Petri网用于Horn子句的逻辑推论

林 闯

(国家信息中心信息科学与应用研究所,北京 100045)

APPLICATION OF PETRI NETS TO LOGICAL INFERENCE OF HORN CLAUSES

Lin Chuang

(*Institute of Information Science and Application State Information Centre, Beijing* 100045)

Abstract This paper studies Petri net models for the Horn clause form of propositional logic. Since finding the T-invariants of Petri net models of logical inference is the key step, the paper investigates the algorithms for computing such invariants. These are based on the idea of resolution, and exploit the presence of one-literal, pure-literal and splitting clauses to lead to faster computation.

摘要 这篇论文探索了命题逻辑的Horn子句的Petri网模型. 求解逻辑推论Petri网模型的T-不变量是求解逻辑推论的核心步骤. 本文提供了计算T-不变量的算法,这些算法基于归约的思想. 另外,在算法中利用单字母规则、纯字母规则和割裂规则可提高算法的速度和简化算法的复杂性.

§1. 引 言

推论问题是人工智能的基础,尤其在诸如数据库、专家系统、决策支持系统和逻辑编程应用中,逻辑推论担当了核心角色. 推论过程就是确定一个给定的命题是否由所收集的一组数据(规律)所蕴含.

Horn子句是子句形式的一个重要子集,这是因为任何可由逻辑表达的问题都可转换成Horn子句的表达式[1].

不同的模型已经用于表示知识推论系统. Petri网之所以被选来摸拟逻辑推论,是因为Petri网具有很好的模型描述特性:并发、不确定和异步. 除此而外,Petri网有很好发展的理论和数学分析方法支持它. 将逻辑推论问题转换成Petri网模型,并用现存的Petri网分析方法去处理逻辑推论问题可以增强以不同和有效的方法处理这类问题的机会.

Lautenbach[2]、Sinachopoulos[3]和Murata[4]已经做了关于逻辑推论Petri网模型的

本文1991年3月2日收到,1991年5月7日定稿. 本工作部分得到国家自然科学基金和中科院管理、决策、信息系统开放实验室的资助. 作者林闯,副研究员,主要研究领域为分布系统,系统性能分析,Petri网应用和人工智能.

一些工作，他们给出了从一组子句转换成一个 Petri 网模型的算法过程，而且建立了一组 Horn 子句不一致的必要网论条件和充分网论条件.已经得出结论：在一组 Horn 子句的 Petri 网模型中，一个目标变迁是潜在可以发生当且仅当(以后，简记为 iff)存在一个非负的 T－不变量，它的支持中包含着这个目标变迁.

我们这篇论文的主要目的和工作是为逻辑推论的 Petri 网模型的 T－不变量计算提供一个新的观察和算法.Petri 网的线性表示和不变方法是逻辑推论模型核心环节，各种逻辑推论规则可以用于逻辑的线性表示，使 Petri 网的逻辑推论过程更加有效.

§2. Petri 网和命题逻辑模型

2.1 Petri 网基本定义

定义 2.1.1：一个 Petri 网是一个五元组$(S, T; F, M_0, W)$，其中：

(a) $(S, T; F)$是一个有限网，S 是位置集合，T 是变迁集合，F 是弧集合.它们有下列性质：

—$S \cap T = \varnothing$(位置和变迁之间不相交)；

—$F \subseteq (S \times T) \cup (T \times S)$(位置和变迁间的流关系)；

—$S \cup T \neq \varnothing$(非空网)；

—$dom(F) \cup cod(F) = S \cup T$(没有孤立元素).

(b) M_0是初始标记.标记 $M: S \to N$，表示标识在位置中的分布.

(c) $W: F \to \{1\}$是弧的权函数，在逻辑模型中，权为“1”.

在图形中，我们用线表示变迁、圆圈表示位置、位置中的黑点表示标识.

$X = S \cup T$ 叫做网元素集合，一个网元素 $x \in X$ 的前集(后集)可以表示为 $^{\cdot}x = \{y \mid y \in X, <y, x> \in F\}$ ($x^{\cdot} = \{y \mid y \in X, <x, y> \in F\}$).

定义 2.1.2：设 $PN = (S, T; F, M_0, W)$是一个 Petri 网：

(a) 一个变迁 $t \in T$ 在标记 M 下可发生 iff $\forall s \in {}^{\cdot}t, M(s) \geqslant W(s, t)$.

(b) 如果 $t \in T$ 在 M 下可发生并执行，得 M 的接继标记 M′，并用 $M[t>M'$ 表示.$\forall s \in S, M'(s) = M(s) - W(s,t) + W(t, s)$ (注：若$(x, y) \notin F$，则定义 $W(x, y) = 0$).

