

## 第一部分

# 专家系统及其应用

# 仪表型实时专家系统 RTES-HU 的实现<sup>①</sup>

何华灿<sup>②</sup> 章晋义<sup>③</sup>

【内容提要】本文介绍了一个基于 RTES 原理构造的已投入实际使用的仪表型实时专家系统。它用于某炼油厂加氢裂化装置 的在线开环指导，帮助操作工人完成优化操作和系统故障诊断。

## (一) 概述

RTES-HU 是一个已通过部级鉴定的仪表型实时专家系统，它是《加氢裂化装置 0520 微机工况监测与操作指导专家系统》的核心部分，主要完成优化操作指导和故障诊断工作，此外还完成历史数据回顾和工况显示等功能。它是根据 RTES 基本模型构造的。

由于加氢裂化反应是一个高温高压下的物理化学过程，反应机理十分复杂，非线性和时滞性都很严重，难以用数学模型和机理模型对其进行精确描述。原装置中虽有常规仪表进行显示、控制和报警，但都是孤立的或小范围相关的，没有综合判断的功能。整个装置的操作、故障诊断和排除，主要依靠工人的经验，经常发生操作不平稳，对故障处理不够及时等现象，影响了生产。

为了利用计算机技术对装置进行改造、挖潜，实现安全、平稳、长周期运行，根据工厂几十年的生产经验，我们设计并实现了本专家系统 RTES-HU

RTES-HU 系统完成下述功能：

- 实时指导操作工在确保质量（生成油比重等）前提下尽量提高收率；
- 实时诊断常见的系统故障，并给出相应的处理措施；
- 自动进行历史数据回顾，找出最接近最优的生产状况的工艺条件供操作工人参考；
- 在系统“空馊”时显示某些重要工况，以使工人监视生产动态；
- 系统内部的控制和管理；

国家自然科学基金资助项目

②何华灿 男 51岁西北工业大学计算机科学与工程系副教授 研究方向人工智能智能控制  
通讯地址 (710072) 陕西西安市西北工业大学 1106 教研室

③章晋义 西北工业大学计算机科学与工程系讲师

## (二) 系统结构

整个系统由三台计算机组成。前端机完成数据采集和预处理的任务；二台上位机是长城 0520-CH，0 号上位机完成工况监测任务；1 号上位机是 RTES-HU 的宿主机，完成故障诊断和操作指导等任务。

系统的结构如图 1 所示。

各部分的主要功能是：

- 数据采集与预处理子

系统，实时采集装置的温度、压力、流量、比重等 162 路信号，进行各种滤波和预处理，然后每 6 秒向二台上位机发送一批数据。

• 若信息不能由数据采集系统直接获取，可由操作工通过观察装置的运行状态获得，然后用人工输入方式送入计算机。

• 监视模块接受前端机送来的数据和人工输入的信息，进行综合分析，如果发现异常情况，则激活专家模块。平时则显示各种工况图，或进行历史数据回顾。

• 专家模块中存有大量的经验知识和数据，一旦出现异常情况，相应的 Petri 网将被激活，系统进入推理状态。如果各种事实足以使推理达到某一结论状态，系统则输出操作指导，供操作工人参考。

系统的软件结构如图 2 所示。

各部分功能如下：

• 知识库 存放知识，知识包括三类：一是加氢装置的操作规程；二是专家的经验知识（即启发性知识）；三是关于系统本身的知识（即如何调度系统，如何区分中断级别并加以处理等）。这些知识以规则形式表示，并组成 Petri 网。

• 数据库 存放从装置中采集来的数据和人工输入的数据以及推理的中间结果，包括即时值数据，2 分钟的数据区，5 分钟的平均数据，半小时的平均数据值和二个月的重点历史数据。推理的中间结果有：当前已具有标志的状态集，上一时间周期中满足的前提集，处在不应期中的中间结论等。

