结论的新规则;

```

if(找到){
    while(该规则的前提部分的规则未测试完)
        { //以该规则的前提部分的属性作为结论，递归调用推理//函数;
            if(上面的推理不成立)
                {
                    return false; //结束，推理失败
                }
            //只要有一个前提不成立，那么整个结论都不成立，取前提//中的下一个属性;
        }
    //while 循环结束，意味着所有的前提成立，即结论成立
    return true;
}
else return false;
    
```

整个推理过程就像一个搜索过程，由于对规则作了严格的规定，因而，只需考虑那些所用到的规则，也就是查找到结论部分与推理的事实相符合的规则，就可以对推理过程做出解释。

(3) 混合推理

前向推理简单，易实现，但目的性不强，需要用启发式知识控制中间结论的选取；反向推理在选择目标时具有很大的盲目性，因此，反向推理比较时结论单一或直接提出结论要求证实的系统；双向推理具有前向和反向推理

的优点。

混合推理先根据已知的事实出发通过前向推理帮助提出假设，再用反向推理进一步寻找支持假设的证据。在双向推理过程中，直到得到结论之前，总是这样一前一后的反复进行推理。

3.6 冲突消解策略

在产生式系统中，若出现了如下情况就认为发生了冲突。

对正向推理而言，如果有多条产生式规则的前件都和已知事实匹配成功；或者有多组不同的已知事实都与同一条产生式规则的前件匹配成功；或者以上两种情况同时出现。

冲突消解的任务是解决冲突。对正向推理来说，它将决定选择哪一组已知事实来激活哪一条产生式规则，使它用于当前的推理，产生其后件指出的结论或执行相应的操作。

目前已有多种消解冲突的策略，其基本思想都是对知识进行排序，常用的有以下几种^[4]。

(1) 按针对性排序

选用针对性较强的产生式规则。因为它要求的条件较多，其结论一般更接近于目标，一旦得到满足，可缩短推理过程。

(2) 按已知事实的新鲜性排序

在产生式系统的推理过程中，每应用一条产生式规则就会得到一个或多个结论或者执行多个操作，数据库中就会增加新的事实。把数据库中后生成的事实称为新鲜的事实，即后生成的事实比先生成的事实有较大的新鲜性。

若一条规则被应用后生成了多条结论，则既可以认为这些结论有相同的新鲜性，也可以认为排在前面(或后面)的结论有较大的新鲜性，这根据情况决定。

(3) 根据领域问题的特点排序

某些领域问题，事先可知道它的某些特点，此时可根据这些特点把知识排成固定的顺序。例如：

- ① 当领域问题有固定的解题次序时，可按次序排列相应的知识，排在

的优点。

混合推理先根据已知的事实出发通过前向推理帮助提出假设，再用反向推理进一步寻找支持假设的证据。在双向推理过程中，直到得到结论之前，总是这样一前一后的反复进行推理。

3.6 冲突消解策略

在产生式系统中，若出现了如下情况就认为发生了冲突。

对正向推理而言，如果有多条产生式规则的前件都和已知事实匹配成功；或者有多组不同的已知事实都与同一条产生式规则的前件匹配成功；或者以上两种情况同时出现。

冲突消解的任务是解决冲突。对正向推理来说，它将决定选择哪一组已知事实来激活哪一条产生式规则，使它用于当前的推理，产生其后件指出的结论或执行相应的操作。

目前已有多种消解冲突的策略，其基本思想都是对知识进行排序，常用的有以下几种^[4]。

(1) 按针对性排序

选用针对性较强的产生式规则。因为它要求的条件较多，其结论一般更接近于目标，一旦得到满足，可缩短推理过程。

(2) 按已知事实的新鲜性排序

在产生式系统的推理过程中，每应用一条产生式规则就会得到一个或多个结论或者执行多个操作，数据库中就会增加新的事实。把数据库中后生成的事实称为新鲜的事实，即后生成的事实比先生成的事实有较大的新鲜性。

若一条规则被应用后生成了多条结论，则既可以认为这些结论有相同的新鲜性，也可以认为排在前面(或后面)的结论有较大的新鲜性，这根据情况决定。

(3) 根据领域问题的特点排序

某些领域问题，事先可知道它的某些特点，此时可根据这些特点把知识排成固定的顺序。例如：

① 当领域问题有固定的解题次序时，可按次序排列相应的知识，排在

前面的知识优先被应用。

② 当已知某些产生式规则被应用后会明显地有利于问题的求解时，就使这些产生式规则优先被应用。

(4) 按条件个数排序

如果有多条产生式规则生成的结论相同，则要求条件少的产生式规则被优先应用，因为要求条件少的规则匹配时花费的时间较少。

(5) 按上下文限制排序

把产生式规则按它们所描述的上下文分成若干组，在不同的条件下，只能从相应的组中选取有关的产生式规则，这样，就可以减少冲突的发生，而且由于搜索范围小，也提高了推理的效率。

在具体应用时，可对上述策略进行组合，目的是尽量减少冲突的发生，使推理有较快的速度和较高的效率。

本系统的冲突消解控制策略发生在规则链表生成过程中，根据领域问题的特点排序，采用的是数据排序方式，即把知识在知识库组织中的顺序决定优先级的次序。再生成动态数据时，首先赋予排序在前的规则高优先级，然后再赋予新的可产生同样结论的规则次优先级，依此类推。在推理机进行推理的过程中，则按照这个优先级别进行冲突消解。这样在设计和维护知识库的时候，就需要考虑规则的优先级别，把最佳解先录入知识数据库中。

3.7 本章小结

知识库是一个专家系统的关键部分之一，知识库的构建和知识表示密切相关。本章介绍了知识的表示方法，主要讨论了产生式表示方法的基本组成部分和产生式表示方法的特点及优缺点；知识的一致性和完整性是影响专家系统性能的重要因素，本章探讨了基于产生式表示的知识检测与求精的方法。

推理机是专家系统的推理执行部件，推理机制的优劣关系到推理机的效率的高低，它是系统成败的关键因素之一。本章介绍了推理机的功能，文中对推理机的推理策略进行了研究设计，基本的推理方法有正向推理、反向推理和混合推理三种。最后，本章对推理冲突消解策略也进行了研究，提出了解决该系统冲突消解策略。

前面的知识优先被应用。

② 当已知某些产生式规则被应用后会明显地有利于问题的求解时，就使这些产生式规则优先被应用。

(4) 按条件个数排序

如果有多条产生式规则生成的结论相同，则要求条件少的产生式规则被优先应用，因为要求条件少的规则匹配时花费的时间较少。

(5) 按上下文限制排序

把产生式规则按它们所描述的上下文分成若干组，在不同的条件下，只能从相应的组中选取有关的产生式规则，这样，就可以减少冲突的发生，而且由于搜索范围小，也提高了推理的效率。

在具体应用时，可对上述策略进行组合，目的是尽量减少冲突的发生，使推理有较快的速度和较高的效率。

本系统的冲突消解控制策略发生在规则链表生成过程中，根据领域问题的特点排序，采用的是数据排序方式，即把知识在知识库组织中的顺序决定优先级的次序。再生成动态数据时，首先赋予排序在前的规则高优先级，然后再赋予新的可产生同样结论的规则次优先级，依此类推。在推理机进行推理的过程中，则按照这个优先级别进行冲突消解。这样在设计和维护知识库的时候，就需要考虑规则的优先级别，把最佳解先录入知识数据库中。

3.7 本章小结

知识库是一个专家系统的关键部分之一，知识库的构建和知识表示密切相关。本章介绍了知识的表示方法，主要讨论了产生式表示方法的基本组成部分和产生式表示方法的特点及优缺点；知识的一致性和完整性是影响专家系统性能的重要因素，本章探讨了基于产生式表示的知识检测与求精的方法。

推理机是专家系统的推理执行部件，推理机制的优劣关系到推理机的效率的高低，它是系统成败的关键因素之一。本章介绍了推理机的功能，文中对推理机的推理策略进行了研究设计，基本的推理方法有正向推理、反向推理和混合推理三种。最后，本章对推理冲突消解策略也进行了研究，提出了解决该系统冲突消解策略。

第4章 通风网络解算与故障诊断系统的设计 与实现

前面已经对这个系统涉及到的知识进行了简略的介绍，本章对通风网络解算与诊断系统进行了具体的设计与实现，通风网络解算模块采用斯考德—恒斯雷法逐次近似算法和相关的图论算法；诊断模块采用基于规则的产生式知识表示方法，建立系统的知识库和推理机模型，在此基础上进行系统的设计与开发。

