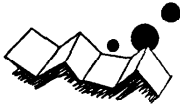


解説

ペトリネットの理論と応用†



山崎 秀記††

1. ま え が き

ペトリネット (Petri net) は、非同期的かつ並行的に動くシステムにおける情報の流れや制御を記述し解析するための数学的モデルである。1962年にPetriによって提案されていらい²⁵⁾、ペトリネットは広範囲に使用され、また広く研究されてきた。現在では、ヨーロッパにおいて、研究集会が毎年開催され、その予稿集も発行されている²⁾。

この解説は、ペトリネットの数学的、オートマトン理論的な性質に重点を置いた研究をできるだけ広く紹介しようとするものである。しかしペトリネットの研究は上に述べた側面だけでなく、多様な立場からなされている。とくにペトリの最初の論文は²⁵⁾、システムにおける情報の流れと制御の基本的性質を研究するものであった。このような方面からの研究、またペトリネットを用いて実際のシステムをモデル化する場合に起る様々な問題に関する研究等にはまったく触れられなかった。これらについては、前述の予稿集等^{3), 8)}を参考にしてほしい。

この解説では、2章でペトリネットの基本的な概念を述べ、3章ではペトリネットによるモデル化の例を示す。4~6章ではペトリネットの理論について、解析問題の種類、解析技法、問題の複雑さ等について述べる。7章では、ペトリネット言語の定義とそれらの性質を示す。8章ではペトリネットの部分クラスとその解析の問題を扱い、9章では、ペトリネットの拡張されたモデルについて簡単にふれる。

2. ペトリネットの定義

2.1 グラフによる表現

ペトリネット (Petri net) N は、場所 (place) p_1, p_2, \dots, p_n と遷移 (transition) t_1, t_2, \dots, t_m という2種類の節点を持つ2分有向グラフで表わされる。枝 (arc)

は場所から遷移へ向うか、遷移から場所へ向うかのどちらかで、場所から場所へや遷移から遷移へ向う枝はない。さらに各場所はいくつかの石 (token) を持つ。この石の配置をペトリネットのマーク付け (marking) といい、最初に与えられたマーク付けをとくに N の初期マーク付け (initial marking) という。

図-1にペトリネットの例を示す。場所は○で表し、その中に石を黒点・で入れ、遷移は棒|で表わす。なお、2本以上の枝が同じ節点から出て、同じ節点へ入っていてもよいことに注意せよ。(図-1では、 t_3 から p_3 へ2本の枝がある。) これを許さない定義もあるが、一般のペトリネットでは、このような多重の枝を許しても許さなくても、その表現能力に差のないことが知られており、ここでは多重枝を許すことにする。

さて、ペトリネットを構成する、場所、遷移、それらを結ぶ枝、初期マーク付けが与えられたとき、そのペトリネットにおける“計算”は次のように行われる。遷移 t は、各場所が t への枝の本数以上の石を持つとき、発火可能 (fireable) である。 t が発火 (fire) すると、ペトリネット中の石は枝にしたがって(増減しながら)場所から場所へ移動する。すなわち、 t への枝を持つ場所の石は、その枝の本数分だけ減り、 t からの枝を持つ場所の石はその枝の本数分だけ増える。

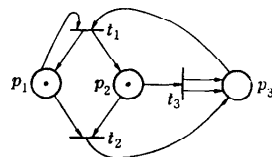


図-1 ペトリネット N

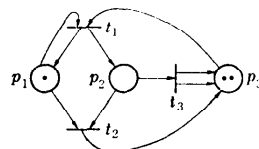


図-2 t_3 発火後のマーク付け

† Theory and Application of Petri Nets by Hideki YAMASAKI (Department of Information Science, Tokyo Institute of Technology).