- 推理机 完成对 Petri 网的推理，根据开始规则检查本时间周期所要搜索的 Petri

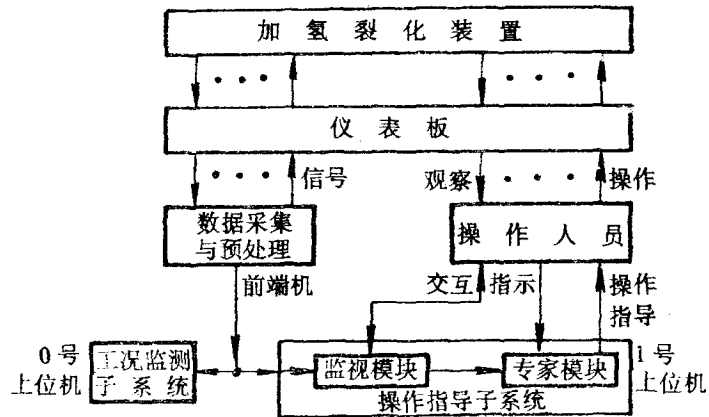


图 1. 系统结构

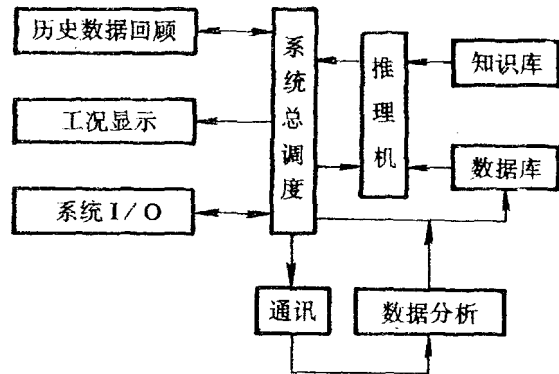


图 2. 系统的软件结构

网，传播标志和做出结论。

- 系统总调度 协调各个模块的执行顺序，包括推理、工况显示、历史数据回顾、输入/输出、通讯等模块的工作顺序。

- 通讯 它是一个中断处理程序，接受数据采集系统发来的数据，存入即时值数据区；计算有关参数的平均值，存放数据库中的相应数据区。

- 工况显示 接受系统调度程序的控制，显示相应的工况图。

- 历史数据回顾 接受键盘中断，在系统调度程序指挥下，进入或退出历史数据回顾程序，将回顾的结果存入到数据库中的相应数据区。

- 系统 I/O 根据系统调度程序的指示将推理机输出的结果输出到打印机或屏幕，接受键盘中断，输入有关数据。

需要说明的是，本系统没有解释和知识获取功能，这不是因为它们不重要，而是因为这是一个实时系统并且系统的环境是微机，实现这两项功能的时空开销都较大，在系统运行时，不允许中断系统的运行来进行解释和知识获取。

### (三) 知识表示、推理和控制策略

在加氢装置中，系统所面临的世界是一个炼油装置，它实时监测装置的生产过程，及时发现异常情况，给出相应的处理措施，实时指导操作工正确操作。在知识获取阶段，我们首先熟悉加氢工艺原理，从该领域专家处总结知识。

开始，我们是以规则形式表示专家的知识。后来发现，采用规则形式不能很好地表达专家的知识。因为这些知识包含了时间性和并行性。

最后我们通过规则中引入不应期的概念来表示时间性，将规则组成 Petri 网来表示并行性。

图 3 是 RTES-HU 中有关生成油比重的一张 Petri 网的局部结构。

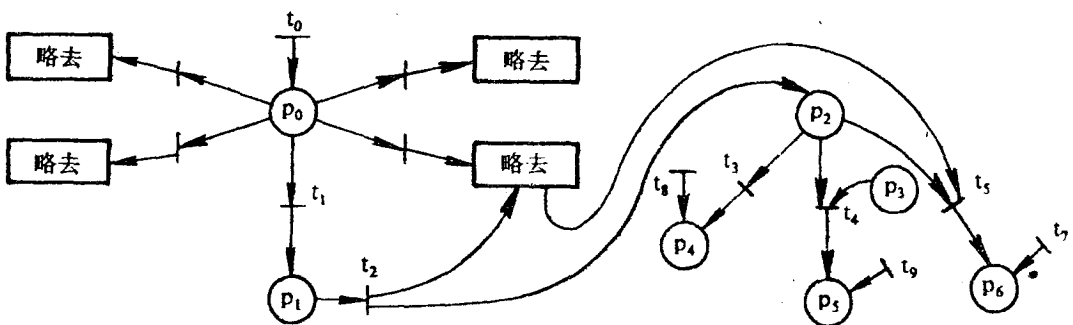


图 3. RTES-HU 中有关生成油比重的 Petri 网的局部结构

在 RTES-HU 的 Petri 网中，场所用来表示加氢装置的各种状态，转换点用来表示规则。状态分为开始状态、中间状态和结束状态，规则分为开始规则、状态规则、转移规则

和结论规则。在上图中， $t_0$  是开始规则， $t_7, t_8, t_9$  是状态规则，其余是转移规则，结论规则没有出现。 $p_0$  是开始状态， $p_4, p_5, p_6$  是结束状态，其余是中间状态。 $p_0$  表示“生成油比重大于 0.7780”， $p_4, p_5, p_6$  分别表示“瓦斯热值低于给定值”；“瓦斯压力低于给定值”和“炉出口串级失灵”。

开始规则决定系统的控制在哪一张 Petri 网上，系统调度规则选用各类规则。状态规则是根据传感器中的数据或人工报告给状态添加标志，转移规则是将标志传播到相应的后继状态中。结论规则是根据结束状态的标志决定输出结论，然后根据结论采取相应的操作。

规则的一般格式是：**IF**  $\langle$ condition $\rangle$   $\langle$ no-reac-time $\rangle$  **THEN**  $\langle$ action $\rangle$

需要解释的是  $\langle$ no-reac-time $\rangle$ ，它是“不应期”的意思，规则的前提除了满足  $\langle$ condition $\rangle$  部分外，还要满足在“不应期”要求，才能执行  $\langle$ action $\rangle$  部分。

RTES-HU 采用的是正向推理，从前端机上每 6 秒接收 162 路数据，经处理后存入相应的数据区，系统根据数据库中的数据，在 Petri 网上完成推理工作。与 RTES 中基于 Petri 网正向推理算法 forward-chaining 相似，除了转换点被打开的条件外，还要增加判断是否在“不应期”之外的条件。若都满足了，该转换点才能被点燃，然后传播标志到相应的场所（状态）中去。

RTES-HU 采用的控制策略是：

(1) 由于系统是一个实时系统，要求对随时间而变化的数据很快做出响应，因此采用的是数据驱动（正向）推理的方法。

(2) 为了解决冲突问题，对规则进行分组固定排序，如涉及故障的规则优先级高于优化的规则。故障或优化中的诸规则也根据工艺要求进行排序。

(3) 优先选用严重性大的规则，而不是使用频率高的规则。

(4) 由于加氢装置是一个时滞性很强的装置，一种状态出现后通常不会即刻消失，在实时专家系统中，若不采取特殊措施，同一条规则往往会被连续触发。为此，我对有些规则设置了“不应期”，一条规则被触发后，在不应期内，它不再响应新的激励。

(5) 本系统是一个长期运行的系统，推理机是一个死循环程序，数据库中的数据不断被新的数据所覆盖。数据库中大多数数据区都是循环链表结构，推理机总是根据新的事实进行推理。