(c) 如果存在着 $M_{i_1}[t_{i_1}>M_{i_2}$，$M_{i_2}[t_{i_2}>M_{i_3}$，…，$M_{i_n}[t_{i_n}>M_{i_{n+1}}$，我们就说变迁执行序列 $\sigma = <t_{i_1}, t_{i_2}, \cdots, t_{i_n}>$在 M_{i_1} 是可执行的.

定义 2.1.3：$PN = (S, T; F, M_0, W)$是一个具有 n 个变迁和 m 个位置的 Petri 网.它的关联矩阵 $C = [C_{i,j}]$具有 n 行和 m 列，每一行代表一个变迁，每一列代表一个位置，$C_{i,j} = W(t_i, s_j) - W(s_j, t_i)$.

关联矩阵是 Petri 网结构的线性代数表示，它也同系统状态(标记)的变化紧密相连.用 C^t 表示关联矩阵与变迁 t 相关的行，我们可以把上述的接继标记计算写作：$M' = M + C^t$.

定义 2.1.4：C 是一个具有 n 行、m 列的 Petri 网 PN 的关联矩阵.一个 n 维整数向量 Y 是 PN 的一个 T－不变量，如果 $C^T Y = 0$.　T－不变量 Y 的支持是变迁的集合，这些变迁所对应的在 Y 中的元素都是正整数，Y 的支持表示为 $|Y|$.一个支持是最小 iff 它不包含除了自身和空集以外的另一个不变量的支持.很容易证明，一个向量 $Y \geqslant 0$ 是一个 T－不变量 iff 存在一个标记 M 和一个变迁执行序列 σ，使得 $M[\sigma > M$ 并且序列 σ 的变迁

执行次数向量 $\bar{\sigma}=Y$.

2.2 Horn 子句模型

一个 Horn 子句一般有这样的形式：$B:-A_1,\cdots,A_n$. 符号：-的含义为：$A_1,\cdots,A_n$ 所有条件都成立，就蕴含结论 B 成立. 在 Horn 子句中，可以有零个或多个条件的"与"，但最多只能有一个结论. 有四种不同形式的 Horn 子句和它们相应的 Petri 网模型：

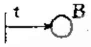
(1) 非空条件和结论的 Horn 子句

$B:-A_1,\cdots,A_n \quad n\geqslant 1$

例如：子句 $C:-A,B$ 可以由 Petri 网做如下模型：

(2) 空条件的 Horn 子句　　$B:-$

这类子句可以被解释为"事实". 一个事实"B"在 Petri 网模型中可以描述为一个"流变迁"，如下图：



(3) 结论为空的 Horn 子句

$:-A_1,\cdots,A_n \quad n\geqslant 1$

这类子句称为"目标"子句，在逻辑中以非的形式出现，它是要被证明的. 例如，子句：$-A_1,A_2$ 在 Petri 网模型中可以描述为一个"漏变迁"(又称目标变迁)，如上图.

(4) 空子句，这类子句可以解释为不一致，在 Petri 网推论算法中表示结束. 空网不在本文定义中.

从一组 Horn 子句转换成 Petri 网或相关矩阵的过程，详见[5, 6]. 下面让我们来看一个例子.

例 1：(来自[4])给定一组 Horn 子句：

1) A　2) B　3) $A\wedge B\rightarrow E$　4) $E\wedge B\rightarrow D$　5) $D\rightarrow A$　6) $D\rightarrow E$

如果我们要证明 $D\wedge E$ 是"真"，即可由这组 Horn 子句所蕴含，我们就可把 $D\wedge E$ 作为目标子句(即，$\neg D\vee\neg E$)加入到这组 Horn 子句中. 我们可以得到如图 1 所示的 Petri 网模型和关联矩阵. 一个子句按序号对应着一个变迁，目标子句对应着变迁 t_7.

$$
\begin{array}{c} \\ t_1\\ t_2\\ t_3\\ t_4\\ t_5\\ t_6\\ t_7 \end{array}
\begin{array}{c} \begin{array}{cccc} A & B & E & D \end{array} \\
\begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ -1 & -1 & 1 & 0\\ 0 & -1 & -1 & 1\\ 1 & 0 & 0 & -1\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & -1 \end{bmatrix} \end{array} = C
$$