4.1 系统的概要设计

根据前面的知识库和推理机的构造原理，通风网络解算与故障诊断系统包括6个大部分，如图4.1所示。

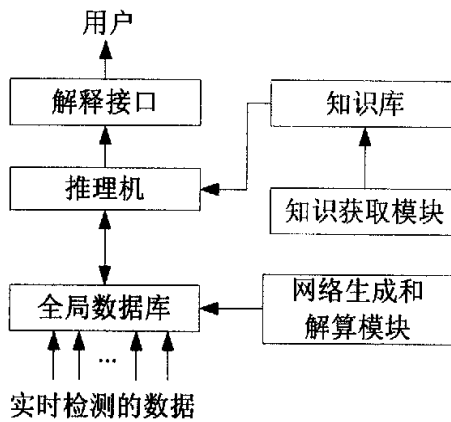


图 4.1 系统结构图

网络生成模块主要是将矿井通风网络的网络结构存储到数据库中。现实

第4章 通风网络解算与故障诊断系统的设计 与实现

前面已经对这个系统涉及到的知识进行了简略的介绍，本章对通风网络解算与诊断系统进行了具体的设计与实现，通风网络解算模块采用斯考德—恒斯雷法逐次近似算法和相关的图论算法；诊断模块采用基于规则的产生式知识表示方法，建立系统的知识库和推理机模型，在此基础上进行系统的设计与开发。

4.1 系统的概要设计

根据前面的知识库和推理机的构造原理，通风网络解算与故障诊断系统包括6个大部分，如图4.1所示。

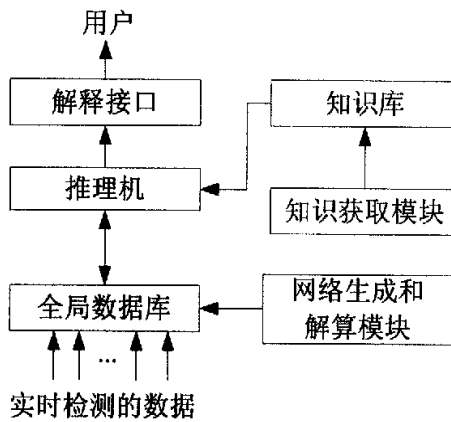


图 4.1 系统结构图

网络生成模块主要是将矿井通风网络的网络结构存储到数据库中。现实

中的矿井通风网络结构抽象成图论中的有向图数据结构，图的节点代表各巷道的交叉点，有向图中弧的属性值中包含巷道的风阻、扇风机的风压和斜率等，在计算机中采用邻接矩阵数据结构表示图的结构。网络解算采用斯考德——恒斯雷蒙逐次近似算法对生成的网络进行解算，计算出各个巷道中的风量值和风流方向，这可以作为网络风流方向和大小诊断的依据。

知识库获取模块主要完成规则库的生成，同时规则录入时需要对知识的一致性和完整性进行检测，确保规则库中没有逻辑表达式等价性的规则、没有循环的规则和无冗余的规则等。

全局数据库相当于黑板结构，用来记录系统推理过程中用到的控制信息、中间假设和中间结果以及结论的数据库或工作存储器。专家系统开始运行时，首先将获得的初始事实放在全局数据库中，推理机根据这些事实以及知识库进行推理，并不断的将中间结果放在全局数据库中，直到得不出结果而退出。

知识库和推理机是整个系统的核心部件，知识库以及推理机的构建和知识表示的关系很大，系统采用产生式知识表示，在此基础上构建了系统的知识库和推理机。推理机中最重要的部分就是系统的推理机制，系统根据实时监测到的数据生成推理的事实，因此，推理机采用正向推理算法。先根据网络检测到的数据生成触发推理机的事实，然后调用推理机根据知识库的规则进行推理，最后给出诊断结果。

解释接口模块是在推理过程中或者推理之后，向用户解释系统是如何得出结论的，一般的专家系统都具有该模块，因为向用户解释推理的过程，可以增加对系统推理的可信度。

该系统采用 Visual C++6.0 编程工具和 Microsoft Access 数据库进行开发。

4.2 矿井通风系统解算和故障诊断系统的实现

为了对网络解算和专家系统模块进行验证，系统实现了某矿井通风网路（如图 4.2 所示）的解算的开发和知识库及推理机的构建。该矿井有 9 条巷道。将通风网络图化成网路图时，复杂网络路应作简化处理^[6]，简化原则：

节点合并——两节点靠近，节点间风阻很小，阻力很小，可将两节点合

中的矿井通风网络结构抽象成图论中的有向图数据结构，图的节点代表各巷道的交叉点，有向图中弧的属性值中包含巷道的风阻、扇风机的风压和斜率等，在计算机中采用邻接矩阵数据结构表示图的结构。网络解算采用斯考德——恒斯雷蒙逐次近似算法对生成的网络进行解算，计算出各个巷道中的风量值和风流方向，这可以作为网络风流方向和大小诊断的依据。

知识库获取模块主要完成规则库的生成，同时规则录入时需要对知识的一致性和完整性进行检测，确保规则库中没有逻辑表达式等价性的规则、没有循环的规则和无冗余的规则等。

全局数据库相当于黑板结构，用来记录系统推理过程中用到的控制信息、中间假设和中间结果以及结论的数据库或工作存储器。专家系统开始运行时，首先将获得的初始事实放在全局数据库中，推理机根据这些事实以及知识库进行推理，并不断的将中间结果放在全局数据库中，直到得不出结果而退出。

知识库和推理机是整个系统的核心部件，知识库以及推理机的构建和知识表示的关系很大，系统采用产生式知识表示，在此基础上构建了系统的知识库和推理机。推理机中最重要的部分就是系统的推理机制，系统根据实时监测到的数据生成推理的事实，因此，推理机采用正向推理算法。先根据网络检测到的数据生成触发推理机的事实，然后调用推理机根据知识库的规则进行推理，最后给出诊断结果。

解释接口模块是在推理过程中或者推理之后，向用户解释系统是如何得出结论的，一般的专家系统都具有该模块，因为向用户解释推理的过程，可以增加对系统推理的可信度。

该系统采用 Visual C++6.0 编程工具和 Microsoft Access 数据库进行开发。

4.2 矿井通风系统解算和故障诊断系统的实现

为了对网络解算和专家系统模块进行验证，系统实现了某矿井通风网路（如图 4.2 所示）的解算的开发和知识库及推理机的构建。该矿井有 9 条巷道。将通风网络图化成网路图时，复杂网络路应作简化处理^[6]，简化原则：

节点合并——两节点靠近，节点间风阻很小，阻力很小，可将两节点合

并为一个点。

巷道合并——对于在某一区段中存在的并联巷道，可合并为一条巷道。

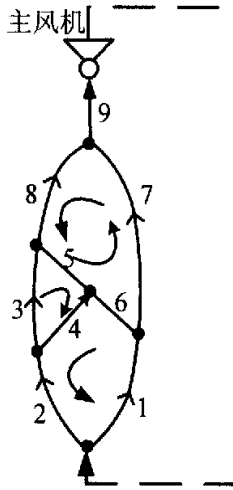


图 4.2 某矿井通风网路

图 4.2 中有 9 条巷道，两条掘进巷道（网络解算时不考虑掘进巷道），各巷道风阻为 $R_1=1.47$, $R_2=1.372$, $R_3=4.116$, $R_4=1.176$, $R_5=0.784$, $R_6=0.98$, $R_7=8.82$, $R_8=0.588$, $R_9=2.93$ ($N \cdot S^2/M^4$)，扇风机风压 1568Pa，风机特性斜率为 -196Pa，流过风机巷道的风量是 $20m^3/s$ 。对矿井通风网络解算出各巷道的分配的风量；矿井总阻力和总分阻。

4.2.1 通风网络解算设计与实现

风网解算是一个多元非线性方程组的线性化渐进解算过程，其线性化方法很多，如斯考德-恒斯雷法、牛顿法、节点压力法等。本文所采用的是目前应用较多的是斯考德-恒斯雷法。它以图论为基础，以风流运动的基本定律为依据，逐次求解网孔的修正风量，直至达到预先给定的精度并获得接近方程组真实解的风量值。

