

79-82

数字网络模式匹配推理算法

关守平 柴天佑

TP18

(东北大学自动化研究中心 沈阳 110006)

摘 要 This paper explores the knowledge-base compiling technique to compile the rule-based of production system into a kind of digital network, which is the foundation of a digital network-based pattern match reasoning algorithm. The algorithm effectively solves the key problem of fast reasoning in the real-time expert system and has the ability to reason non-monotonically and to treat asynchronous events. It realizes the uniformly processing symbolic and numerical information, and guarantees the reasoning response timely and reasoning results consistently.

关键词 Real-time expert system, Production rules, Pattern match, Non-monotonic, Asynchronous events.

1. 引言

在实时专家系统领域,面临着诸如^[1]时序推理,非单调推理,时限推理,非精确推理,异步事件处理等问题。时限推理问题,即在有限的时间内给出推理结论,是实时专家系统区别于传统专家系统的主要特征之一。目前,这方面的研究工作主要有:(1)级进推理方法^[2],(2)有限深度搜索方法^[3],(3)高速推理算法,如 Rete 算法^[4],Treat 算法^[5],(4)多处理机并行推理方法^[6],(5)知识库编译的方法^[7]。以上各种方法,虽然各自的着重点不同,但都是基于符号推理的模式。作者认为,如果利用知识库编译技术,将符号信息映射成数字信息,用数值运算代替符号匹配给出推理结论,再借助于一种高速推理算法(如下文所述),无疑将大大加速推理进程。同时,也实现了符号/数值信息的统一处理,使专家系统机制与常规控制算法可在同一编程环境下用实时语言(如 Ada, PERAL)或过程性语言(如 C, PASCAL)统一编程,在 Ven-Norman 串行机上实现基于传感信息推理的实时专家系统。最近 Gudwin 等人已经做了一些有益的工作^[8]。

2. 数字向量式知识表达方法

在实时专家系统中,知识多采用产生式规则,即

“if...then...”的形式,如下述一组规则:

例 1 if e_1 and e_2 then e_3 and e_4
if e_1 and e_3 and e_5 then e_6
if e_2 and e_4 then e_7 and e_8
if e_8 then e_9

本文中,规则左手部(LHS)和右手部(RHS)都采用合取的形式,若规则 LHS 中有析取式(RHS 中一般不会有析取式),则做如下变换:

if m and $(n \text{ or } p)$ then $t \Rightarrow \begin{cases} \text{if } m \text{ and } n \text{ then } t \\ \text{if } m \text{ and } p \text{ then } t \end{cases}$

设规则中的每一项模式 e_i 代表一个事件,则集合 $E = \{e_i, i=1, 2, \dots, n\}$ 代表规则库中所有事件组成的集合, n 维向量 $X = [x_1, x_2, \dots, x_n]^T$ 与集合 E 相对应,记作 $X \in E$,代表 n 维向量组成的状态空间,其对应关系为:

$$X \in E, x_i = \begin{cases} 1, e_i \text{ is true} \\ 0, e_i \text{ is false,} \\ i=1, 2, \dots, n \end{cases} \quad (1)$$

将事件集 E 划分成三个子集 E_I , E_M 和 E_O , E_I 对应于输入事件集,即直接与输入变量相联系的事件; E_O 为输出事件集,即与输出变量相联系的事件; E_M 为中间事件集,代表推理过程的中间结论。 E_I , E_M 和 E_O 对应的向量分别为 X_I , X_M 和 X_O , 记作 $X_I \in E_I$, $X_M \in E_M$ 和 $X_O \in E_O$ 。如例 1 中,假设 e_1 和 e_2 为真,则有:

关守平 博士研究生,主要从事智能控制、专家系统、过程控制等方面的研究。

$X =$

$$[x_1 | e_1 x_2 | e_2 x_3 | e_3 x_4 | e_4 x_5 | e_5 x_6 | e_6 x_7 | e_7 x_8 | e_8 x_9 | e_9]^T$$

$$= [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1]^T$$

$$X_1 = [x_1 | e_1 \ x_2 | e_2 \ x_3 | e_3]^T = [1 \ 1 \ 0]^T$$

$$X_M = [x_3 | e_3 \ x_4 | e_4 \ x_5 | e_5]^T = [1 \ 1 \ 1]^T$$

$$X_O = [x_6 | e_6 \ x_7 | e_7 \ x_8 | e_8 \ x_9 | e_9]^T = [0 \ 1 \ 1]^T$$

推理的过程就是从 X_1 出发, 经过规则库匹配推理, 最终求出 X_O 的过程, 即 $X_1 \rightarrow \text{规则库} \rightarrow X_O$, 再利用 X_O 对应的事件集 E_O 产生相应的输出。整个专家系统的工作过程如图 1 所示。