#### (四) 中断处理

•RTES-HU 系统是一个仪表型实时专家系统。由于在 PC 机上目前没有找到合适的实时操作系统，因此程序充分利用 Compiler BASICA 2.0 的各种陷阱技术完成中断处理。

本系统的中断分为四级：1) 外部中断；2) 时间中断；3) 键盘中断；4) 顺序程序（主程序）

优先级规定为：外部中断 > 时间中断 > 键盘中断 > 主程序

1. 外部中断 外部中断处理程序也就是数据接收程序，用汇编语言编制的。系统运

行前，先将中断处理驻留在内存，中断向量表相应地改变，即将驻留程序（外部中断处理程序）的入口地址填入到该向量中。前端机每6秒中断上位机，外部中断处理程序就接受前端机送来的数据存放到内存约定的数据区内，

2. 时间中断 系统每6秒中断一次执行时间中断处理程序，接收新数据，计算、分析后将结果存入到数据库中相应的数据区。

3. 键盘中断 键盘中断是 Compile BASICA 2.0 提供的功能，用户要中止系统的正常运行，进行其它工作时，可按 F1, F2, ……，F10 功能键。本系统中 F2 … F10 中断优先级相同，但 F1 的优先级高于其它键盘中断。

4. 主程序 主程序的优先级最低，它可被任何中断程序所中断，它主要完成监测工作。

### (五) 辅助功能

RTES-HU 系统除了具有推理功能外，还提供了重点温度显示窗口，在每一个窗口内，显示了最近一分钟内各个重点温度的值。另外，还显示最近八小时的生成油比重直方图。最近24小时 < 320 收率直方图等。

每隔四小时，系统还将重点参数保存到文件中，车间要求给出一组优化参数后，系统要在保存这些数据的文件中寻找最接近优化参数的三组数据，这个功能我们称作历史数据回顾。

系统采用的是模式识别中的聚类分析方法

- 由于工艺上的要求，判别函数由车间给出，只要找到与给定的特征最相近的三组数据即可

- 机器每四小时保存一批数据，我们称为模式样本，将模式样本的各个测量值组成向量形式，称其为特征向量，它相当于多维空间中的一个点，整个模式样本集的特征向量可看成分布在特征空间中的一些点。点间距离可作为相似性的度量，按点间距离，作为分类的依据。设 X 和 Z 为二个模式样本，我们用欧氏距离来定义二个样本间的相似性。

$$D = \|X - Z\|$$

显然，模式 X 与 Z 间的距离越小，则愈相似。这与习惯用的距离概念一致，但使用时需要注意特征分量的量纲。此外，每个元素的重要程度也不一样。

车间给出要优化的特征向量是 (REVQLL, REVSZB, REVYLY, REVQLB, REVJCB, REVZLB, REVZLL, REVJCL)其中

REVQLL	轻腊量	REVSZB	生成油比重
REVYLY	原料油流量	REVQLB	轻腊油比重
REVJCL	集柴油流量	REVJCB	集腊油比重
REVZLL	重腊油比重	REVZLB	重腊油比重

设某一向量是 (DQLL, DSZB, DG104, DQLB, DJCB, DZLB, DZLL, DJCL)，计算这二个向量距离方法如下：

$$D_1 = |DQLL-REVQLL| \cdot W_1 / L_1$$

$$D_2 = |DSZB-REVSZB| \cdot W_2 / L_2$$

$$D_3 = |DG104-REVYLY| \cdot W_3 / L_3$$

$$D_4 = |DQLB-REVQLB| \cdot W_4 / L_4$$

$$D_5 = |DJCB-REVJCB| \cdot W_5 / L_5$$

$$D_6 = |DZLB-REVZLB| \cdot W_6 / L_6$$

$$D_7 = |DZLL-REVZLL| \cdot W_7 / L_7$$

$$D_8 = |DJCL-REVJCL| \cdot W_8 / L_8$$

欧氏距离  $D$  为

$$D = \sqrt{D_1^2 + D_2^2 + \dots + D_8^2}$$

其中  $(W_1, W_2, \dots, W_8)$  是各个分量的重要程度, 目前定义为

$$(1, 1, 0.8, 0.5, 0.5, 0.5, 0.5, 0.5)$$

$(L_1, L_2, \dots, L_8)$  是用统一量纲的, 即各个分量一律化成 0 到 1 之间的一个数。

利用这个方法我们从 360 个向量中, 找出最小的三组作为历史数据回顾的结果存放到数据库中, 以备需要时显示或打印。

#### (六) 小结

RTES-HU 系统已投入抚顺石油三厂实际使用, 根据鉴定委员会认为, 该系统一年所产生的经济效益为 400 万元。

(参考文献略)

# RTES 实时专家系统<sup>①</sup>

章晋义<sup>②</sup> 何华灿<sup>③</sup>

【内容提要】本文对基于规则矩阵和 Petri 网的实时专家系统及其知识表示与推理的方法进行研究。给出了 RTES 所需的理论基础，介绍了实现 RTES 的方法，描述了 RTES 建造工具的系统结构和主要功能。该系统的设计思想已在抚顺石油三厂的优化操作指导与故障诊断专家系统中实现，并于 1987.9 投用。