网络解算需要的数据库表如表 4.1，4.2 和 4.3 示。

并为一个点。

巷道合并——对于在某一区段中存在的并联巷道，可合并为一条巷道。

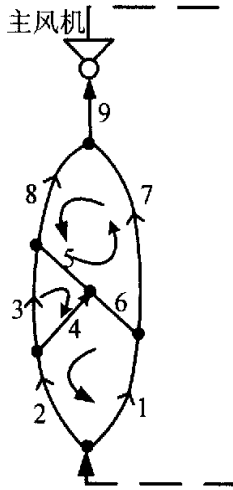


图 4.2 某矿井通风网路

图 4.2 中有 9 条巷道，两条掘进巷道（网络解算时不考虑掘进巷道），各巷道风阻为 $R_1=1.47$, $R_2=1.372$, $R_3=4.116$, $R_4=1.176$, $R_5=0.784$, $R_6=0.98$, $R_7=8.82$, $R_8=0.588$, $R_9=2.93$ ($N \cdot S^2/M^4$)，扇风机风压 1568Pa，风机特性斜率为 -196Pa，流过风机巷道的风量是 $20m^3/s$ 。对矿井通风网络解算出各巷道的分配的风量；矿井总阻力和总分阻。

4.2.1 通风网络解算设计与实现

风网解算是一个多元非线性方程组的线性化渐进解算过程，其线性化方法很多，如斯考德-恒斯雷法、牛顿法、节点压力法等。本文所采用的是目前应用较多的是斯考德-恒斯雷法。它以图论为基础，以风流运动的基本定律为依据，逐次求解网孔的修正风量，直至达到预先给定的精度并获得接近方程组真实解的风量值。

网络解算需要的数据库表如表 4.1，4.2 和 4.3 示。

表 4.1 网路顶点表

字段	类型	备注
vexId	int	顶点标识
vexName	文本	顶点名称

表 4.2 网路顶点和弧的数目表

字段	类型	备注
vexNum	int	网络节点数目
arcNum	int	网络巷道数目

表 4.3 网络弧的信息表

字段	类型	备注
laneWayName	文本	巷道名称
arcTail	文本	巷道的起点
arcHead	文本	巷道的终点
fengZu	Float	巷道的风阻
fengLiang	Float	巷道的分量
leiXing	byte	巷道的类型
fengYa	Float	风机风压
xieLv	Float	风机特性斜率

风网解算计算步骤如下。

(1) 读取有关参数

模块从数据库中读取风网解算所需的参数，并检验参数的完整性和有效性。

(2) 对各分支进行排序

风网中的分支可分为三种类型：固定风量分支；风机所在分支（简称风机分支）；一般分支（除此以外的分支）。排序后的分支顺序为：一般分支、固定风量分支、风机分支。其中风机分支和固定风量分支内部可任意排序，一般

分支按风阻由小到大进行排列。本例巷道排列：8、5、6、4、2、1、3、7、9。

(3) 确定余树弦

利用通风网路图的余树作为选择回路的基础分支。固定风量分支和风机分支作为指定的余树，再从一般分支中选择风阻大的分支作为余树，使余树的总数为 $N-J+1$ 个，以保证每个回路有余树且仅有一个。以固定风量分支为余树的回路不参加风网解算的迭代过程。

(4) 选回路

采用“加边法”选择独立回路，既把余树弦的分支除外，按分支排序次序把边（分支）一条一条地加在余树弦上，直到回路数等于 $N-J+1$ 。

(5) 赋风量初值

在迭代计算之前应对风网图的余树给定风量初值，然后根据风量平衡方程对每一个回路中的各分支赋风量初值，作为迭代计算的基础，其中固定风量的分支以固定风量为初值。最初拟定值越接近正确值，就越能减少迭代次数，即加快收敛速度。不过，即使最初拟定值与正确值相差较大，经过若干次迭代，也能得到接近于真实值的结果。本例中初拟订各巷道风量 (m^3/s) 为： $Q_1=8.7\text{m}^3/\text{s}$ ， $Q_2=11.3\text{m}^3/\text{s}$ ， $Q_3=11.3\text{m}^3/\text{s}$ ， $Q_4=0$ ， $Q_5=0$ ， $Q_6=0$ ， $Q_7=8.7\text{m}^3/\text{s}$ ， $Q_8=11.3\text{m}^3/\text{s}$ 。

(6) 迭代计算

迭代计算是已知风网各分支的风阻、自然风压和风机分压，求解风网风量分配的过程。迭代计算以回路为单位，直到所有回路的修正风量都达到预定精度为止。本实例中的风量修正精度 $\Delta Q \leq 0.001\text{m}^3/\text{s}$ 。

(7) 计算固定风量分支的阻力和风阻值

网络的风量计算完成后，为了保证固定风量分支的风量值，必须以固定风量分支的风量为依据调节其风阻。根据需要计算出的风阻值可供阻力调节时参考。其程序流程如图 4.3 所示。

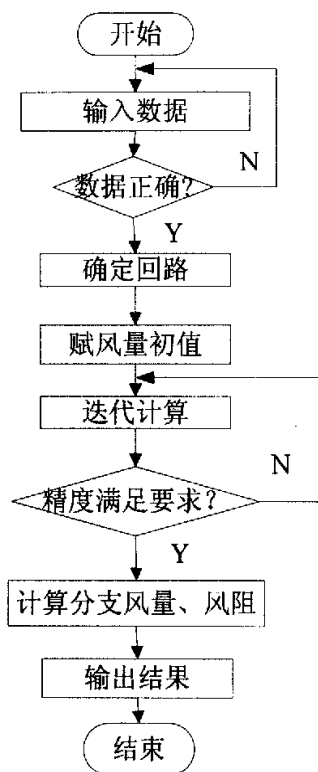


图 4.3 风网解算流程图

网络解算采用的数据结构和算法

采用邻接矩阵存储通风网络结构，数据结构如下所示：

class adjMatrix//邻接矩阵弧的信息

{public:

 CString hdmc; //巷道名称

 float fengZu;

//风阻值，风阻正负代表着风流方向，值为正表示分支风流方//向和回路相同，否则值为-1

 int intArcType; //0 风机分支,1 固定风量分支,2 一般分支

 float fengLiang;

```

float fengYa;
};

class MGraph//网络结构
{ public:
    CString vexs[MAX_VERTEX_NUM]; //存放节点的名称
    adjMatrix arcs[MAX_VERTEX_NUM][MAX_VERTEX_NUM];
    //邻接矩阵
    int vexNum, arcNum; //网络节点和弧的数目
    bool* visited; //顶点的访问标志
};

MGraph Net; //有向网络用来保存风量和风阻的，还有弧的方向
MGraph G; //无向图，用来判断回路的

```

网络解算中用到了斯考德——恒斯雷法和图的深度优先搜索法。部分代码如下：

```

//寻找回路个数
for( row_int=0; row_int<m_vexNum; row_int++)
for(column_int=0; column_int<m_vexNum; column_int++){
    type=getType(row_int, column_int);
    if(0==type)
    { //如果分支为风机分支
        tailHead[intRoad][0]=getVex(row_int);
        tailHead[intRoad][1]=getVex(column_int);
        setNull(row_int, column_int); //设置图的对应边为空
        intRoad++;
    }
    else
        if (1==type) { //如果分支为固定分支
            tailHead[intRoad][0]=getVex(row_int);

```

```
        tailHead[intRoad][1]=getVex(column_int);
        setNull(row_int, column_int);
        //设置图的对应边为空
        intRoad++;
    }
}
int tTemp=m_arcNum-m_vexNum+1-intRoad;
    //arcNum-vexNum+1 回路数
for(int k=0; k<intTemp; k++)
    //找一般分支中分阻大的直到找够回路数;
//剩下的分支构成生成树
for( row_int=0; row_int<m_vexNum; row_int++)
    for(column_int=0; column_int<m_vexNum; column_int++)
        {
            if (getGraghValue(row_int, column_int)==1)
                {
                    if(getFuZu(row_int, column_int)&&
                        getFuZu(row_int, column_int)>maxTemp)
                        {
                            rowTemp=row_int;
                            columnTemp=column_int;
                            maxTemp=getFuZu(row_int, column_int);
                        }
                }
        }

    tailHead[intRoad][0]=getVex(rowTemp);
    tailHead[intRoad][1]=getVex(columnTemp);
    setNull(rowTemp, columnTemp);
//设置无向图的对应边为空
    intRoad++;
```

```
}  
//////////加边生成回路  
CString headNodestr,tailNodestr;  
for(int i=0; i<intRoad; i++)  
{  
    headNodestr=tailHead[i][1];  
    tailNodestr=tailHead[i][0];  
    findRoad(headNodestr,tailNodestr,i);  
}
```