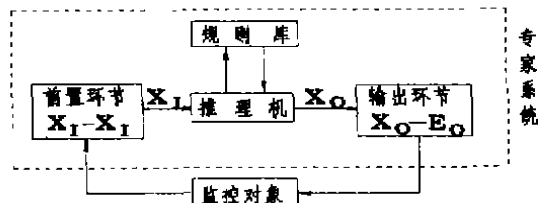


图 1 专家系统工作简图

3. 数字网络模式匹配推理算法—DNDF

从图 1 可以看出, 前置环节实现了符号信息数字化, 接下来的问题就是如何利用数字信息在规则库中进行推理, 这就需要对规则库进行编译, 转换成可以直接利用数字信息进行推理的形式。

在传统的产生式规则系统中, 每一个推理过程都要用到所有事实的信息, 无法保留以往推理过程中产生的有用信息, 因此, 不可避免地产生事实迭代利用的问题, 影响推理进程的快速性。另外, 在实时环境中, 输入模式的变化是有限的, 在监控对象稳定的情况下, 每次只有少量输入模式发生变化 (否则系统将会处在一种剧烈变化的动荡环境中), 因此, 保留以往推理过程的信息, 每次推理只将变化后的新信息注入推理系统, 无疑将大大加速推理过程的进行, 同时还可有效解决实时环境中的非单调推理、异步事件处理等问题。网络法是利用数字向量进行推理, 并实现上述思想的有效途径。

网络法首先由规则库编译器将规则库映射成一个推理网络, 以例 1 为对象, 形成的网络如图 2 所示。

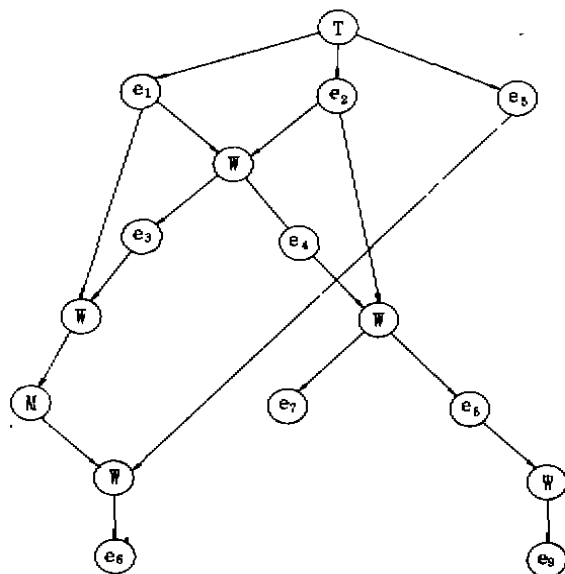


图 2 规则网络图

网络中, T 为起始结点, 即输入事实; e_i 为模式结点, 代表规则库中的每一事件; W 为 wait 结点, 保留推理过程中的信息; M 为 middle 结点, 起过渡连接的作用; 标有箭头的连接线表示推理方向。

上述网络的计算机实现, 应给每类结点定义一种数据结构, 形成一个数字推理网络, 图 3 以表格形式给出每类结点的结构形式。

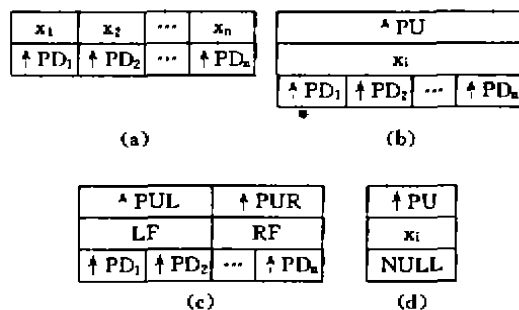


图 3 结点结构示意图

对于图 3 中的各类结点, 说明如下:

(1) 图 (a)、(b)、(c)、(d) 分别表示起始结点、具有后代的 e_i 结点、wait 结点和终止结点。middle 结点与 e_i 结点结构相同, 只是 $x_i = 0$;

(2) x_i ($i = 1, 2, \dots, n$) 为第 i 个状态变量;