†† 東京工業大学理学部情報科学科

例えば図-1 では、 t_2 と t_3 が発火可能で、 t_1 が発火すると、マーク付けは図-2 になる。さらに図-2 では t_1 だけが発火可能である。

2.2 ベクトルによる表現 (ベクトル置換系)

前節で、ペトリネット N の状態はマーク付けと呼ばれる石の配置で表わされる事を述べた。 N の場所が p_1, p_2, \dots, p_n の n 個あるとすると、そのマーク付けは自然数 $N(=\{0, 1, 2, \dots\})$ 上の n 次元ベクトル x で表わされる。つまり、ベクトル x の第 i 成分は場所 p_i が持つ石の数である。例えば、図-1 のマーク付けはベクトル $(1, 1, 0)$ で、図-2 のマーク付けはベクトル $(1, 0, 2)$ で表わす。

N の遷移 t_j は、 t_j の発火条件を示すベクトル u_j と、 t_j の発火によるマーク付けの変化を示すベクトル v_j との対で表わす。これらは共に整数 Z 上の n 次元ベクトルで、 u_j の第 i 成分は、場所 p_i から遷移 t_j への枝の数に負の符号を付けた数、 v_j の第 i 成分は、(t_j から p_i への枝の数) - (p_i から t_j への枝の数) である。したがってマーク付け x において、 $x + u_i \geq 0$ ならば、遷移 t_i は発火可能であり、 t_i を発火するとマーク付けは $x + v_i$ になる。

図-1 のペトリネットをベクトルを用いて表わすと、次のようになる。

- 初期マーク付け $x_0 = (1, 1, 0)$
- 遷移 $t_1: u_1 = (-1, 0, -1) \quad v_1 = (0, 1, -1)$
- $t_2: u_2 = (-1, -1, 0) \quad v_2 = (-1, -1, 1)$
- $t_3: u_3 = (0, -1, 0) \quad v_3 = (0, -1, 2)$

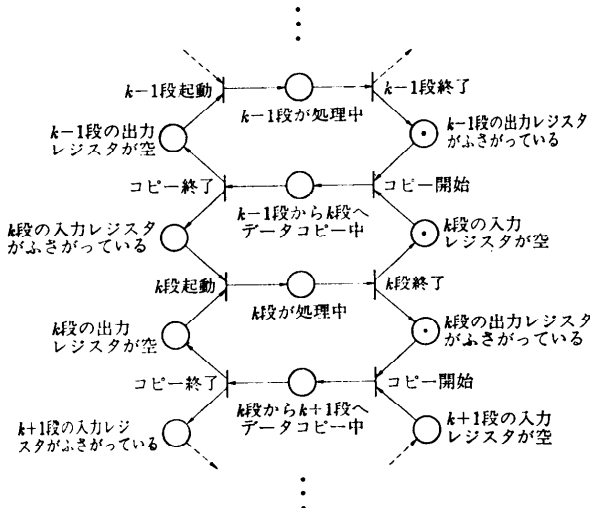


図-3 非同期的パイプライン制御装置を表わすペトリネット

なお、一般にこのように与えられたシステム、

$$(x, (u_1, v_1), (u_2, v_2), \dots, (u_m, v_m))$$

ただし、 $x \in N^n$, $u_i, v_i \in Z^n$, $u_i \leq v_i$ ($1 \leq i \leq m$), をベクトル置換系 (vector replacement system) という。したがって、ベクトル置換系は、ペトリネットのベクトルによる表現と見てよい。以後、この解説では、グラフによる表現や、ベクトルによる表現をとくに区別せずに用い、ペトリネット N の場所 p_1, p_2, \dots, p_n , 遷移 t_1, t_2, \dots, t_m , ベクトル x_0, u_i, v_i 等を固定して考える。

3. ペトリネットによるモデル化

ペトリネットは、構成要素が独立に、したがって非同期的かつ並行的に動くシステムをモデル化するために有用な道具である。この章では、簡単な例についてペトリネットによるモデル化を紹介する。

3.1 パイプライン制御装置

パイプラインと呼ばれる、計算機の次のような制御機構を考えよう。パイプラインは、丁度近代の工場における流れ作業のように、一連の処理が各段階に分れていて、各段は各々独立に、1つ手前の段からデータが送られると、それに適当な処理を施して次の段に送る、ということを繰り返す。

上記のパイプラインを非同期的に制御する機構を考える。各段階は各々入力レジスタと出力レジスタを持つものとする。各段は、その入力レジスタに値が入っていてかつ出力レジスタが空のとき、データを処理する。こうして出力レジスタに送られたデータは、次の段の入力レジスタが空くと、そこに移される。