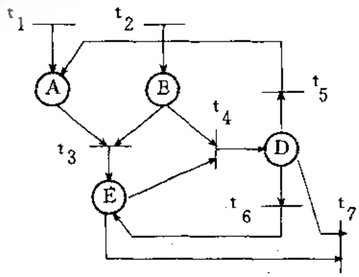


图 1　例 1 所对应的 Petri 网模型和关联矩阵

§3. 命题逻辑 Petri 网模型的逻辑推论

在通常，我们检测 J 是否蕴含 Q，就检测 J 和 $\neg Q$ 的一致性，如不一致，就证明了 J 蕴含着 Q. 另外，在[3]和[4]中，已有定理证明：在一组 Horn 子句 J 的 Petri 网模型 PN

的表示中，J 是不一致的 iff PN 有一个 T－不变量 Y 并且 $Y \geqslant 0$，$Y(tg) \neq 0$(目标变迁 tg 所对应的执行次数值不为零).这样，逻辑推论问题就变成求解相应 Petri 网模型的 T－不变量.Martinez 已经给出了一个求解一般 Petri 网模型 T－不变量算法[7]，它是将关联矩阵中的符号相反的行对进行相加运算，以达到变迁合并的目的.这种运算在逻辑模型中可以看作是逻辑归约过程.如果我们能充分地利用逻辑规则来简化逻辑模型的逻辑归约过程，就能达到简化 Petri 网模型的 T－不变量的计算.

我们介绍三个逻辑规则用于加速 T－不变量的计算.假定 J 是一组 Horn 子句.

(1)**单字母规则**:如果 J 中有一条规律仅包含一个字母 L，我们可以消除 J 中所有包含着 L 的规律，而得 J'.那么就有：J 是不一致的 iff J'是不一致的.

在例 1 中，t_1 行是单位子句仅有字母 A，而 t_5 行是包含 A 的子句.所以，在开始执行 T－不变量计算前，就可以消除 t_5 行，达到简化计算的目的.同时，t_5 行的消除并不影响包括目标变迁最小支持 T－不变量的计算.

(2) **纯字母规则**:如果字母¬L 不出现在 J 中，那么字母 L 就是纯的.如果字母 L 在 J 中是纯的，那么我们可消除所有包含着 L 的子句，而获得 J'.如果 J'是空，J 是一致的；否则，J 是不一致的 iff J'是不一致的.

关联矩阵某一列元素都具有相同符号，这样的列和其非零元素所对应的行是不参加求 T－不变量运算的.及时消除这样的行和列可以简化计算的复杂性.

当我们不能使用单字母规则和纯字母规则时，可以考虑使用割裂规则.

(3)**割裂规则**:假定 J 有这样的形式：$(A_1 \vee \neg L) \wedge \cdots \wedge (A_m \vee \neg L) \wedge (B_1 \vee L) \wedge \cdots \wedge (B_n \vee L) \wedge G$.其中，$A_i$，$B_i$ 和 G 既不包含 L 也不包含¬L.这样，我们就可把 J 割裂成两部分：$J_1 = A_1 \wedge \cdots \wedge A_m \wedge G$，$J_2 = B_1 \wedge \cdots \wedge B_n \wedge G$.有结论：J 是不一致的 iff J_1 和 J_2 都是不一致的.

把一个关联矩阵按某个位置分成两个矩阵，尔后分别计算 T－不变量.这些 T－不变量要进行线性合并，才能得到原矩阵的 T－不变量.这种合并要满足割裂位置的流关系和流量平衡关系.

现在我们来利用上述三个规则来修改 T－不变量的算法，使之更有效.将 Martinez 的算法定义为过程 1，单字母规则的应用编写在过程 2，纯字母规则的应用编写在过程 3.用上述三个过程和割裂规则编写计算 Petri 网逻辑模型的 T－不变量算法.

过程 1:C 是一个 Petri 网的 n×m 的关联矩阵，I_n 是单位矩阵.

(1) $A_1 := C$; $D_1 := I_n$.

(2) Repeat for i=1 until i=m do

(2.1)对矩阵 A 中第 i 列元素符号相反的任意两行，在矩阵[D ⋮ A]中进行相加运算，并将产生的新行加入[D ⋮ A]中.

(2.2) 从[D ⋮ A]中消除在 A 中第 i 列元素不为零的行，尔后消除第 i 列.

过程 2:

输入：一个关联矩阵 C，有 n 行和 m 列.

(1) Repeat for k=1 until k=n do

if C 的第 k 行仅包含"0"和唯一的"1"(或"－1")在第 P_i 列

then 除了第 k 行外，标定所有在 P_i 列有值为"1"(或"－1")的行.

(2)消除所有标定的行，而且修改 n 使得：

$n_1 := n$－消除行数.

输出：一个关联矩阵C′，有n行和m列.

过程3：

输入：一个关联矩阵C，有n行和m列.