【关键字】实时专家系统 时间推理 不确定性推理 Petri 网 规则矩阵

## (一) 引言

### 1. 国内外现状

尽管专家系统的应用领域越来越广，数量越来越多，但能投入实际使用的却很少。其原因往往不在系统的性能上，而是在人类工程和输入输出能力上<sup>[1]</sup>。

目前情况表明，研制直接从各种传感器上接受信息的仪表型实时处理专家系统是今后一段时间的发展方向之一。

国外研究实时专家系统是从 80 年代开始的，早期的实时专家系统如 RECTOR, VM, AIRPLANE 等都是实验室里的产品，没有投入实际使用。目前，无论是军方还是厂家都迫切希望将实时专家系统投入使用。如美国陆海空三军的人工智能研究中心<sup>[2]</sup>、美国的“星球大战”计划、欧洲的“尤里卡”计划、英国的 Alvey 计划和实时专家系统用户协会 RESCU 都在加紧开发实时专家系统<sup>[3]</sup>。

我国近五年来也开始从事实时专家系统的研究。除我校外，北京钢铁学院、清华大学、华东化工学院、沈阳自动化所等单位都在从事这方面的研究，但还未投入实际使用。

### 2. 实时专家系统的特点

实时专家系统一般用于过程控制。控制过程首先对实时性要求很高，其次对可靠性要求也比其它领域高。因此，实时专家系统的结构不宜过于复杂，知识表示也应采用简单的形式。第三，由于控制对象的特性往往具有不同程度的非线性和时变性，环境干扰严重，这就要求实时专家系统能够不断采集过程信息进行各种预处理（滤波、线性化等）。推理要

国家自然科学基金资助项目·航空航天部科学基金资助项目

②章晋义男 27岁西北工业大学计算机科学与工程系讲师 研究方向人工智能专家系统  
通讯地址(710072)陕西西安市西北工业大学 1106 教研室

③何华灿 西北工业大学计算机科学与工程系副教授

能适应随时间变化的知识和数据。第四，当异常情况发生时，系统要能方便地引入人的干预。第五，对于某些应用来说，推理过程必须能够对几个进程并行处理。

### 3. 设计原则

**RTES(Real Time Expert System)** 是实时专家系统的模型系统，它应满足：1) 时间性，**RTES** 的知识表示应能反映数据和知识随时间的变化；2) 实时性；3) 并行性；4) 通用性。

## (二) **RTES** 的理论基础

**RTES** 的基础有两部分，一是规则矩阵，另一是 **Petri** 网理论。选它们作为 **RTES** 的知识表示能部分满足上述要求。这里给出的规则矩阵的定义参考了文献<sup>[7]</sup>。**Petri**网的形式定义是以袋(**bag**)论为基础的。

### 1. 规则矩阵

目前大多数专家系统是基于规则的产生式系统，这种系统功能强但执行速度慢。对实时系统来说利用规则矩阵进行推理是一个很好的方法。

规则被表示成  $C(j) \Rightarrow C(i)$ ，其中  $C(j)$  称为前件， $C(i)$  称为后件，前件  $C(j)$  为真蕴含着后件  $C(i)$  为真。以后我们统称  $C(j)$  为条件(condition)。

定义 2.1 对  $K$  条规则，一个规则矩阵  $R$  是一个  $K \times K$  的矩阵，其中

$$R(i,j) = \begin{cases} 1 & \text{如果 } C(j) \Rightarrow C(i) \\ 0 & \text{其它} \end{cases}$$

显然规则矩阵  $R$  的对角线上的元素皆为 1，因为每个条件  $C(i)$  蕴含它自己，即  $C(i) \Rightarrow C(i)$ ，规则矩阵相当于产生式系统中的规则集。

条件  $C(i)$  ( $i=1, 2, \dots, K$ ) 用于表示事实。比如  $C(15)$  定义为“反应器最高点温度在 450 以上”等等。

定义 2.2 真值状态向量  $U = (u(1), u(2), \dots, u(K))^T$ ，其中

$$u(i) = \begin{cases} 1 & \text{如果 } C(i) \text{ 为真} \\ 0 & \text{如果 } C(i) \text{ 为假或不知道} \end{cases} \quad i = 1, 2, \dots, K.$$

真值状态向量表示了当前系统的状态，它相当于产生式系统中的全局数据库。将规则矩阵乘以真值状态向量，我们就完成了一次正向推理，所得的结果相当于对全局数据库做了一次改变。

例 2.1  $C(1)$ 、 $C(2)$  和  $C(3)$  是三个条件，定义两条规则  $C(1) \Rightarrow C(2)$ ， $C(2) \Rightarrow C(3)$  真值状态向量元素值分别为  $u(1)=1$ ， $u(2)=0$ ， $u(3)=0$ 。

根据定义 2.1 和定义 2.2, 分别构造规则矩阵  $R$  和真值状态向量  $U$ :

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

将  $R$  乘以  $U$  得  $U'$ , 相当于完成了一次推理, 即  $C(1) \Rightarrow C(2)$ ,  $C(1)$  为真, 则  $C(2)$  为真。

$$R \cdot U = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = U'$$

将  $R$  再乘以  $U'$  得  $U''$ , 相当于又完成了一次推理, 即  $C(2) \Rightarrow C(3)$ ,  $C(2)$  为真, 则  $C(3)$  为真。

$$R \cdot U' = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = U''$$

这与假言三段论推得的结果完全一致. 规则矩阵  $R$  相当于图 1 所示有向图的一个邻接矩阵<sup>[8]</sup>.