(3) $\uparrow PU$ 为指向父结点的指针, $\uparrow PD_i$ 为指向第 i 个子结点的指针, $\uparrow PUL$ 和 $\uparrow PUR$ 为指向左、右父结点的指针;

(4) LF 和 RF 为左、右标志。

由上述结点形式形成的数字推理网络如图 4 所示,其形成按如下算法由规则库编译器完成:

FORMING NETWORK 算法

step1: 形成起始结点 T

step2: 将规则库中输入事件集中的元素排在第

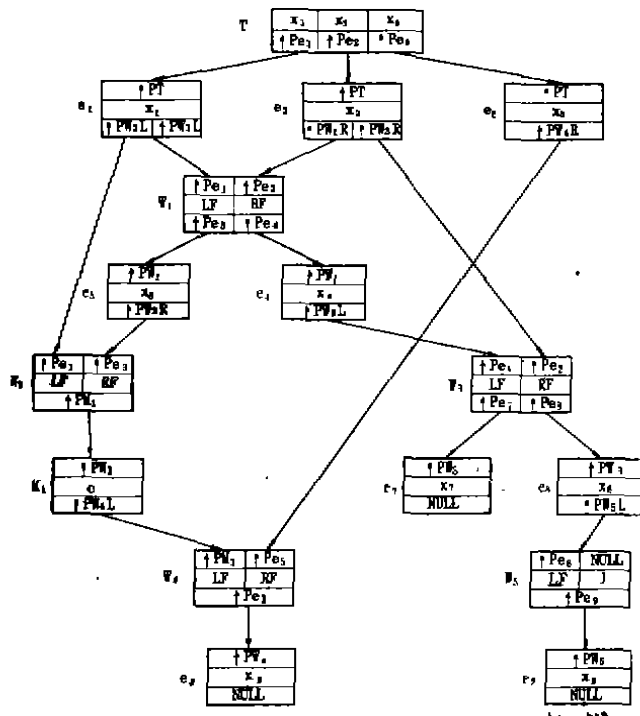


图 4 数字规则网络图

一层,建立 T 到 e_i 的联系(指针);

step3: 按规则形成上层节点的 wait 子结点,建立与上层节点的联系,若 wait 节点只有一侧输入,则置另一标志(LF 或 RF)为 1;

step4: 按规则形成上层 wait 结点的子结点 e_i ,若 $e_i \in E$,则置该结点为 middle 结点 M ,建立与上层节点的联系;

step5: 规则库中所有规则用完, goto step6,否则 goto step3;

step6: 网络形成。

由于网络具有记忆作用,因此类似于 Rete 算法,作用于网络的输入向量应是变化后的信息 ΔX_i ,这一特点是解决非单调推理和异步事件处理的关键。 ΔX 的取值是由本次输入向量 X_i 与上次输入向量 X_i^0 的状态变量的一对一相减,即

$$\Delta X_i = X_i - X_i^0 = [x_i^1 - x_i^0 \ x_i^2 - x_i^0 \ \dots \ x_i^n - x_i^0]^T \\ = [\Delta x_1 \ \Delta x_2 \ \dots \ \Delta x_n]^T$$

其中 $\Delta x_i (i=1, 2, \dots, n)$ 的取值为 $+1, 0, -1$ 。+1 代表新增的事实, 0 代表维持不变的事实, -1 代表失效的事实。

由图 4 及 ΔX 的定义,得到如下数字网络深度优先模式匹配推理算法 DNDFA (Digital Network Depth-First Algorithm)。

* 算法 DNDFA

step1: 以变化后信息向量 ΔX 作为输入向量,将 ΔX 中非零变量的后继结点放入 OPEN 表中,调整 OPEN 表,使同一结点只在 OPEN 表中出现一次(适用于非单调推理和异步事件的情况);

step2: 若 OPEN 表为空, goto step4;

step3: 取 OPEN 表中最前面的结点放入 CLOSED 表中,取出状态变量 Δx_i ;

1) 若输入向量 ΔX 的变量 Δx_i 为 1 或结点中存储的变量为 0 (对应 middle 结点),则取出结点的后继 wait 结点放入 WAIT 表中,并将所指出的 wait 结点相应侧的标志 (LF 或 RF) 置为 1,调 WAIT 表处理算法, goto step2;

2) 若输入向量 ΔX 的变量 Δx_i 为 -1, 则: a) 取出结点的后继 wait 结点,若该结点所指出的 wait 结点的相应侧标志 (LF 或 RF) 为 1, 则将标志清零,并将该 wait 结点放入 WAIT1 表中,调 WAIT1