(1) $i_1=m$.

(2)Repeat for $k_1=1$ until $k=i$ do

If 第k列包含着仅"1"和"0"(或者"−1"和"0"),

Then

begin 删除所有在k列有非零元素的行，并且$n_1=n-$删除行数.删除第k列并且$m_1=m-1$.

end;

输出：一个关联矩阵C′，有n行和m列.

算法1：(计算逻辑模型的T−不变量)

输入：一个关联矩阵A，有n行和m列.

(1) $C_1=A$; Call 过程2;

(2) $C_1=C'$;Call 过程3;

(3) 从C′中选择"0"元素最少的列P_i;

(4)删除P_i列，而且割裂C′为C_1和C_2使得：

C_1包含在P_i列有值为"1"或"0"的行，

C_2包含在P_i列有值为"−1"或"0"的行.

(5) $C_1=C_1$;Call 过程2;

(6) $C_1=C'$;Call 过程3;

(7)Call 过程1为C_1部分计算T−不变量，可得a个不变量，$Y_j(0\leqslant j\leqslant a)$.

(8) $C_1=C_2$;Call 过程2;

(9) $C_1=C'$,Call 过程3;

(10)Call 过程1为C_2部分计算T−不变量，可得b个不变量，$X_k(0\leqslant k\leqslant b)$.

(11) For $0\leqslant j\leqslant a$, $0\leqslant k\leqslant b$

If $\exists\ Y_j(l)\neq 0\wedge|Y_j(l)|\in P_i$ and $\exists\ X_k(h)\neq 0\wedge|X_k(h)|\in P_i$

then

begin for $\forall\ |Y_j(l)|\in P_i\sum Y_j(l)=d$;

for $\forall\ |X_k(h)|\in P_i\sum X_k(h)=q$;

获得d和q的最小公倍数e;

$X=\frac{e}{d}Y_j+\frac{e}{q}X_k$;

删除Y_j和X_k;

end;

输出：与关联矩阵A相关的具有最小支持、包含目标变迁的T−不变量.

我们将这个算法应用到例1，见图2所示.经过过程2的计算，从矩阵C获得C_1，即t_5行被消除.经过过程3，C_1按E列元素割裂成两个矩阵C_2和C_3.再经过过程2，C_2变成C'_2和C_3变成C'_3.过程1分别对C'_2和C'_3计算不变量，可得：$X_1=\langle 1\ \ 1\ \ 1\rangle$(列$t_1\ t_2\ t_3$)，$Y_1=\langle 1\ \ 1\ \ 1\rangle$(列$t_2\ t_4\ t_7$)

扩充X_1和Y_1为包含所有变迁的向量，可得：$X'_1=X_1=\langle 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\rangle$(列$t_1\ t_2\ t_3\ t_4\ t_5\ t_6\ t_7$) $Y'_1=\langle 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1\rangle$(列$t_1\ t_2\ t_3\ t_4\ t_5\ t_6\ t_7$)，且有$\forall\ |X'_1(i)|\in E\sum X'_1(i)=1$和$\forall\ |Y'_1(i)|\in E\sum Y'_1(i)=2$.因此，可得原矩阵C的T−不变量：$X=2*X'_1+Y'_1=\langle 2\ \ 3\ \ 2\ \ 1\ \ 0\ \ 0\ \ 1\rangle$(列$t_1\ t_2\ t_3\ t_4\ t_5\ t_6\ t_7$).

不变量 X 包含着目标变迁 t_7，这样我们就证明了 Horn 子句从(1)到(6)蕴含着 $D \wedge E$. 同样的结果也可单独使用过程1获得，但显然要复杂多了.

$$C_1=\begin{array}{c} \\ t_1\\ t_2\\ t_3\\ t_4\\ t_6\\ t_7\end{array}\!\!\begin{array}{c}\begin{array}{cccc}A & B & E & D\end{array}\\ \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ -1 & -1 & 1 & 0\\ 0 & -1 & -1 & 1\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & -1\end{bmatrix}\end{array}\Rightarrow$$

$$C_2=\begin{array}{c} \\ t_1\\ t_2\\ t_3\\ t_6\end{array}\!\!\begin{array}{c}\begin{array}{ccc}A & B & D\end{array}\\ \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1 & -1 & 0\\ 0 & 0 & -1\end{bmatrix}\end{array}\Rightarrow C'_2=\begin{array}{c} \\ t_1\\ t_2\\ t_3\end{array}\!\!\begin{array}{c}\begin{array}{cc}A & B\end{array}\\ \begin{bmatrix}1 & 0\\ 0 & 1\\ -1 & -1\end{bmatrix}\end{array}$$