网络解算运行结果如图 4.4 所示。

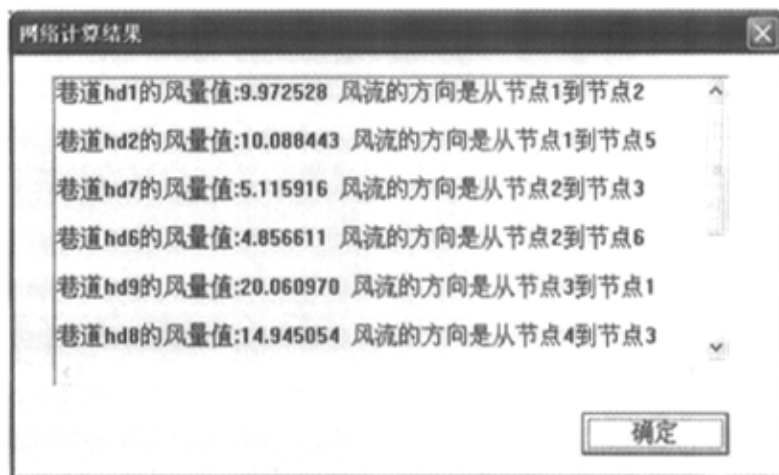


图 4.4 网络解算结果

4.2.2 故障诊断知识库设计与实现

知识库以及推理机的构建和知识表示的关系很大，系统采用产生式知识表示，这些规则通过相连的推理链联系起来形成规则网络，用这样的网络来表示知识的实体。

```
}  
//////////加边生成回路  
CString headNodestr,tailNodestr;  
for(int i=0; i<intRoad; i++)  
{  
    headNodestr=tailHead[i][1];  
    tailNodestr=tailHead[i][0];  
    findRoad(headNodestr,tailNodestr,i);  
}
```

网络解算运行结果如图 4.4 所示。

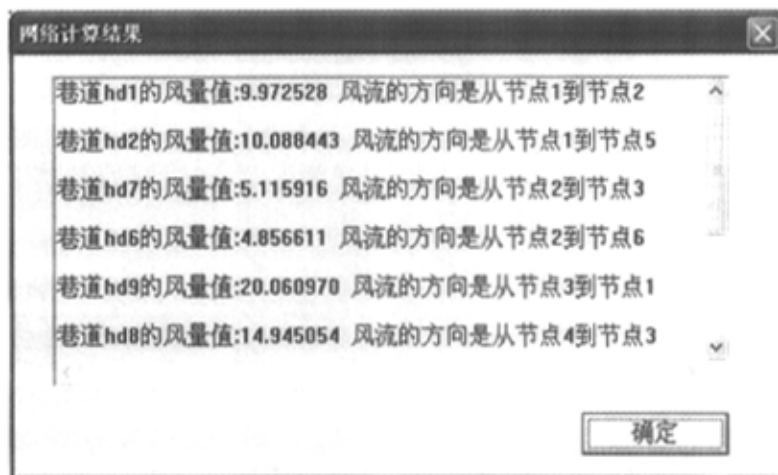


图 4.4 网络解算结果

4.2.2 故障诊断知识库设计与实现

知识库以及推理机的构建和知识表示的关系很大，系统采用产生式知识表示，这些规则通过相连的推理链联系起来形成规则网络，用这样的网络来表示知识的实体。

1. 知识库的设计

知识库主要用来存放领域专家提供的专门知识。系统的知识库是在专家指导和参考矿井通风书籍下进行知识概念化、形式化，最终完成知识获取工作，采用产生式的知识表示方法。例如对下面通风网络诊断的内容进行了概念化和形式化。

(1) 扇风机诊断

扇风机的全压是扇风机对空气做功，消耗于每 1 立方米空气的能量，其值为风机出口全压于风机入口全压之差。在没有自然风压存在时，全压等于静压加上风机出口动能损失。

克服管网阻力的风压称为扇风机静压。

风机的实际流量：一般是指单位时间内通过风机入口空气的体积。

扇风机的功率：扇风机的有效功率即扇风机的输出功率，又称空气功率或全压功率

如果扇风机静压过大，巷道中风阻过大，巷道有塌方；如果扇风机全压过小，巷道中的风阻过小，掘进巷道透了。

(2) 风筒诊断

有效风量率（供风率） P_e ：是指风筒送达掘进面的风量 Q_h 与局扇风量 Q_a 的比值；风量比 P ：是局扇风量 Q_a 与送达工作面的风量 Q_h 之比。是 P_e 的倒数。

风量比 P 是大于 1 的系数，它越大，表明风筒漏风越严重。当风筒选定后， P 为定值。风筒诊断流程如图 4.5 示。

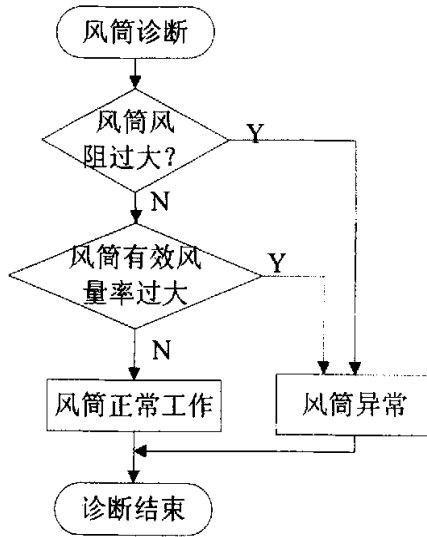


图 4.5 风筒诊断流程

(3) 瓦斯诊断

瓦斯诊断流程如图 4.6 示。

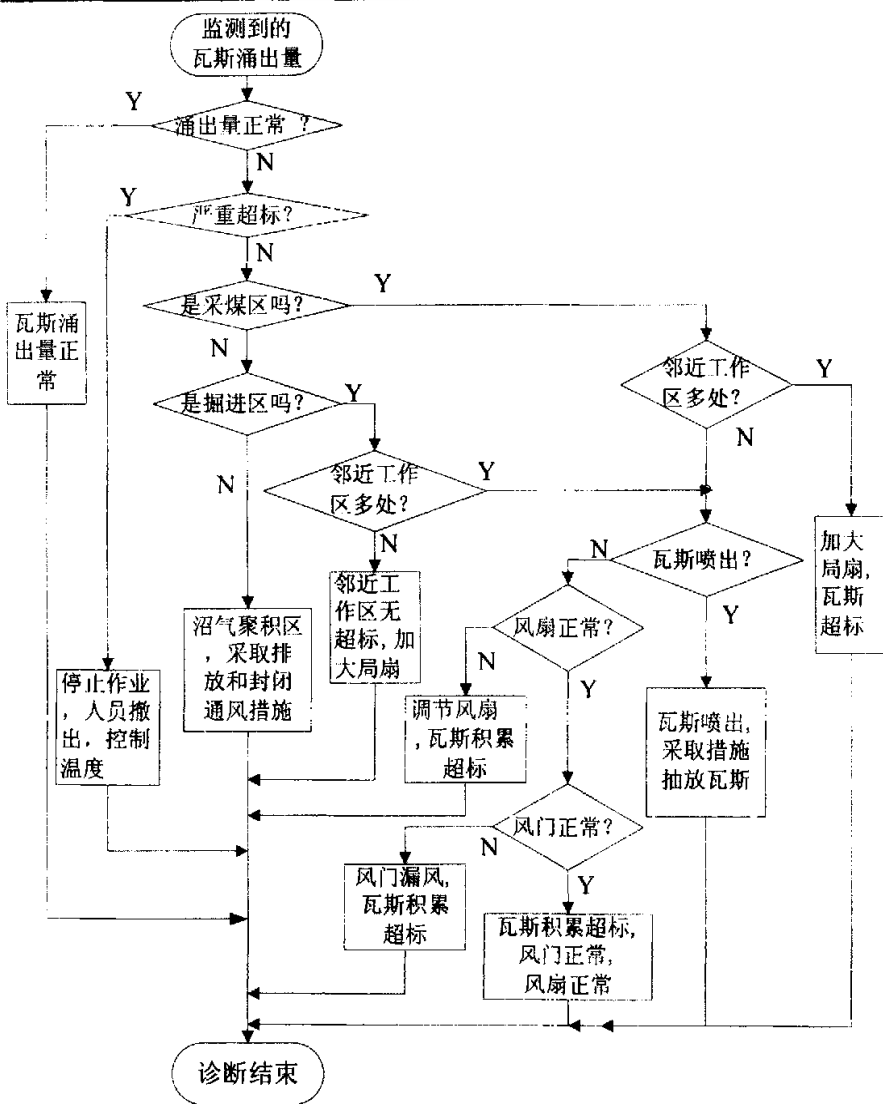


图 4.6 瓦斯诊断流程

将图 4.5 和图 4.6 的诊断内容用产生式的知识表示如下。

规则 1: if 扇风机静压大于 1560pa then 静压过大

规则 2: if 扇风机全压小于 2480pa then 全压过小

规则 3: if 回采工作面风速低于 1m/s then 回采工作面的风速过小

规则 4: if 回采工作面风量过小 and 扇风机静压过大 then 回采工作面冒顶

规则 5: if 风筒风量比大于 then 风筒漏风严重

规则 6: if 风筒风阻大于 then 风筒阻力过大

规则 7: if 风筒风量比小于 2 风筒漏风正常

规则 8: if 风筒风阻小于 2 风筒风阻正常

规则 9: if 风筒漏风严重 then 风筒不正常

规则 10: if 风筒阻力过大 then 风筒不正常

规则 11: if 回采工作面风量过小 and 扇风机正常 and 风门正常 then 回采工作面风量过小, 扇风机风量需加大, 调整扇风机叶片角度, 控制风机风量; 或者更换功率大的扇风机