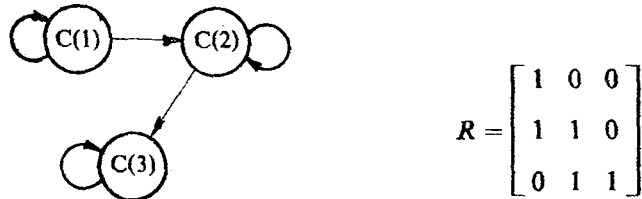


图 1. 有向图与它的邻接矩阵

规则矩阵  $R$  的闭包  $Q$  具有这样的性质, 它包含  $R$  和  $R$  中所有的隐含规则, 这个性质的证明见<sup>[7]</sup>, 我们仍以例 2.1 来说明  $Q$  的性质。

$$Q = R \cdot R \cdot R = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$Q \cdot U = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = U''$$

这样，我们只需做一次矩阵乘积，便得到了最终的结果。对实时系统来说，Q在离线时求出，求出后存入到在线计算机系统用于推理，无疑这会大大提高推理速度。

前件和后件是复合条件时的处理，将在下面给出。

## 2. Petri 网

(1) Petri 网是德国 C.A.Petri 提出，因它能很好地模拟异步并行操作，所以在并行处理和分布式计算机等领域得到了广泛的重视，作为模拟用的数学工具。

(2) Petri 网具有很多重要特性，如并行性、异步性、通用性、层次性，而这些特性对 RTES 是很可贵的。

(3) Petri 网的定义<sup>[14]</sup>

一个 Petri 网由四部分组成：场所 (place) 集 P，转换点 (transition) 集 T，输入函数 I 和输出函数 O。其中场所和转换点是 Petri 网的不加定义的原始概念。

定义 2.3 一个 Petri 网 C 是一个四元组， $C = (P, T, I, O)$ ，其中  $P = \{p_1, p_2, \dots, p_n\}$  是场所的一个有限集合， $n > 0$ ； $T = \{t_1, t_2, \dots, t_m\}$  是转换点的有限集合， $m > 0$ 。场所集和转换点集是不相交的， $P \cap T = \Phi$ 。I:  $T \rightarrow P^\infty$  是输入函数，它是转换点到场所袋的一个映射，O:  $T \rightarrow P^\infty$  是输出函数，它也是转换点到场所袋的一个映射。

有时也可将输入和输出函数定义成场所到转换点袋的映射。我们定义一个转换点  $t_i$  是场所  $P_j$  的输入，如果  $P_j$  是  $t_i$  的一个输出；一个转换点  $t_i$  是场所  $P_j$  的输出，如果  $P_j$  是  $t_i$  的一个输入。

定义 2.4  $I = P \rightarrow T^\infty$

$O = P \rightarrow T^\infty$

使得  $\#(t_j, I(P_i)) = \#(P_i, O(t_j))$

$\#(t_j, O(P_i)) = \#(P_i, I(t_j))$

例 2.2  $C = (P, T, I, O)$

$P = \{P_1, P_2, P_3, P_4, P_5\}$ ,  $T = \{t_1, t_2, t_3, t_4\}$

$I(t_1) = \{P_1\}$   $O(t_1) = \{P_2, P_3, P_5\}$

$I(t_2) = \{P_2, P_3, P_5\}$   $O(t_2) = \{P_3\}$

$I(t_3) = \{P_3\}$   $O(t_3) = \{P_4\}$

$I(t_4) = \{P_4\}$   $O(t_4) = \{P_4, P_3\}$

或者

$I(P_1) = \{t_1\}$   $O(P_1) = \{t_1\}$

$I(P_2) = \{t_1, t_4\}$   $O(P_2) = \{t_2\}$

$$I(P_3) = \{t_1, t_4\}$$

$$O(P_3) = \{t_2, t_3\}$$

$$I(P_4) = \{t_3\}$$

$$O(P_4) = \{t_4\}$$

$$I(P_5) = \{t_1, t_2\}$$

$$O(P_5) = \{t_2\}$$

(4) **Petri 网图** 上面的定义严格，适于 **Petri 网** 的研究，但不直观。我们可用圆圈表示场所，短直线表示转换点，场所和转换点之间用带箭头的弧连接起来，它们分别表示输入函数和输出函数，如此形成一个 **Petri 网图**。

例 2.3 正对例 2.2 所示的 **Petri 网**，我们画出它的的 **Petri 网图** 见图 2)。

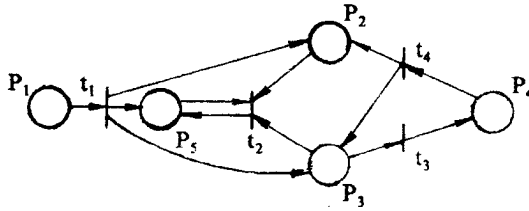
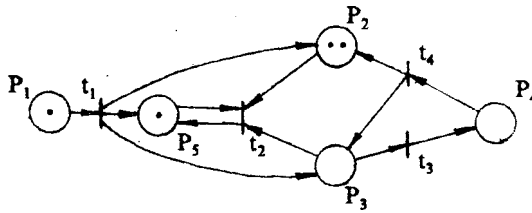


图 2. Petri 网图

(5) **Petri 网标记 (Marking)** **Petri 网** 的标记  $\mu$  反映了该网中场所内标志 (token) 的分布。标志(token)也是 **Petri 网** 的不加定义的原始概念。

定义 2.5 **Petri 网**  $C=(P, T, I, O)$  的标记  $\mu$  是场所集到正整数  $N$  的一个函数  $\mu: P \rightarrow N$  场所  $P_i$  的标志数为  $\mu_i(i=1, 2, \dots, n)$ ,  $\mu$  定义成一个  $n$ -向量  $\mu=(\mu_1, \mu_2, \dots, \mu_n)$ , 其中  $n=|P|$ 。

在 **Petri 网** 执行期间，网中标志的数目以及在场所中的分布是变化的。在 **Petri 网图** 上，标志是由场所圆圈内印上黑点来表示。例如图 3。



它的标记  $\mu=(1, 2, 0, 0, 1)$

图 3. 有标记的 **Petri 网图**

一个 **Petri 网**  $C=(P, T, I, O)$  带有标记  $\mu$  就变成一个有标记的 **Petri 网**  $M=(C, \mu)$  或  $M=(P, T, I, O, \mu)$ 。由于对全部标记集合来说，每个场所中的标志的个数是不受限制的，所以一个 **Petri 网** 有无穷个标记，当然它是可数无穷大。