$$C_3=\begin{array}{c} \\ t_1\\ t_2\\ t_4\\ t_7\end{array}\!\!\begin{array}{c}\begin{array}{ccc}A & B & D\end{array}\\ \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -1 & 1\\ 0 & 0 & -1\end{bmatrix}\end{array}\Rightarrow C'_3=\begin{array}{c} \\ t_2\\ t_4\\ t_7\end{array}\!\!\begin{array}{c}\begin{array}{cc}B & D\end{array}\\ \begin{bmatrix}1 & 0\\ -1 & 1\\ 0 & -1\end{bmatrix}\end{array}$$

$$[I_3 \mid C'_2]=\begin{array}{c} \\ t_1\\ t_2\\ t_3\end{array}\!\!\begin{array}{c}\begin{array}{ccc:cc}\multicolumn{3}{c:}{} & A & B\end{array}\\ \begin{bmatrix}1 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 1\\ 0 & 0 & 1 & -1 & -1\end{bmatrix}\end{array}\Rightarrow \begin{array}{c} t_1+t_3\\ t_2\end{array}\!\!\begin{array}{c}\begin{array}{ccc:c}\multicolumn{3}{c:}{} & B\end{array}\\ \begin{bmatrix}1 & 0 & 1 & -1\\ 0 & 1 & 0 & 1\end{bmatrix}\end{array}\Rightarrow t_1+t_2+t_3[1\quad 1\quad 1]\cdots X_1$$

$$[I_3 \mid C'_3]=\begin{array}{c} \\ t_2\\ t_4\\ t_7\end{array}\!\!\begin{array}{c}\begin{array}{ccc:cc}\multicolumn{3}{c:}{} & B & D\end{array}\\ \begin{bmatrix}1 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & -1 & 1\\ 0 & 0 & 1 & 0 & -1\end{bmatrix}\end{array}\Rightarrow \begin{array}{c} t_2+t_4\\ t_7\end{array}\!\!\begin{array}{c}\begin{array}{ccc:c}\multicolumn{3}{c:}{} & D\end{array}\\ \begin{bmatrix}1 & 1 & 0 & 1\\ 0 & 0 & 1 & -1\end{bmatrix}\end{array}\Rightarrow t_2+t_4+t_7[1\quad 1\quad 1]\cdots Y_1$$

图2　例1T—不变量的求解过程

讨　论：由于篇幅限制，一阶谓词逻辑的高级 Petri 网模型就不在这里描述了，读者可阅[5]. 通过这篇文章的叙述，我们可以看到逻辑推理的 Petri 网模型可以表达形象思维，推论过程图象化为 Petri 网标识的流动过程. Petri 网的关联矩阵表示和不变量计算分析技术为逻辑推论提供了有力的分析和计算工具，另一面逻辑推理的某些规则又为 Petri 网 T—不变量的计算简化提供有益的启示.

在这篇文章里，我们仅局限于 Horn 子句的逻辑推论研究. 非 Horn 子句的 Petri 网模型不能直接用 T—不变量技术来检测其不一致性. 存在着 T—不变量只是一组子句不一致的必要条件，而不是充分条件(对一组 Horn 子句来说是充要条件). 解决非 Horn 子句 Petri 网模型求解问题是我们下一步的研究工作.

参考文献

1　R. Kowalski, Logic for Problem Solving, New York: Elsevier Science, 1979.

2　K. Lautenbach, On Logical and Linear Dependencies, Sankt Augustin, Germany, GMD Rep. 147, 1985.

3　A. Singachopoulos, Derivation of a Contradiction by Resolution Using Petri Nets, Petri Net Newsletter, Vol. 26, April 1987, 16—29.

4　G. Peterka and T. Murata, Proof Procedure and Answer Extraction in Petri Net Model of Logic Programs, IEEE Trans. on Software Engineering, Vol. 15, No. 2, Feb. 1989, 209—217.

5　C. Lin, A. Chaudhury, A. Whinston and D. C. Marinescu, Logical Inference of Horn Clauses in Petri Net Models, Technical Report, University of Texas, 1990.

6　T. Murata and D. Zhang, A Predicate—Transition Net Model for Parallel Interpretation of Logic Progams, IEEE Trans. on Software Engineering, Vol. 14, No. 4, April 1988, 481—497.

7　J. Martinez and M. Silva, A Simple and Fast Algorithm to Obtain All Invariants of Generalized Petri Nets, Informatik—Fachbrichte 52 (C. Girrault and W. Reising, eds.), Springer—Verlag. 1982, 301—303.