これをペトリネットで表現したものを図-3 に示した。図-3のペトリネットでは、場所は条件を表わし、遷移は事象を表わす。そして、場所に石があることは、対応する条件が成り立っていることを示す。この例のように、ペトリネットは、適当な条件のもとで事象が発生し、また事象の発生によって条件が変化するという、事象と条件の相互作用をモデル化できる。なお、その場合、各場所の持つ石の数は高々1である。

3.2 生産者消費者問題

生産者消費者問題 (producer-consumer problem) では、生産者が“物”を次々に生産してバッファに貯め、一方消費者はバッファ

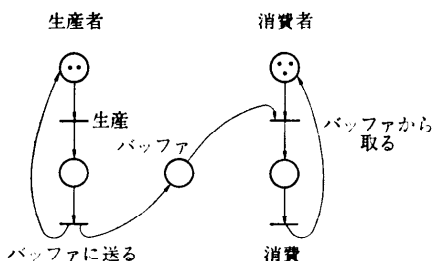


図-4 生産者消費者問題

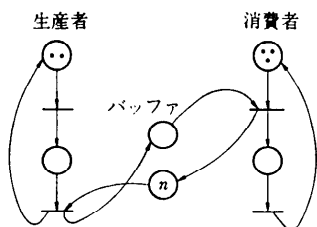


図-5 バッファの上限が n の生産者消費者問題

から次々と物を消費する。これをペトリネットモデル化したものが図-4である。

上の図-4は、生産者が2人居て、消費者が3人居る場合である。さらにバッファに上限 n があるときは、図-5のようにすればよい。ここに示した手法は、ある場所が持てる石の数に上限を設けたいときに有効な手法である。

3.3 食事する哲学者の問題

食事する哲学者の問題 (dining philosophers problem) は、Dijkstra⁶⁾ が提案した問題である。5人の哲学者が円形のテーブルを囲んで、中華料理を食べるのだが、はしが、各哲学者の間に1本ずつ(全部で5本)しかない。哲学者は食事と冥想を交互にくり返すが、全員が一斉に食事をしようとして不幸にも1本ずつはしを持ちあってしまうと、全員がもう一方のはしがあくのを永久に待ち続けなければならず(デッドロック, deadlock) 飢死してしまう。これをさけるためのひとつの解は、はしをつねに2本同時に持つようにする。ということで、そのための機構をペトリネットモデル化したのが、図-6である。

3.4 読み書き問題

読み書き問題 (reader-writer problem) は、いくつかのリードプロセス (reader process) やライトプロセス (writer process) がデータを共有している場合にプロセスをどう制御するかという問題である。

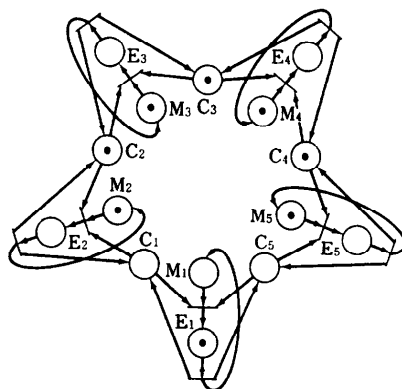


図-6 食事する哲学者の問題
Mi: 冥想, Ei: 食事, Ci: はし

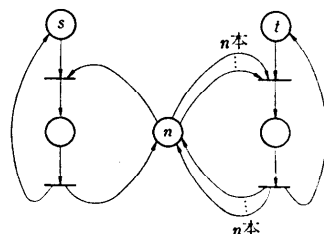


図-7 読み書き問題

リードプロセスは、データを変更しないので、ライトプロセスがどれもデータに触れていなければ、他のリードプロセスと同時にデータを読んでもかまわない。一方、ライトプロセスはデータを変更するので、他のプロセスはリードプロセスもライトプロセスも排除しなければならない。

リードプロセスが s 個、ライトプロセスが t 個ある場合の解を図-7に示した。図-7ではさらにデータを同時に読めるリードプロセスの数は n 個までという制限がついている。しかしリードプロセスがいくらかも増えて、しかもデータを同時に読めるリードプロセスの数に制限がないシステムは、ペトリネットではモデル化できない。それは、現在データを読み込み中のリードプロセスの数を石の数で覚えておく必要があり、そのための場所を作ったとしても、その場所に石がなければ発火してよい(ライトプロセスを)という条件はペトリネットでは表わせないからである。このことから、ペトリネットにより強力な機能を付けて、そのモデル化能力を増す方面の研究も行われている(9章参照)。