规则 12: if 回采工作面风量过小 and 扇风机正常 and 风门漏风 then 风门漏风, 导致回采工作面风量过小

规则 13: if 回采工作面风量过小 and 扇风机静压过大 then 回采工作面冒顶

规则 14: if 掘进区的风量过小 and 风筒正常 and 局部风机正常 then 掘进巷道风量过小, 局扇工作正常, 调整主扇和局扇的参数

规则 15: if 掘进区的风量过小 and 局部风机不正常 then 局部风机不正常, 导致掘进巷道风量过小

规则 16: if 掘进区的风量过小 and 风筒漏风严重 then 风筒漏风严重, 导致掘进巷道风量过小

规则 17: if 掘进区瓦斯超标 and 掘进区邻近工作区瓦斯无超标 and 局扇正常 then 加大局扇功率

规则 18: if 掘进区瓦斯超标 and 掘进区邻近工作区瓦斯无超标 and 局扇不正常 then 局扇不正常, 检查局扇, 掘进区瓦斯超标

规则 19: if 掘进区瓦斯积累超标 and 掘进区邻近工作区瓦斯多处超标 and 局扇不正常 then 局扇不正常, 检查局扇, 掘进区多处瓦斯超标

规则 20: if 掘进区瓦斯积累超标 and 掘进区邻近工作区瓦斯多处超标 and 风筒不正常 then 风筒不正常, 检查风筒, 掘进区多处瓦斯超标

规则 21: if 掘进区瓦斯积累超标 and 掘进区邻近工作区瓦斯多处超标 and 主扇不正常 then 主扇不正常, 检查主扇, 掘进区多处瓦斯超标

规则 22: if 掘进区瓦斯积累超标 and 掘进区邻近工作区瓦斯多处超标 and 风门正常 and 局扇正常 and 主扇正常 then 调整主扇和局扇参数, 掘进区多处瓦斯超标

规则 23: if 掘进区瓦斯喷出超标 then 掘进区瓦斯喷出, 采取措施抽放瓦斯

规则 24: if 回采区瓦斯超标 and 主扇正常 then 回采区瓦斯超标, 加大扇风机功率

规则 25: if 回采区瓦斯喷出 then 回采区瓦斯喷出, 采取措施抽放瓦斯等等。

针对以上知识库的表示方法, 将规则存储在关系数据库中, 在数据库中定义了规则表, 如表 4.4 示。

表 4.4 规则库表

字段	类型	注释
RuleName	文本	规则名
Pre	文本	规则前件
Con	文本	规则后件
Active	Bool	规则触发状态
Type	数字	规则诊断类型

字段 Type 的值为 0 时表示规则用于诊断物理设备; 当字段 Type 的值为 1 时表示规则用于诊断非物理设备。

2. 知识的获取与检测

(1) 知识的获取

根据矿井通风专家系统的知识特点, 系统采用基于规则的产生式知识表示方法, 将从通风专家那里收集整理知识以产生式的形式存放于规则库中。每条规则的形式都是“IF 前提, THEN 结论”, 前提可以用 AND, OR 连接

子句组成，结论部分由一个或多个要执行的行动子句或结果组成。而知识库在物理结构上是采用关系数据库来存放规则知识。

依据知识库的规则结构建立规则的获取界面如图 4.7，以交互式的方式获取规则。

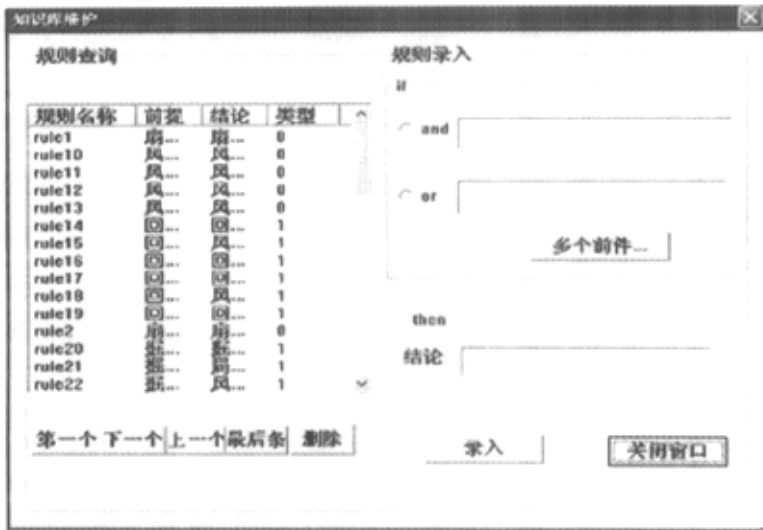


图 4.7 知识库维护窗口

鼠标点击 and 按钮时，可输入合取式前提条件，鼠标点击 or 按钮时，可输入析取式。如果输入多个前提条件，可点击多个前提按钮。

在结论文本框中输入规则的结论。

每输入完一条规则，按下“录入”按钮，系统即对输入的规则进行识别、分解、转换，判断知识的一致性和完整性，最后输入到件规则库中。

本系统提供了一个交互式的知识获取工具，以提高数据录入的效率和正确性。

上述整个规则获取的程序设计算法流程图如图 4.8 所示：

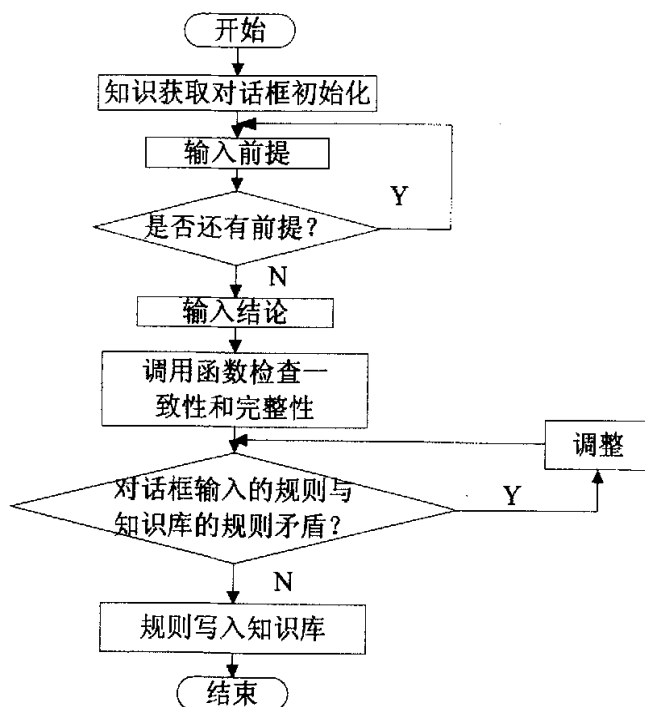


图 4.8 规则获取的程序流程图

(2) 知识一致性和完整性的检测

知识库的一致性和完整性是知识库维护的一个基本方面，所谓知识库内部各知识之间的冗余、矛盾和从属等关系，是知识库管理的一项基本指标，也是知识库实用化的一个部分。一般而言，这是一个较为困难的问题。在前面已经分析了产生式规则的一致性和完整性，并给出了相关定义及标准。

① 逻辑表达式等价性的检测

规则在录入到知识库之前，需要进行规则等价性的检测。主要思想是：将产生式规则的条件化为合取范式（每个合取项都是规则元素，即原子公式），其中合取项的个数记为 $|A|$ (A 为规则前件条件的逻辑表达式)，图 4.9 给出了等价性检测算法流程图。

