7H-2

ペトリネット解析を用いた循環ルールの検出

服部貴志　宮永喜一　栃内香次
北海道大学

1．はじめに

知識ベースシステムを構築する際の知識表現として，ルールモデルやフレームモデル，意味ネットモデル等が提案されている．なかでもルールモデルは知識表現形式が統一されており，個々のルールの意味付けが容易で，簡潔にシステムを記述できることから，広く利用されている．しかし一方では，知識ベース（ルール集合）が大規模化すると，ルール同志の関係が把握し難くなり，推論の制御や推論過程の理解にも困難を来す場合がある．

筆者らはこの様な問題点を解決するために，ルール集合をペトリネットによって表現し，そのネットワーク構造に着目してルール集合の構造化を試みた[1),2)]．本稿ではこの手法の応用として，ペトリネットの線形代数による解析方法を用いて，ルール集合の中から循環ルールを検出する方法について述べる．なお本稿では，前向き推論を行うＰＳ（Ｐroduction　Ｓystem）であるＯＰＳ５[3)]を対象とする．

2．ペトリネット[4),5)]によるＰＳのモデル化[2)]

ペトリネット構造は，プレース（place）とトランジション（transition）から構成されており，システムの実際の状態はプレース上のトークンの分布で表される．

ルール集合をペトリネット表現するに当たって，各構成要素は次のように対応する．

プレース	→	条件要素パタン，ルール名
トランジション	→	ルールの実行
トークン	→	WM（Working Memory）要素，ルールの制御

クラス名に対応するプレースは，該当するWM要素がWM内に存在すればトークンを持ち，ルール名に対応するプレースは，そのルールの実行が完了した時にトークンを持つ．各プレースとトランジションはルールの条件部と実行部によって決定される半順序関係によって結合される．例として，図１（ａ）のルール集合をペトリネット表現したものを図１（ｂ）に示す．

3．循環ルールの検出

ペトリネット構造を次の５項組で表現する．

$$M=(P,\ T,\ I,\ O,\ \mu)$$

Ｐ：プレース集合
Ｔ：トランジション集合
Ｉ：各トランジションの入力関数
Ｏ：各トランジションの出力関数
μ：マーキング集合

また次の式（１）で定義される生起行列Ｄを導入する．

$$D=[d(t,\ p)] \qquad (1)$$

$$d(t,\ p)=\begin{cases} w(p,\ O(t)) & \cdots\ p\in O(t),\ p\notin I(t) \\ -w(p,\ I(t)) & \cdots\ p\in I(t),\ p\notin O(t) \\ w(p,\ O(t))-w(p,\ I(t)) & \cdots\ p\in I(t),\ p\in O(t) \\ 0 & \cdots\ \text{その他} \end{cases}$$

ｗはアークの多重度を表す．この生起行列Ｄを用いると，あるマーキングμとμから到達可能であるマーキングμ'との関係は式（２）で表される．

$$\mu=\mu'+D^t\cdot\sigma \qquad (2)$$

ただしD^tはＤの転置行列，σは発火ベクトルと呼ばれる$|T|$次元ベクトルで，その第ｊ成分はμからμ'が生成されるまでのトランジションt_jの発火回数を表す．

次に式（２）を利用して，循環ルールを検出する方法について述べる．

（３－Ａ）複数ルールが循環する場合

複数ルールがサイクリックに発火し続ける場合は，

Detection of Cyclic Rules Using Petri Nets Analysis
Takashi Hattori, Yoshikazu Miyanaga, Koji Tochinai
Hokkaido University

2．で導入したペトリネットのうちルール名に対応するプレースのみを含むネットワークに式（2）を適用すれば検出できる．すなわち，この場合にはマーキングμが一定間隔で循環することになるので，

$$\mu = \mu + D^t \cdot \sigma$$

$$\therefore\ D^t \cdot \sigma = 0 \qquad (3)$$

式（3）において$\det D^t = 0$ならばσは非0の解を持つので，結局マーキングが循環する条件は

$$\det D^t = 0 \qquad (4)$$

例えば図1のルール集合に対しては，

$$\begin{array}{cc} & \begin{array}{cccc} s & r_1 & r_2 & r_3 \end{array} \\ D = & \left[\begin{array}{cccc} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & -1 \end{array}\right] \begin{array}{c} t_1 \\ t_2 \\ t_3 \\ t_4 \end{array} \end{array}$$