4. ペトリネットの理論における解析の問題

システムをペトリネットでモデル化したとして、次の目標は、システムが望ましい性質を持っているか否かを、それを記述するペトリネットの性質を調べて、導き出すことである。この章では、ペトリネットの解析において取り上げられる基本的な用語と問題を解説し、次の章で解析法を解説する。しかし、ペトリネットに関する解析の問題には非常に難しいものや決定不能なものが多い。

4.1 到達可能集合

ペトリネット N のマーク付け x 上で発火可能な遷移 t_i を発火して、マーク付け y がえられるとき、 t_i と書く。 $x_1 \xrightarrow{t_{i_1}} x_2 \xrightarrow{t_{i_2}} \dots \xrightarrow{t_{i_l}} x_{l+1}$ のとき、 $x_1 \xrightarrow{t_{i_1} \dots t_{i_l}} x_{l+1}$ と書き、 x_{l+1} は x_1 から発火列 $t_{i_1} \dots t_{i_l}$ によって到達可能 (reachable) であるという。とくに発火列を指定する必要がなければ、 $x_1 \xrightarrow{*} x_{l+1}$ と書く。また、 $l=0$ のとき、 x_1 は x_1 から到達可能であり、 $x_1 \xrightarrow{\lambda} x_1$ (λ は空列) と書く。

ペトリネット N の初期マーク付け x_0 から到達可能なマーク付けの集合、 $\{y | x_0 \xrightarrow{*} y\}$ を N の到達可能集合 (reachability set) といい、 $R(N)$ と書く。

図-1 のペトリネット N では、例えば、 $(1, 1, 0) \xrightarrow{t_1} (1, 0, 2) \xrightarrow{t_1} (1, 1, 1) \xrightarrow{t_1} (1, 2, 0)$ となる。また、 N の到達可能集合は、 $\{(1, m, n) | m=0 \text{ ならば } n \geq 2\} \cup \{(0, m, n) | n \geq 1\}$ である。

さて、 $x \xrightarrow{t_{i_1} \dots t_{i_l}} y$ のとき、 $y = x + \sum_{k=1}^l v_{i_k}$ であることに注意しておこう。(v_{i_k} は t_{i_k} の発火によるマーク付けの変化を表わすベクトル。) したがって、発火列 $t_{i_1} \dots t_{i_l}$ とその遷移を適当に並べ換えてえられる発火列 $t_{i'_1} \dots t_{i'_l}$ が共に x で発火可能だとすると、それらが発火させた結果は同じマーク付け y になる。

4.2 遷移の生死

遷移 t がマーク付け x で死んでいるというのは、 t が x から到達可能なすべてのマーク付け (x を含む) で発火可能でないことをいう。マーク付け x で死んでいる遷移は、以後決して発火することがない。したがって、例えば初期マーク付けで死んでいる遷移は、ペトリネットから除いても、到達可能集合に影響しない。

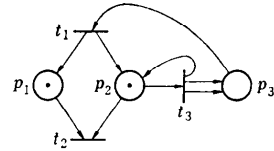


図-8 ペトリネット

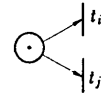


図-9 衝突の例

すべての遷移が死んでいるマーク付けをデッドロック (deadlock) という。したがってデッドロックにおいては、どの遷移も発火可能でない。

一方、遷移 t がマーク付け x で生きている (live) というのは、 x から到達可能なすべてのマーク付けにおいて t が死んでいないことである。これは x で t が死んでいないこととは異なることに注意しよう。ペトリネットの初期マーク付けにおいて、遷移 t が生きているとき、単に t は生きているという。

例として上の図-8 のペトリネットを考えよう。ここでは、 t_1, t_2, t_3 は死んでいないが生きてもいない。なぜなら、最初 $(1, 1, 0)$ で t_2 が発火するとマーク付けは $(0, 0, 0)$ となり、これはデッドロックである。それに対し、マーク付け $(0, 1, 0)$ ではすべての遷移が生きている。なぜなら $(0, 1, 0)$ から到達可能なマーク付けでは p_2 と p_3 の石の数が共に 0 になることはないので、発火列、 $t_1 t_3 t_2$ 又は $t_3 t_1 t_2$ がいつも発火可能である。