(6) Petri 网的执行规则 Petri 网的执行是靠转换点的点燃来实现的，下面我们给出打开和点燃的定义。

定义 2.6 在有标记的 Petri 网  $M=(P, T, I, O, \mu)$  中 转换点  $t_j \in T$  是打开的，如果对所有的  $p_i \in P$  有

$$\mu(P_i) \geq \#(P_i, I(t_j))$$

例如在图 3 中， $t_1$  是打开的，其余转换点则不是打开的。

定义 2.7 在有标记的 Petri 网  $M=(P, T, I, O, \mu)$  中，转换点  $t_j$  被点燃，如果它是打开的，点燃一个打开的转换点  $t_j$  产生新的标记  $\mu'$  定义如下

$$\mu'(P_i) = \mu(P_i) - \#(P_i, I(t_j)) + \#(P_i, O(t_j))$$

(7) Petri 网的状态空间 Petri 网的状态是由它的标记来定义的，因此，一次转换点的点燃就意味着网的状态的一次改变。具有  $n$  个场所的 Petri 网的状态空间是全部标记的集合，即  $N^n$ 。点燃一个转换点引起的状态的改变用偏函数  $\delta$  定义，称其为“下状态函数”。

定义 2.8 下状态函数  $\delta : N^n \times T \rightarrow N^n$ ，且转换点  $t_j \in T$  是打开的，当且仅当

$$\mu(P_i) \geq \#(P_i, I(t_j))$$

如果  $\delta(\mu, t_j)$  有定义，则  $\delta(\mu, t_j) = \mu'$ ，其中

$$\mu'(P_i) = \mu(P_i) - \#(P_i, I(t_j)) + \#(P_i, O(t_j))$$

给定一个 Petri 网及初始标记  $\mu^0$ ，可以靠不断地点燃转换点来实现 Petri 网的执行。 $\mu^0$  中的  $t_j$  点燃，就产生一个新的  $\mu^1 = \delta(\mu^0, t_j)$ 。在  $\mu^1$  中又会有新的转换点如  $t_k$  点燃，则又会产生一个新的  $\mu^2$ ， $\mu^2 = \delta(\mu^1, t_k)$ 。在各标记中，至少还有一转换点是打开的，则上述过程将继续下去。如果到达某一个标记，其中没有一个转换点是打开的，这时 Petri 网的执行就停止。由此产生二个序列：

一个是标记序列  $(\mu^0, \mu^1, \mu^2, \dots)$ ，

一个是转换点序列  $(t_{j(0)}, t_{j(1)}, t_{j(2)}, \dots, t_{j(k)})$ ；

其中  $\delta(\mu^k, t_{j(k)}) = \mu^{k+1}$ ， $k=0, 1, 2, \dots$ 。一般说来，给定  $\mu$  序列，也能推导出  $t$  序列。

(8) Petri 网的可达性集合

定义 2.9 在有标记的 Petri 网  $M=(P, T, I, O, \mu)$  中，如果存在一个转换点  $t_j \in T$ ，使得  $\delta(\mu, t_j) = \mu'$ ，则标记  $\mu'$  是从  $\mu$  直接可达的。

我们定义有标记的 Petri 网  $M=(P, T, I, O, \mu)$  的可达性集合  $R(M)$  为从  $\mu$  出发能够达到的所有标记的集合。它是直接可达的自反传递闭包。

定义 2.10 可达集  $R(M)$  是一个有标记的 Petri 网  $M=(P, T, I, O, \mu)$  的最小的标记集合，定义为：

1)  $\mu \in R(M)$

2) 如果  $\mu' \in R(M)$ ，且  $\mu'' = \delta(\mu', t_j)$ ，则  $\mu'' \in R(M)$

### 3. RTES 所用的 Petri 网

基本 Petri 网不便于我们构造系统。因此，我们对它进行了一些限制和扩充。

(1) 对输入输出函数的限制 输入函数、输出函数定义为转换点集  $T$  到场所集  $P$  的映

射

$$I: T \rightarrow P$$

$$O: T \rightarrow P$$

(2) 对标记的扩充 在每个场所  $P_i$  中最多只可以有二个不同的标志“·”和“×”，因而标记  $\mu = (\mu_1, \mu_2, \dots, \mu_n)^T$

其中  $\mu_i = (n_1, n_2)$   $n_1, n_2$  的域是  $\{0, 1\}$

作这样规定主要是为了推理（包括正向推理，反向推理和混合式推理）

(3) 执行规则 执行规则分为正向执行规则和逆向执行规则

1) 正向执行规则 在有标记的 Petri 网  $M = (P, T, I, O, \mu)$  中， $t_j \in T$ ，如果  $I(t_j)$  中所有元素均具有标志“·”，则  $t_j$  是打开的。如果  $t_j$  是打开的，则点燃  $t_j$ 。  $t_j$  点燃后，新的标记变化如下：

a) 抹掉  $I(t_j)$  中所有元素的标志“·”；

b) 给  $O(t_j)$  中所有元素加上标志“·”，如果某个  $P_i \in O(t_j)$  已有标志“·”，则保持不变。

2) 逆向执行规则 在  $M = (P, T, I, O, \mu)$  中， $t_j \in T$ ，如果  $O(t_j)$  中只要有一个元素具有标志“×”，则  $t_j$  是逆打开的。如果  $t_j$  是逆打开的，则逆点燃  $t_j$ 。  $t_j$  逆点燃后，新的标记变化如下：