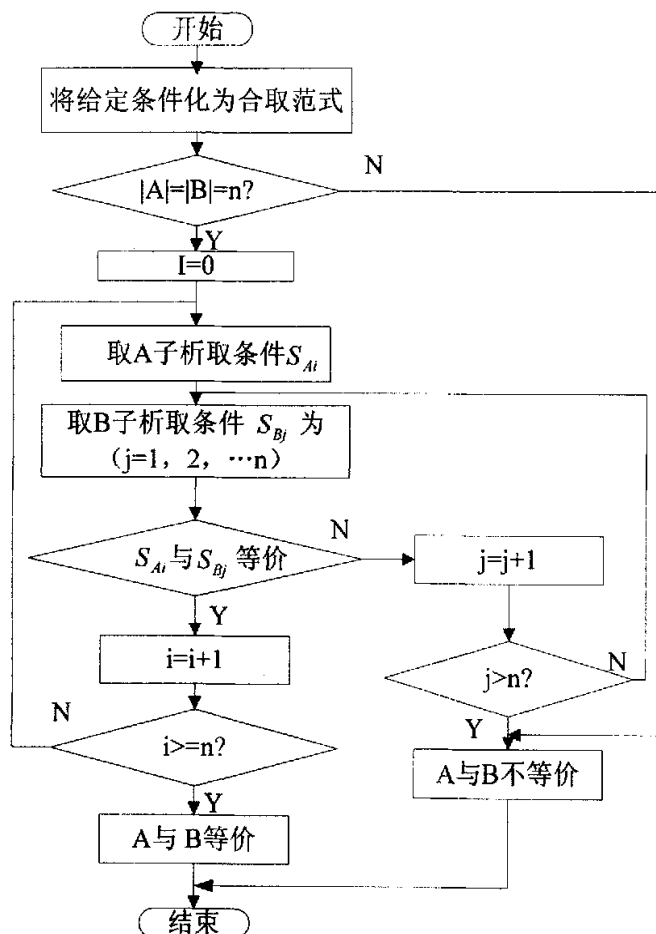


图 4.9 判断前提等价的算法流程

其中 A, B 为两个待比较的规则前件。 S_i 为前件 A 的第 i 个子析取条件, S_j 为前件 B 的第 j 个子析取条件。

规则库中的规则依次和要输入规则的前提比较, 如果相等则设标志; 如果不相等则继续比较; 如果所有的标志都为 1, 则等价。

具体算法如下:

```
int recordCount=0;
```

```
RuleNode* node=ruleChain->getFirstRule();
```

```
while (node!=NULL){
    ruleName=node->name;
    rulePre=node->premise;
    ruleCon=node->conclusion;
    recordCount++;
    CString pre_str[10],rulePreTemp; //存放规则前提,
    int findAndFlag,i=-1;
    bool loopEnd=false;
    rulePreTemp=rulePre;
    findAndFlag=rulePreTemp.Find("and");
    while (!loopEnd){
        if (findAndFlag!=-1){
            // while(rulePre->GetLength(>0){
            i++;
            pre_str[i]=rulePreTemp.Left(findAndFlag);
            pre_str[i].TrimLeft();
            pre_str[i].TrimRight();
            rulePreTemp.Delete(0,findAndFlag+3);
        }
        else {
            i++;
            pre_str[i]=rulePreTemp;
            pre_str[i].TrimLeft();
            pre_str[i].TrimRight();
            loopEnd=true;
        }
        findAndFlag=rulePreTemp.Find("and");
        //分离多个前提,findAndFlag 记录 and 的位置
    }
    int equalNum=-1;
```

```
if (preCount==i){///规则等价判断
    for(int j=preCount; j>=0; j--){
        for(int k=i; k>=0; k--){
            if(0==detectFlag[i])
                if (strPre[j].CompareNoCase(pre_str[k])==0){
                    detectFlag[k]=1;
                    equalNum++;
                    break;
                }
        }
    }
}
if (equalNum==preCount)
    {///输出等价
    PreEqualError preEqualError;
    preEqualError.m_edit1.Format("%d",recordCount);
    preEqualError.m_edit2.Format("%d",recordCount);
    preEqualError.DoModal();
    equalFlag=true;
    break;
}
}
```

如果录入规则的前提和实际规则库前提的次序相同，经过比较，能比较出相等的结论，如果次序不同，也能比较出相等的结论；如果两个规则前提条件数目相同，但规则内容不同，能得出规则不等价的结果。

它的界面如图 4.10 示。

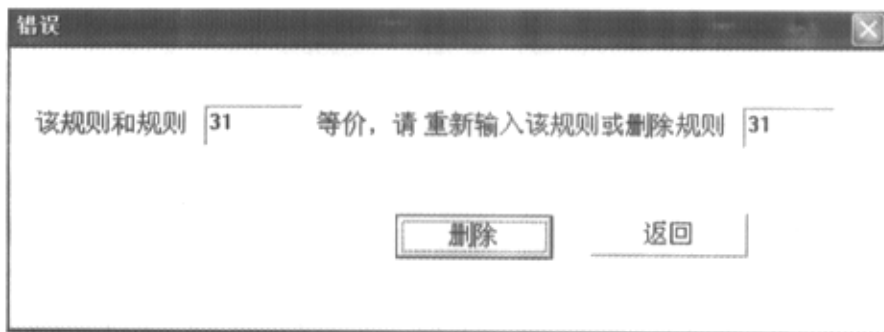


图 4.10 发生等价时出现的界面

此界面是在知识获取界面中,当鼠标按“插入”按钮后,后台进行的等价性判断中,出现和其它规则等价时的界面,它提醒用户有等价规则,应对它或另一规则进行修改。如果和所有的规则都不等价将不出现上述提示信息,继续进行该规则是否构成循环的检测。

②为了找到循环规则链,必须判断一条规则的结论部分与其它规则的条件部分是否等价,这与两个前提条件的比较是一致的。下面给出的算法和上面的算法思路不同,它通过运用等价性判断和推理算法来进行循环的检测,即每输入一条规则,就进行推理,判断推理的结论是否和输入规则的前提等价,如果有等价,则存在循环;否则,没有循环。

具体算法如下:

```
while (nodeRule!=NULL){
    CString pre_str[10],rulePreTemp; //存放规则前提,
    int findAndFlag,i=-1;
    bool loopEnd=false;
    rulePreTemp=nodeRule->premise;
    ruleCon1=nodeRule->conclusion;
    findAndFlag=rulePreTemp.Find("and");
    recordCount++;
    while (!loopEnd){
```

```
if (findAndFlag!=-1){
    // while(rulePre->GetLength(>0){
    i++;
    pre_str[i]=rulePreTemp.Left(findAndFlag);
    pre_str[i].TrimLeft();
    pre_str[i].TrimRight();
    rulePreTemp.Delete(0,findAndFlag+3);
}
else {
    i++;
    pre_str[i]=rulePreTemp;
    pre_str[i].TrimLeft();
    pre_str[i].TrimRight();
    loopEnd=true;
}
findAndFlag=rulePreTemp.Find("and"); //分离多个前提,
}
//推理
for(; i>=0; i--){
    CFact* node=factChain->getHead();
    while (node!=NULL){
        if(0==node->fact.Compare(pre_str[i])){
            matchResult++; //匹配成功
            break;
        }
        else{//找出下一个事实
            node=node->next;
            if(NULL==node)//没有匹配的事实
                i=-2;
        }
    }
}
```

```

if(matchResult==temp_int){//相等则表示匹配成功
    for(; temp_int>=0; temp_int--){
        CFact* node=factChain->getHead();
        while (node!=NULL){
            if(0==node->fact.CompareNoCase(pre_str[temp_int])){
                factChain->deleteFact(node->fact);
                break;
            }
            else{
                node=node->next;
                if(NULL==node)//没有匹配的事实
                    break;
            } } }
        factChain->appendFact(ruleCon1);
    } }

```

循环诊断界面如图 4.11 示。

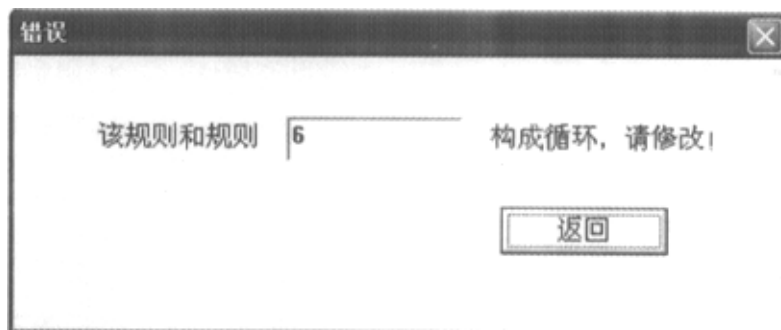


图 4.11 发生循环时的界面

此界面是在知识获取界面中，当鼠标按<插入>按钮后，后台进行的等价性判断之后，出现和其它规则构成循环的界面，它提醒用户该规则和其它规则构成循环，应对它或其它规则进行修改。如果不循环将不出现上述提示信

