

# 计算机医疗诊断系统的推理算法

陈 阳 军

**提 要** 本文所介绍的是KD型计算机医疗诊断系统所采用的推理算法，对诊断系统来说具有普遍意义。作者在此算法的基础上，已成功地发展了上消化道系统，血液系统以及下消化道系统的疾病诊断系统（在CROMEMCO-III与TQ-100汉字终端的联机系统上实现）。

## 一、系 统 结 构

KD型计算机医疗诊断系统主要由引导模块、诊断模块、治疗模块、知识获取模块以及知识库所组成。引导模块用于初步诊断，而诊断模块则在初诊的基础上进行进一步的严密的推理判断。治疗模块在确诊之后根据患者年龄、身体状况以及其它特征指标给出治疗方案。而知识获取模块用于推理规则、短句的修改和补充。知识库存贮着医学专家的经验 and 数据，它表现为短句集合和推理规则的集合。它们之间的关系如图1所示。其中患者数据为系统输入，诊断结果和治疗方案为系统输出。

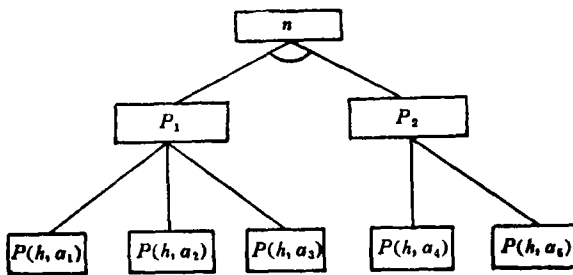


图1 系统的模块化结构

## 二、推理规则的分类

知识库提供引导模块，诊断模块和治疗模块所用的全部知识，由短句集合  $L$  和推理

规则集合  $I$  所组成。

设  $H$  表示全体人类的集合,  $N$  表示病名集合。

又设  $X$ 、 $Y$ 、 $Z$  分别表示病症集合、病史集合和体征集合。

谓词  $P$  的意思为…具有…,  $Q$  的意思为…是…, 则对于任意  $l \in L$ , 恒可表示为:

$$P(h, x), P(h, y), P(h, z)$$

$$\text{或} \quad Q(h, x), Q(h, y), Q(h, z)$$

其中  $h \in H, x \in X, y \in Y, z \in Z$ .

例如  $P(h, x)$  可解释为: 患者  $h$  具有症状  $x$ .

我们将  $I$  划分为  $I_1, I_2, I_3$ , 亦即

$$I = I_1 \vee I_2 \vee I_3$$

对于任意  $I \in I_1$ , 恒可表示为:

$$n \leftarrow A_1, A_2, \dots, A_m$$

其中  $A_i = P_{1i}(h, \alpha_{1i}) \vee P_{2i}(h, \alpha_{2i}) \vee \dots \vee P_{ji}(h, \alpha_{ji}) \vee \dots$  ( $i=1, \dots, m$ )

$\alpha_{ji} \in X \vee Y \vee Z, n \in N, \vee$  表析取关系。

上式解释为: 如果  $A_1, A_2, \dots, A_m$  同时成立, 则  $n$  成立。

例如, 规则:

如果患者有溃疡病史

或 有反复出血症状

或 有慢性反复发作的规律性上腹部疼痛

同时 上腹剑突下有压痛

或 胃液分析胃酸高

则 患者患有消化性溃疡

可表示成:

$$n \leftarrow (P(h, \alpha_1) \vee P(h, \alpha_2) \vee P(h, \alpha_3)) \wedge (P(h, \alpha_4) \vee P(h, \alpha_5))$$

其中  $\alpha_i (i=1, \dots, 5)$  分别表示上一条规则中所出现的病史、症状或体征。  $n$  表示病名‘消化性溃疡’。

这部分规则, 最主要为诊断模块所用。

对于任意  $I \in I_2$ , 恒可表示为:

$$P_i \leftarrow \sim P_j$$

表示一种不精确的推测, 其含义为, 如果  $P_j$  成立, 则  $P_i$  可能亦成立。例如:

如患者腹痛, 则系统可推测其为上腹痛, 右下腹痛等等。

这构成了初始引导的基础。

又记  $M$  为药名的集合, 则对于任意  $I \in I_3$ , 恒可表示为:

$$m_1, m_2, \dots, m_i \leftarrow P_1 \wedge P_2 \dots \wedge P_j$$

其中  $m_1, m_2, \dots, m_i \in M$ , 其含义为如果  $P_1, \dots, P_j$  同时成立, 则可选药为  $m_1$ , 或  $m_2, \dots$ , 或  $m_i$ 。例如, 如患者患有食管—贲门粘膜撕裂症, 并且具有呕吐现象, 则可选用爱茂尔、胃复安、吐来抗等等。这部分规则主要为治疗模块所用。

以下只着重讨论集合  $I_1$  中的规则。

### 三、推理算法

为了有效地实现推理算法，我们用与或树的形式来存贮推理规则。例如，以上所说的推理规则：

$$n \leftarrow (P(h, \alpha_1) \vee P(h, \alpha_2) \vee P(h, \alpha_3)) \wedge (P(h, \alpha_4) \vee P(h, \alpha_5))$$

可用以下树形结构来表示：

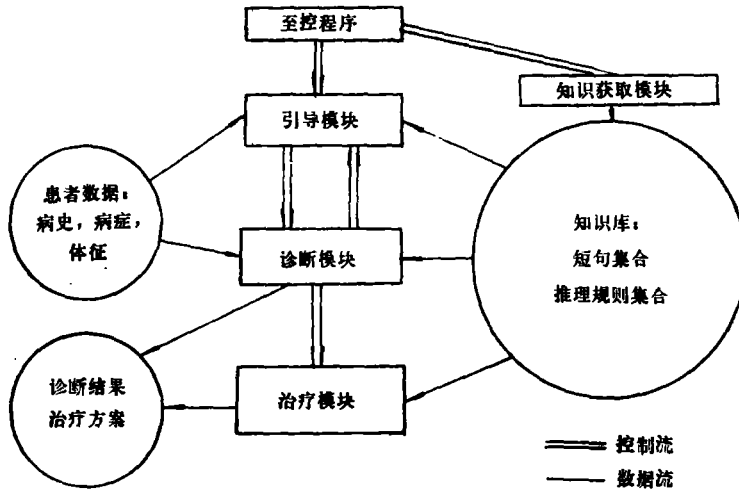


图2 推理规则与或树

图中，用弧线联连起来的节点为与节点，没有用弧线相联的节点为或节点。任何复杂的推理规则，均可通过一系列变换转化成上述标准形式。但这往往会失掉其原有的逻辑上的直观意义，所以我们尽量保持其原有形式不变，而用层次更深的与或树来表示。尽管如此，我们仍然可以通过设计一个对任何层次的与或树都适用的算法来简洁地实现任何复杂的规则的推理。

为了实现推理算法，需要设置三个堆栈 $S_0$ 、 $S_1$ 、 $S_2$ ，三者均具有以下结构（如图3）。

运算： $PUSH(S_i, T)$ 表示元素 $T$ 进栈 $S_i$ ；

$PUSH(S_i(E), q)$ 表示元素 $q$ 进栈 $S_i$ 的节点域；

$S_i(L) \leftarrow x$ 表示 $x$ 值记入当前栈 $S_i$ 顶元素的标号域；

$S_i(P) \leftarrow x$ 表示 $x$ 值记入当前栈 $S_i$ 顶元素的指针域；

$TEMP \leftarrow S_i(E)$ 表示栈 $S_i$ 顶元素送给暂存单元 $TEMP$ ，但不出栈；

$POP(S_i)$ 表示栈 $S_i$ 顶元素出栈。

$\vee$ 表示或关系， $\wedge$ 表示与关系， $(q, SON_i)$ 表示 $q$ 元素的第 $i$ 个子节点， $TEMP_1$ 、 $TEMP_2$ 为两暂存单元。