a) 抹掉  $O(t_j)$  中所有元素的标志“×”。

b) 给  $I(t_j)$  中所有元素加上标志“×”。如果某个  $P_i \in I(t_j)$  已有标志“×”则保持不变。

### (三) RTES 的实现及建造工具

#### 1. 知识表示

(1) 规则矩阵 这里我们给出如何建立复合规则的规则矩阵。

复合规则一般由下面四种类型的规则所组成：

类型 1:  $(a \text{ AND } b) \Rightarrow c$

类型 2:  $a \Rightarrow (c \text{ AND } d)$

类型 3:  $(a \text{ OR } b) \Rightarrow c$

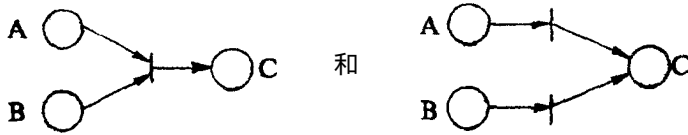
类型 4:  $a \Rightarrow (b \text{ OR } c)$

类型 1 的规则就是这里所说的复合规则，类型 2 的规则可被分成二个简单的规则  $[a \Rightarrow c]$  和  $[a \Rightarrow d]$ 。类型 3 规则等价于简单的规则  $[a \Rightarrow c]$  和  $[b \Rightarrow c]$ ，类型 4 的规则本系统不考虑。即本系统的复合规则，仅限于类型 1。

$K$  个条件  $C(1), \dots, C(K)$  存贮到一个动态的数据结构中，使得新的条件可以增加进去。假设原有的  $K$  个条件是单个原子的，也即没有合取和析取，新增加一批规则是复合规则，这时存贮条件的数据结构中，具有新的元素  $C(K+1), \dots, C(K+L) (L > 1)$ ，它表示若干个条件  $C(j) (j \in \{1, 2, \dots, K\})$  的合取。同时，真值状态向量变为  $U = (u(1), \dots, u(K), \dots, u(K+L))^T$ 。

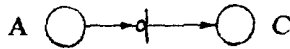
(2) Petri 网 用 Petri 网可以“模拟”其他各种知识表示方法如：

a. 逻辑运算对  $C=A \wedge B$  和  $C=A \vee B$  有



其中的标志定义为“真”如 A 场所具有标志“·”，表示 A 为真，

对于非运算，则需对转换点重新加以规定：用  $\rightarrow \text{d} \rightarrow$  表示输入场所均不具有标志，才能被点燃，对  $C=\sim A$  有



b. 语义网 我们知道语义网是由结点和弧组成，结点表示概念、事件等，弧表示结点之间的关系。这里，我们用 Petri 网的场所表示语义网的结点，转换点表示弧。凡是可用语义网表示的知识一定可用 Petri 网表示。

c. 框架 众所周知，框架也可以定义为一组语义网的结点和弧所组成<sup>[9]</sup>。根据 Petri 网的层次性可知，它同样可以表示框架。因为  $Slot_i$  是一个槽，它可以指向另一个框架，根据 Petri 网的层次性可知， $Slot_i$  又可以是一个 Petri 网的子网。

d. AND/OR 图 前面介绍了 Petri 网表示 AND、OR 运算，因此，对于 AND/OR 图，Petri 网同样可以表示。

e. 状态空间 Petri 网本身就有状态空间的定义，其标志的集合所构成的标记表示了状态，Petri 网的转换点的点燃表示操作算子的执行<sup>[10]</sup>

f. 规则 规则的形式是 IF <condition> THEN <action>。我们可用 Petri 网的输入场所表示 <condition>，输出场所表示 <action>，转换点表示蕴含关系。

g. 知识结构 知识结构是一种数据结构，可存贮特定问题领域的知识。对 Petri 网来说，它的场所、转换点、输入输出函数以及标记也构成了一种结构，它可以表示复杂的知识结构，

h. 过程性表达和说明性表达 说明性表达 如谓词 是将多数知识表示为一批静态的事实，附带一组处理它们的通用程序；过程性表达是将大量知识表示为使用知识的过程。

在 RTES 中 在大型机上 或离线的计算机上用 Petri 网表示的知识都是静态的，即知识分解为场所集和转换点集，附带一个根据 Petri 网执行规则构造的推理程序，它偏重于说明型。虽然推理效率较低，但便于知识获取和知识库的维护。在微机 上 或在线计算机上)，系统是根据在大型机上调试过的知识库经“编译”后生成的，偏重于过程型，强调执行效率。

i. 并行性和时间性 RTES 是面向实时领域的，在这个领域中，其数据和知识随时间的变化而变化。正如前面所讨论过的，Petri 网由于它固有的并行性和异步性(时间性)使得用 Petri 网表示知识后，在 Petri 网上完成推理的过程，也就完成了异步并行操作。

j. 不应期 为了解决知识随时间而变化带来的问题，我们在规则中引入了“不应期”

的概念。在不应期内，即使规则的前提被匹配，它也不能被执行；只有在“不应期”以外，它才能响应，有关“不应期”后面还要涉及。

## 2. 推理

### (1) 基于规则矩阵的推理

a. 正向推理 首先求出各个条件包括复合条件的真值，然后进行规则矩阵与真值状态向量相乘，规则矩阵  $R$  和它的闭包  $Q$  我们都要用到。采用  $R$  是便于解释和获取的知识的确证，方便知识库的修改。采用  $Q$  是为了加快推理速度，在现场的系统将用  $Q$  完成推理工作。

b. 反向推理 反向推理即从结论到前提，在规则矩阵的有向图中，反向推理即将有向图的箭头改变方向。改变方向后所得的矩阵是原来矩阵的转置。

规则矩阵  $R$  的闭包  $Q$  的转置  $Q^T$  也具有这个性质。

(2) 基于 Petri 网的推理 在 Petri 网上推理可完成异步并行操作，它无需回溯。

a. 正向推理 为了实现对 Petri 网的推理，对 Petri 网中的任一个场所  $p_i$ ，我们定义如果  $I(p_i) = \Phi$ ，则为初始场所，它的标志是由数据处理程序从传感器中采集的数据处理后决定是否添加的。推理的结束是看  $O(p_i) = \Phi$  的场所是否具有标志，如果有，则推理结束；否则继续推理。 $O(p_i) = \Phi$  的场所称终结场所。