息, 这样这条规则就通过了一致性和完整性的检测。需要指出的是, 由于和其它规则构成循环, 就不可能都是只有一条规则来构成循环, 可能有多条, 就需要对相关的规则进行分析, 然后进行修改。

规则冗余、冲突检测部分是在等价性判断的算法的基础上进行, 它的思想是系统先检测两条规则的条件部分是否等价, 若等价, 可根据其结论部分是否矛盾来判断冲突。这样在判断规则冲突的同时, 也检测出了冗余的规则, 可以避免冗余。

4.2.3 故障诊断推理机的设计与实现

推理机是一切基于知识的智能系统的核心, 推理机的合理性直接反映到推理机结果的正确性与推理过程的实时性。在前面曾提到, 对于基于规则的知识库系统其推理方式有三种: 正向推理, 反向推理, 混合推理。本系统是一个基于产生式系统的知识库, 规则处理形式为“IF.....THEN...”的表达式。前面几章已经探讨了系统的特点, 系统推理机制采用正向推理算法。

用于存储推理过程的事实信息的数据结构和存储规则库的数据结构定义如下:

事实节点类: .

```
class Fact{
    CString name;
    CString value;
    CFact* next;
};
```

其中 name 表示事实名称, value 描述综合数据库中事实
规则节点类:

```
class RuleNode
{
    RuleNode(CString name, CString premise, CString con, bool
        flag, int type); //构造函数
    CString name, premise, conclusion;
```

息, 这样这条规则就通过了一致性和完整性的检测。需要指出的是, 由于和其它规则构成循环, 就不可能都是只有一条规则来构成循环, 可能有多条, 就需要对相关的规则进行分析, 然后进行修改。

规则冗余、冲突检测部分是在等价性判断的算法的基础上进行, 它的思想是系统先检测两条规则的条件部分是否等价, 若等价, 可根据其结论部分是否矛盾来判断冲突。这样在判断规则冲突的同时, 也检测出了冗余的规则, 可以避免冗余。

4.2.3 故障诊断推理机的设计与实现

推理机是一切基于知识的智能系统的核心, 推理机的合理性直接反映到推理机结果的正确性与推理过程的实时性。在前面曾提到, 对于基于规则的知识库系统其推理方式有三种: 正向推理, 反向推理, 混合推理。本系统是一个基于产生式系统的知识库, 规则处理形式为“IF.....THEN...”的表达式。前面几章已经探讨了系统的特点, 系统推理机制采用正向推理算法。

用于存储推理过程的事实信息的数据结构和存储规则库的数据结构定义如下:

事实节点类: .

```
class Fact{
    CString name;
    CString value;
    CFact* next;
};
```

其中 name 表示事实名称, value 描述综合数据库中事实
规则节点类:

```
class RuleNode
{
    RuleNode(CString name, CString premise, CString con, bool
        flag, int type); //构造函数
    CString name, premise, conclusion;
```

```
//规则名称, 规则前件, 规则结论
bool flag;
//flag 为 true 表示规则触发过, flag 为 false 表示规则//没触发过
int type; //代表规则诊断的类型
RuleNode* next;
};
```

事实链表类定义如下:

```
class CFactChain{
private:CFact* headFact;
public:
CFact* nextFact(CFact *node);
CFact* getHead();
CFactChain();
virtual ~CFactChain();
bool isEmpty();
void deleteFact(CString fact);
void appendFact(CString fact_str);
void DestroyChain();
};
```

规则链表类定义如下:

```
class RuleChain
{//链表没有头结点
private:RuleNode* head;
public:
RuleNode* getFirstRule();
RuleNode* nextRule(RuleNode* node);
CString findCon(CString* facts);
RuleChain();
virtual ~RuleChain();
```

```
bool isEmpty();  
void deleteRule(CString* name);  
void appendRule(CString& name, CString& pre, CString& con, bool  
flag);  
void DestroyChain();  
};
```

推理过程代码如下:

```
while (!loopEnd){  
    findAndFlag=rulePreTemp.Find("and");  
    //分离多个前提, findAndFlag 记录 and 的位置  
    if (findAndFlag!=-1){  
        i++;  
        pre_str[i]=rulePreTemp.Left(findAndFlag);  
        pre_str[i].TrimLeft();  
        pre_str[i].TrimRight();  
        rulePreTemp.Delete(0, findAndFlag+3);  
    }  
    else {  
        if(-1==findOrFlag){  
            i++;  
            pre_str[i]=rulePreTemp;  
            pre_str[i].TrimLeft();  
            pre_str[i].TrimRight();  
            loopEnd=true;  
        }  
        else{  
            rulePreTemp.TrimLeft();  
            rulePreTemp.Delete(0, rulePreTemp.Find("(")+1);  
            findOrFlag=rulePreTemp.Find("or");  
            while(!loopEnd) { //提取 or 的关键字
```

```
if(findOrFlag!=-1){
    j++;
    pre_str1[j]=rulePreTemp.Left(findOrFlag);
    pre_str1[j].TrimLeft();
    pre_str1[j].TrimRight();
    //rulePreTemp.TrimLeft();
    rulePreTemp.Delete(0, findOrFlag+2);
    findOrFlag=rulePreTemp.Find("or");
}
else{
    j++;
    pre_str1[j]=rulePreTemp;
    pre_str1[j].TrimLeft();
    pre_str1[j].TrimRight();
    int temp_int=pre_str1[j].Find("");
    pre_str1[j].Delete(temp_int);
    loopEnd=true;
}
}
} }
///在事实列表中匹配该规则前提
temp_int=i;
for(; i>=0; i--){
    CFact* node=factChain->getHead();
    while (node!=NULL){
        if(0==node->fact.Compare(pre_str[i])){
            matchResult++; //匹配成功
            break;
        }
    }
}
```



```
else{
    //找出下一个事实
    node=node->next;
    if(NULL==node)//没有匹配的事实
        i=-1; //没有匹配规则的事实，匹配结束
}
}while(node!=NULL)
}for(; i>=0; i--)
if(matchResult==temp_int){//相等则表示 and 匹配成功
//从事实列表中删出匹配成功的事实
for(; temp_int>=0; temp_int--){
    factChain->deleteFact(pre_str[temp_int]);
    for(; temp_intJ>=0; temp_intJ--){
        factChain->deleteFact(pre_str1[temp_intJ]);
    }
}
//把结论加入事实列表中
factChain->appendFact(con);
setRuleFlag(*ruleChain->getFirstRule(),ruleName);
```

正向推理是以已知事实作为出发点进行推理，使用正向推理，必须考虑到冲突消解。根据通风领域问题的特点排序，采用的是数据排序方式，即诊断物理设备，如诊断风机、风门、风筒等的规则排在规则链的前面，诊断非物理设备如风量、瓦斯等的规则排在规则链的后面。

4.3 本章小结

本章主要介绍了系统的设计与实现，文章首先对系统进行了概要设计，然后在第2章和第3章通风网络解算和知识库推理机的研究基础上编码实现了该系统，进一步验证了文中网络解算和推理算法及知识库构建策略的正确性。

```
else{
    //找出下一个事实
    node=node->next;
    if(NULL==node)//没有匹配的事实
        i=-1; //没有匹配规则的事实，匹配结束
}
}while(node!=NULL)
}for(; i>=0; i--)
if(matchResult==temp_int){//相等则表示 and 匹配成功
//从事实列表中删出匹配成功的事实
for(; temp_int>=0; temp_int--){
    factChain->deleteFact(pre_str[temp_int]);
    for(; temp_intJ>=0; temp_intJ--){
        factChain->deleteFact(pre_str1[temp_intJ]);
    }
}
//把结论加入事实列表中
factChain->appendFact(con);
setRuleFlag(*ruleChain->getFirstRule(),ruleName);
```

正向推理是以已知事实作为出发点进行推理，使用正向推理，必须考虑到冲突消解。根据通风领域问题的特点排序，采用的是数据排序方式，即诊断物理设备，如诊断风机、风门、风筒等的规则排在规则链的前面，诊断非物理设备如风量、瓦斯等的规则排在规则链的后面。

4.3 本章小结

本章主要介绍了系统的设计与实现，文章首先对系统进行了概要设计，然后在第2章和第3章通风网络解算和知识库推理机的研究基础上编码实现了该系统，进一步验证了文中网络解算和推理算法及知识库构建策略的正确性。