表处理算法; 否则继续; b) goto step2;

step4: SOLVED 表中的变量形成输出向量 X_0 。算法 DNDFA 结束。

* WAIT 表处理算法

step1: 若 WAIT 表为空, 则算法结束;

step2: 若 WAIT 表中最前面的结点放入 CLOSED 表中;

1) 若 $LF=RF=1$, 取后继结点的子结点 wait 结点。若无 wait 结点, 则将后继结点中的变量 x_i 取出

放入 SOLVED 表中;否则将所指向的 wait 结点相应侧的标志(LF 或 RF)置为 1,放入 WAIT 表的前面, goto step1;

2)若 $LF=0$ 或 $RF=0$, goto step1. [END]

·WAIT1 表处理算法

step1:若 WAIT1 表为空,则算法结束;

step2:取 WAIT1 表最前面的结点放入 CLOSED 表中,若 $LF=0$ 且 $RF=1$ 或 $LF=1$ 且 $RF=0$,则取其后继结点 n_x ;

1)若 n_x 无 wait 子结点,则将 n_x 中的状态变量 x_i 从 SOLVED 表中删除, goto step3;

2)若 n_x 有 wait 子结点,则取出 wait 子结点。若 n_x 所指向的 wait 结点相应侧的标志(LF 或 RF)为 1,则将标志位清零,并将 wait 结点放入 WAIT1 表的前面;否则继续;

step3: goto step1. [END]

当一次推理过程还没有结束,输入向量就发生变化时(即推理过程中出现非单调性和异步事件现象),只需将变化后的状态变量所对应的结点加入 OPEN 表的后面,即可在原推理过程的基础上继续进行,并可保证推理结论的一致性。

4. DNDFA 算法的性能分析

数字网络与传统的产生式规则穷尽式匹配策略相比较,无论从工作效率和编程实现方面,都具有极大的优势。

(1)传统的产生式系统,是基于一种符号式匹配推理策略,这种方式难以用适于控制的过程性语言(如 C、PASCAL)或实时语言(如 Ada, PERAL)实现,若用智能语言(如 Prolog, Lisp),则推理速度低,难以满足实时控制的要求。而数字网是属于数值处理的范畴,完全可以用适于实时控制的语言编程实现,达到了推理过程与控制算法统一处理的目的。

(2)传统的产生式系统在匹配推理时,每条规则所包含的模式都需进行匹配,而不管规则之间的模式是否有重复;数字网络则不然,对于规则库中的每种模式只匹配一次,尤其是对于推理过程的中间结论不予处理,在网络中只起到一种连结过渡作用,这

样就大大地减少了模式的匹配数量。

(3)数字网络的最大优势在于它的记忆功能,只对变化的信息起作用,每个输入模式只输入网络一次,直到它再次发生变化为止。对于控制系统而言,稳定运行时每次输入信息的改变并不是全部,只是一少部分而已(否则系统就会处在一个剧烈的动荡中),因此,规则的增多对于数字网络推理反应时间的影响并不显著,这就有效地解决了传统产生式系统规则库规模受限的问题。

(4)传统的产生式系统对于实时推理过程中的非单调性和异步事件问题无能为力,而数字网络则借助其记忆功能巧妙地解决了这些问题。

(5)算法 DNDFA 虽然是针对确定性推理而言,但只需略加修改,便可方便地推广到不确定性推理(例如 CF-模型推理)。

参考文献

- [1] 关守平,柴天佑,实时专家系统的现状及展望,第二届全国智能控制专家讨论会论文集,1994
- [2] Lattimer, W. M., An expert system for real-time control, IEEE Software, March, 1986
- [3] Karf, R. E., Real-time heuristic search; first result, Proc. AAAI-87 WA(1987)
- [4] Forgy, C. L., Rete, a fast algorithm for the many pattern/many object pattern match problem, Artif. Intell., 19(1), 1982
- [5] Miranker, D. P., Treat, a better match algorithm for AI production systems, same to [3]
- [6] Moor, R. L., et al., A real-time expert system for process control, Proc. 11th conf. on Artif. Intell. Application, 1985
- [7] Lauber, R. J., Artificial intelligence techniques in real time control systems, Proc. 12th IFAC World Congress, Sydney, Australia, Vol. I, 1993
- [8] Gudwin, R. R., et al., A real-time procedure for knowledge procession, same to [7]