となり，$\det D^t = 0$である．

（3－B）1個のルールが無限回実行される場合

この問題はクラス名を表すプレースが有界であるか否かを判定する問題に還元されるので，次の定理[5)]を利用する．

<定理>ペトリネットMにおいて，$D \cdot \sigma = 0$の解のうち全ての成分値が正であるσ_sを持つための必要条件は，ペトリネットMが有界であることである．

従ってあるペトリネットが有界でないことを示すには，$D \cdot \sigma = 0$がσ_sなる解を持たないことを示せば良い．例として図2の様な場合を考えると，

$$\begin{array}{cc} & \begin{array}{ccc} s & x_1 & x_2 \end{array} \\ D = & \left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] t_1 \end{array}$$

$D \cdot \sigma = 0$の解を$\sigma_v = [v1\ v2\ v3]^t$とすると，v1，v2は任意，v3は0となり，明らかに全ての成分が正である解は存在しない．依ってこのペトリネットは有界ではなく，t_1は無限に発火する．

4．おわりに

本稿ではペトリネットを用いて，ルール集合の中から循環ルールを検出する方法を示した．今後は同様の方法を用いて，発火しないルールの検出及びネットの変形によるシステムの効率化を進める予定である．

5．参考文献

1）服部他：”プロダクションシステムのペトリネットによるモデル化”，昭和62年度電気関係学会北海道支部大会，pp.325,（1987）

2）服部他：”ペトリネットモデルによるルールの階層化”，第2回ディジタル信号処理シンポジウム，pp.177-182,（1987）

3）Lee Brownston, Robert Farrell, Elaine Kant, Nancy Martin：”Programming Expert Systems in OPS5”，Addison-Wesley,（1985）

4）J．L．Peterson著，市川／小林訳：”ペトリネット入門”，共立出版,（1984）

5）W．Reisig著，長谷川／高橋訳：”ペトリネット理論入門”，シュプリンガー・フェアラーク東京株式会社,（1988）

(a)

rule1 ： if x_0 then $-x_0$，$+x_1$
rule2 ： if x_1 then $+x_2$
rule3 ： if $x_1 x_2$ then $-x_2$，$+x_3$
rule4 ： if x_3 then $+x_1$

(b)

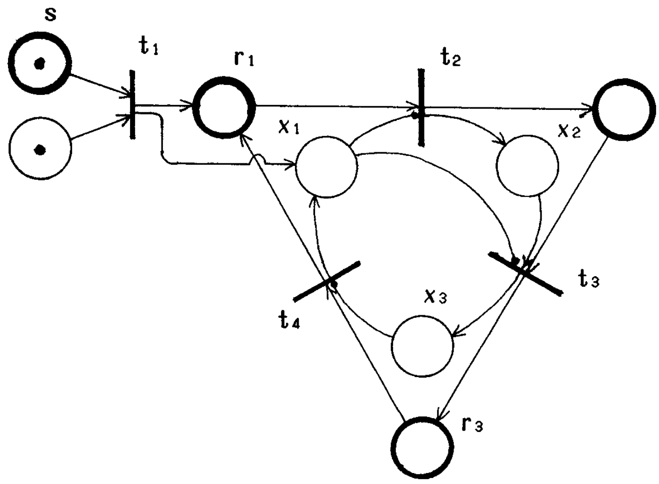

図1　複数ルールの循環

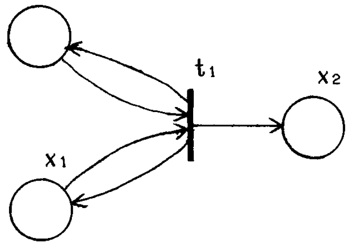

図2　1個のルールの無限発火