结 论

矿井通风是矿山生产的一个重要环节,一个安全、可靠、经济、实用的矿井通风系统,对保证井下安全生产有重要意义。本文介绍了矿井通风网络的相关技术,对网络解算算法进行了比较,重点介绍了斯考德——恒斯雷法的数学模型,给出了斯考德——恒斯雷法的网络解算步骤,设计了煤矿通风网络在计算机中的存储模型。

对专家系统原理进行了深入的研究,介绍了专家系统知识的主要表示方法,重点对产生式表示方法和知识表示的检测与求精进行了研究,根据客观情况和实际表示的因果关系,系统采用的是产生式规则的知识表示;阐述了推理机的功能,初步研究了推理机的控制和冲突消解策略,本系统使用的是正向推理方法,冲突消解控制策略发生在规则链表生成过程中,根据领域问题的特点排序,采用的是数据排序方式,即把知识在知识库组织中的顺序决定优先级的次序。在通风专家指导下,构建了系统的知识库,以某矿井井下通风网络结构为实例,设计和实现了通风网络解算和故障诊断系统,进一步验证了网络解算步骤和知识库与推理机构建策略的正确性。

本系统针对黑龙江省内的煤矿实现矿井通风系统监测智能化,能科学准确、快速地分析矿井通风状况,对通风系统进行科学的评价,找出存在的问题及其原因,提出解决问题的方案,减少煤矿通风隐患,实现煤矿安全生产,经济与社会效益十分显著,也为社会稳定、人民生命安全做出了突出贡献。

建造专家系统是一个长期性、复杂性和艰巨性的任务,决定了该专家系统要在未来几年的时间逐步的完善,具体来说系统还需要在如下一些方面继续完善:一是需要进一步增加知识库的内容,进一步增加规则,这就要求建造者和通风专家交流,将他们的知识归纳总结整理以扩充知识库,系统知识库的扩充是一个不断丰富、不断修改完善的过程。二是该系统目前各部分的功能有待于进一步完善,如获得故障监测实时数据模块目前还没有实现,需要进一步的开发。

参考文献

- [1] 中国大百科全书总编辑委员会矿冶编辑委员会等. 中国大百科全书矿冶卷. 北京, 上海: 中国大百科全书出版社, 1984: 386-388 页, 401-590 页
- [2] 谭国运等. 矿井通风网络分析及电算方法. 北京: 煤炭工业出版社, 1991: 165 -173 页, 196-200 页
- [3] 王英敏. 矿井通风与防尘. 北京: 冶金工业出版社, 1993: 160-171 页, 212-254 页
- [4] 徐瑞龙. 通风网络理论. 北京: 煤炭工业出版社, 1993: 37-38 页, 66-81 页, 161-179 页
- [5] 杨运良. 矿井通风系统网路结构复杂程度的评价. 煤矿安全. 1998: 29 (1): 321—334 页
- [6] 吴中立. 矿井通风与安全. 徐州: 中国矿业大学出版社, 1989: 130-150 页
- [7] 李恕如, 王义章. 矿井通风网络图论. 北京: 煤炭工业出版社, 1984: 48-60 页, 86-90 页
- [8] 王永庆. 人工智能原理与方法. 西安交通大学出版社, 1998: 297-306 页
- [9] 武波, 马玉祥著. 专家系统. 北京理工大学出版社, 2001: 11-12 页, 22-46 页, 71-78 页
- [10] 蔡自兴, 徐光佑. 人工智能及其应用. 北京: 清华大学出版社, 1996: 131-135 页
- [11] Cai Zixing, Tang Shaoxian. A Multirobotic Planning Based on Expert System. High Technology Letters, 1995, 1(1): 76-81P
- [12] 尹朝庆, 尹皓. 人工智能与专家系统. 北京: 中国水利水电出版社, 2002: 128-189 页
- [13] 陈文伟编著. 智能决策技术. 电子工业出版社, 1998: 60-63 页
- [14] 史忠植编著. 知识工程. 清华大学出版社, 1988: 35-46 页

- [15] 陈世福, 陈兆乾等编著. 人工智能与知识工程. 南京大学出版社, 1997: 68-74 页
- [16] 张全寿. 专家系统建造原理及方法. 北京:中国铁道出版社, 1992: 77-81 页
- [17] Wang, Y J. Mutmansky, J' an M. Modeling mine ventilation networks using five basicnetwork element. Mining Engineering, 1997, 49(3): 65-69P
- [18] 赵复申, 黄力波, 杨运良, 张敬军. 新庄煤矿通风系统分析与优化改造. 焦作工学院学报(自然科学版). 2001, 20(5) 22-24 页
- [19] 沈海鸿, 谢晨阳, 刘培根. 预测控制理论在矿井通风系统的应用, 辽宁工程技术大学学报(自然科学版), 2001, 20 (3) 33-35 页
- [20] 肖位枢. 图论及其算法. 北京: 航空工业出版社, 1993
- [21] 万江平. 用 JAVA 语言开发专家系统. 计算机应用研究. 2000 (5) :62-64 页
- [22] 李化. 玉米专家系统知识库的构建. 太原理工大学学报. 2003, 3(2) : 116-118 页
- [23] 程慧霞. 用 C++建造专家系统. 电子工业出版社, 1996
- [24] Fu K S. Learning Control Systems and Intelligent Control Systems:A Intersection of Artificial Intelligence and Automatic Control. IEEETrans, 1991, 16(1):70-72P
- [25] (美) Joseph Giarratano, Gary Riley 著. 印鉴, 刘星, 成汤庸译. 专家系统原理与编程. 机械工业出版社, 2000
- [26] W. Krentzer, B. McArthur. Developing Expert System. John wiley &sons, New York, 1990
- [27] 王树林, 袁志宏. 专家系统设计原理. 北京: 科学出版社, 1991: 19-23 页
- [28] F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. Building Expert Systems. Addison-Wesley, 1983
- [29] D. A. Waterman. A Guide to Expert Systems. Addison-Wesley 1985
- [30] David J. Kruglinski. 潘爱民, 王国印译. Visual C++ 技术内幕. 清华

大学出版社, 1999

- [31] 张惠忱. 计算机在矿井通风中的应用. 徐州:中国矿业大学出版社, 1992, 62-234 页
- [32] 夏云庆. Visual C++ 6.0 数据库高级编程. 北京: 北京希望电子出版社, 2002.
- [33] 李凤霞, 薛静峰, 黄都培. Visual C++6.0 实用教程. 北京: 电子工业出版社, 2001
- [34] Wang Y J. Solving Mine Ventilation Networks with Fixed and Non-fixed Branches [J]:Mining Engineering, 1990, 42(12): 342-345P
- [35] 王钰, 袁小红等. 关于知识表示的讨论. 计算机学报. 1995 (3) :212-225 页

攻读硕士学位期间发表的论文和取得的科研成果

- [1] 吴元宝, 孙长嵩, 王南. A Colliery Ventilation Expert System Design Based on CLIPS. 在 1st Workshop on Multidisciplinary Researches for Human Life and Human Support 会议上发表
- [2] 孙长嵩, 黄剑锋, 姬雪敏, 吴元宝. 一个基于资源复用的 CIMS 集成建模方案. 在《计算机集成制造系统——CIMS》2004 年专刊上发表
- [3] 参加了“煤矿通风系统在线监测异常数据的分析与处理专家系统”的开发

致 谢

首先我要向导师孙长嵩教授表达最诚挚的谢意，感谢孙老师两年多来给予我的无私指导和帮助。入学以来，无论在学习上还是在生活上，孙老师都给予了我悉心的指导和无尽的关爱。尤其在本文的选题、收集资料、撰写、修改和定稿的过程中，均得到了他的悉心指导。孙老师知识渊博、工作作风严谨、严于律己、宽以待人，无论走到哪里，他的这些崇高品质对我是永远的鞭策。恩师的教诲和关爱都会深深地埋藏在我的内心深处。

本论文的研究工作同时也是在副导师袁杰副教授的悉心指导和无微不至的关怀下完成的，袁杰老师严谨求实的治学态度，脚踏实地的工作作风，孜孜以求的进取精神，耐心严格的教导教诲，使我能克服各种困难，顺利完成论文。

在此也衷心感谢刘大昕教授、张健沛教授和杨静教授，在两年多的学习、科研工作中给予了我诸多指导，使我受益匪浅。

最后，感谢闵远利、宋铮、金飞虎、甘国华、张崇惠等同学为本文提出的宝贵意见以及为我无私地提供参考资料。