因为 RTES 是实时专家系统，所以它的推理程序是一个死循环程序，无论最后是否做出结论，它都将执行下去。

b. 反向推理 对于反向推理，可以看成 Petri 网中改变箭头的方向，将终结场所看成是“初始场所”而将初始场所看成是“终结场所”。推理从终结场所开始，直到遇到初始场所为止，如果初始场所中中断处理程序给该场所添加了标志“·”并且反向推理程序将另一个标志“x”，也传播到该场所，这时所做的假设就得到了证实。

## 3. RTES 中的不确定性

在实时领域中同样存在着证据的不确定性和知识的不精确性，不确定性推理方法有很多<sup>[15][16]</sup>，在此不一一叙述。这里我们给出对规则矩阵和 Petri 网的模糊推理方法。

(1) 基于规则矩阵的模糊推理 由于证据的不确定性（数据的不确定性），使得真值状态向量  $U$  中的元素的值就不是 1 或者 0，我们取 0 到 1 之间的数表示条件的不确定性，1 代表完全肯定，0 表示完全否定，这样得到的真值状态向量就成了模糊向量。

$$U = (u(1), \dots, u(K))^T \quad \text{其中 } u(i) \in [0, 1]$$

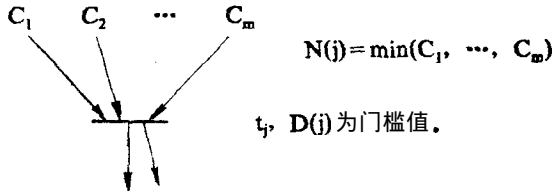
对于规则的不确定性，在规则矩阵  $R(i, j)$  中的一个模糊值表示条件  $C(j)$  蕴含条件  $C(i)$  的信任程度，比如

$$R(i, j) = 0.7 \quad \text{表示 } C(j) \stackrel{0.7}{\Rightarrow} C(i)$$

$R(i, j)$  取值也是在 0 到 1 之间，这样构成的规则矩阵就是一个模糊矩阵。完成模糊推理的工作就相当于二个模糊矩阵相乘。矩阵相乘时，原来的 AND 运算改为 MIN 运算，原来的 OR 运算改为 MAX 运算。

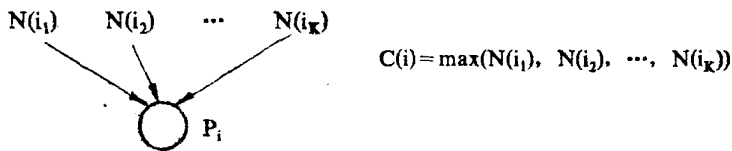
(2) 基于 Petri 网的模糊推理对 Petri 网的模糊推理要复杂一些。首先将标志重新定义，原来定义每个场所或者有标志，或者无标志。现在定义一个场所的标志是一个 0 到 1 之间的数。它或者从某个转换点传播过来或者从传感器中得到。

其次每一个转换点也有一个 0 到 1 之间的数，我们称为阈值  $D$ ，如果转换点  $t_j$  的输入部分的可信度大于等于阈值  $D$ ，则该转换点才能被点燃。如果有若干个从其它场所指向转换点  $t_j$  的箭头，则所有到达  $t_j$  的模糊值  $C_1, C_2, \dots, C_m$  必须进行 MIN 运算，得到一个新的模糊值  $N(j)$ 。它就为转换点  $t_j$  的输入部分值，



这意味着合取条件中弱证据占优势。如果  $N(j) > D(j)$ ，则转换点  $t_j$  被点燃，它将模糊值  $N(j)$  传播到它的输出场所中去。否则  $N(j) = 0$ 。

另外，对从一个以上转换点传播过来的模糊值，进入某个场所  $p_i$ ，这时，该场所的模糊值就是将传播过来的模糊值进行 MAX 运算后得到的结果，称为  $C(i)$



这意味着析取条件中强证据占优势。

这样定义的 Petri 网，我们称为模糊 Petri 网。我们得到三个向量：

$C = (c(1), \dots, c(k))$ ，表示  $k$  个场所具有的模糊值，称场所标志向量。

$N = (N(1), \dots, N(L))$ ，表示  $L$  个转换点具有的模糊值，称转换点向量。

$D = (D(1), \dots, D(L))$ ，表示  $L$  个转换点上的阈值，称阈值向量。

#### 4. RTES 建造工具

现有的专家系统工具都是用于非实时的，不满足实时的要求。本节介绍了我们所设计的一个工具系统，它是面向实时的。

实时控制系统所面临的主要问题是如何在保证优异控制性能的基础上，解决其结构的复杂性与控制的实时性之间的矛盾。此外，由于现场条件和成本因素，不能选择各种 LISP 机或其它类型专用机。为了解决这些矛盾，较好的方法是在大型机上构造通用的实时专家系统工具，通过它生成一个“编译”过的实时专家系统，装入微机后在现场使用。

在大型机上，领域专家和知识工程师一道建立知识库，用仿真数据库中的数据对知识