以下是用自然语言和以上运算式、符号表出的推理算法，其基本思想是：从上向下

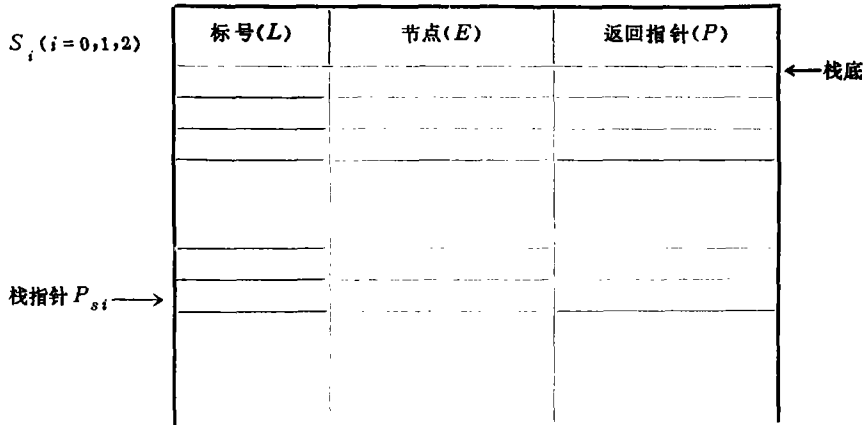


图 3

找到叶节点，然后从叶节点逐级向上搜索至根节点。或者推出结论，或者说明结论不成立。

推理算法：

- (0)  $l \leftarrow 0; PS_0 \leftarrow 0; PS_1 \leftarrow 0; PS_2 \leftarrow 0;$
- (1)  $PUSH(S_0(E), root); S_0(L) \leftarrow l; l \leftarrow l+1; \quad [与或树根节点进栈]$   
 $PS_0 \leftarrow PS_0 + 1;$
- (2) if  $S_0$  栈顶元素为叶节点 then goto(4)  
     else goto(3);
- (3)  $x \leftarrow s_0(L); TEMP_1(E) \leftarrow S_0(E); i \leftarrow 0; \quad [S_0$  栈顶元素的所有子节点进栈  $S_0]$   
     Repeat  
          $PUSH(S_0(E), (TEMP_1(E), SON_i)); i \leftarrow i+1;$   
          $S_0(P) \leftarrow x; S_0(L) \leftarrow l; l \leftarrow l+1; PS_0 \leftarrow PS_0 + 1$   
     Until 所有子节点都已进栈  $S_0$ ; goto(2);
- (4)  $TEMP_1 \leftarrow POP(S_0); PS_0 \leftarrow PS_0 - 1;$
- (5) if  $TEMP_1$  为真 then  $PUSH(S_1, TEMP_1); PS_1 \leftarrow PS_1 + 1; f \leftarrow 1;$   
     else  $PUSH(S_2, TEMP_2); PS_2 \leftarrow PS_2 + 1; f \leftarrow 0;$
- (6) if  $S_0$  空 then 结束  $[若 root$  在  $S_1$  中则说明推出结论，否则  $root$  在  $S_2$  中]  
     else  $TEMP_2(E) \leftarrow S_0(E);$
- (7) if  $f=0$  then goto(9)  
     elst goto(8);
- (8) if  $TEMP_1$  与  $TEMP_2$  为  $\wedge$  关系 then goto(2)  
     else  $TEMP_2 \leftarrow POP(S_0); PS_0 \leftarrow PS_0 - 1; if TEMP_2(L) = TEMP_1(P)$   
     then  $PUSH(S_1, TEMP_2); PS_1 \leftarrow PS_1 + 1; goto(6);$
- (9) if  $TEMP_1$  与  $TEMP_2$  为  $\vee$  关系 then goto(2)  
     else  $TEMP_2 \leftarrow POP(S_0); PS_0 \leftarrow PS_0 - 1; if TEMP_2(L) = TEMP_1(P)$   
     then  $PUSH(S_2, TEMP_2); PS_2 \leftarrow PS_2 + 1; goto(6);$

此算法最终在(6)结束。若  $root$  在  $S_1$  中, 则说明推出结论, 且可根据标号、返回指针之间的联系对结论进行说明。若  $root$  在  $S_3$  中则说明结论不成立, 并且也可以根据标号、返回指针之间的联系说明结论不成立的理由。

### 参 考 文 献

- [1] Nils J. Nilsson, Principles of Artificial Intelligence.
- [2] E. H. Shortliffe, Computer-Based Medical Consultation.
- [3] Elaine Rich, Artificial Intelligence.

## The Reasoning Method of Computer-Based Medical Diagnosis System

Cheng Yangjun

### Abstract

This paper presents the reasoning method applied to KD Computer-Based Medical Diagnosis System. Upon this reasoning method, the author successfully developed the diagnosis system for the upper-digesting system, the blood-circulating system and the lower-digesting system diseases (achieved on the combination system of CROMEKO-III to the TQ-100 reading end).