遷移の間の衝突という概念は、ペトリネットの基本的な概念である。マーク付け x において 2 つの遷移 t_i と t_j がともに発火可能で、かつ t_i を発火すると t_j が発火可能でなくなるか t_j を発火すると t_i が発火可能でなくなるかするとき、 x で t_i と t_j は衝突 (conflict) しているという (図-9)。すべての到達可能なマーキングにおいて衝突がおきないとき、そのペトリネットは持続的 (persistent) であるという。これは、1度発火可能になった遷移 t_i は、 t_i 自身を発火しない限りずっと発火可能であり続けることを意味している。例えば図-8 では、 t_2 と t_3 が衝突しており、したがってこのペトリネットは持続的でない。

4.3 解析の問題

ペトリネットにおける解析の問題をいくつか上げてみよう。

(1) 与えられたペトリネットは安全か
 到達可能なマーク付けがすべてベクトル $\mathbf{1}=(1, 1, \dots, 1)$ 以下のとき、そのペトリネットは安全 (safe) であるという。この問題は決定可能である (5.1 節)。

(2) 与えられたペトリネットは有界か
 到達可能集合が有界、すなわちある k が存在して、到達可能なマーク付けがすべて $k \cdot \mathbf{1}$ 以下のときそのペトリネットは有界 (bounded) であるという。この問題も決定可能である (5.1 節)。

(3) 与えられたペトリネットは保存的か
 ペトリネット中の石の総数が遷移の発火によって不変のとき、そのペトリネットは保存的 (conservative) であるという。この問題も決定可能である (5.2 節)。

(4) 等価問題 (equivalence problem)
 2つの与えられたペトリネットの到達可能集合が等しいか否かという問題は決定不能である¹³⁾。

(5) 到達可能問題 (reachability problem)
 与えられたペトリネットが与えられたマーク付けに到達可能か否かを判定する問題である。最近解かれたが、その判定アルゴリズムは非常に複雑である^{17), 20), 26)}。

(6) 生きの問題 (liveness problem)
 与えられたペトリネットにおいて、与えられた遷移が生きているか否かを問う問題で、これは到達可能問題に帰着できることが知られている¹¹⁾。

(7) 遷移の死にの問題
 与えられた遷移が与えられたマーク付けで死んでいるか否かを判定する問題である。これは次の被覆可能問題の一種として形式化でき、決定可能である。

(8) 被覆可能問題 (coverability problem)
 ペトリネット N とマーク付け α が与えられたとき、 N の到達可能集合 $R(N)$ の中に α 以上のマーク付けがあるか否かを判定する問題であり、これは決定可能である (5.1 節)。とくに α として遷移 t_i の発火条件を示すベクトル $(-u_i)$ をとれば、この問題は遷移 t_i が死んでいるか否かを判定する問題になる。

5. 解析技法

この章では、ペトリネットの解析において最も基本的な2つの道具、到達可能木と線形代数的手法を解説する。

5.1 到達可能木¹⁶⁾

ペトリネットの到達可能集合がもし有限ならば、それらを節点に持つ有限木の形に表わすことができる。

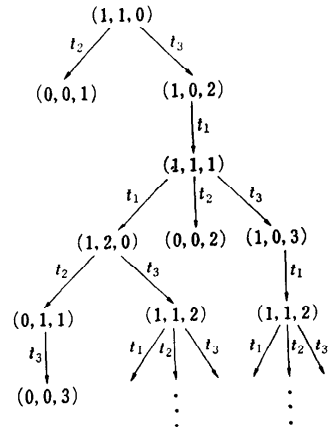


図-10 図-1 のペトリネットの到達可能マーク付けの木

しかし一般には到達可能集合は有限とは限らない。図-1 のペトリネット N の到達可能集合を木の形に書いたものが図-10 である。

図-10 の木では、各節点にマーク付けのラベルが付き、節点を結ぶ枝には、そのマーク付けの変化を引き起こした遷移のラベルが付いている。木の根に初期マーク付けを付け、各節点から、そのマーク付けで、発火可能な遷移に対応する枝を伸ばしてゆけば、図-10 がえられる。図-10 は無限の木であり、それを有限的に表現するために、特別な記号 ω を導入する。ペトリネットの1つの場所の石の数をいくらでも大きくできるときには、それを ω で表わす。すなわち、初期マーク付けから始めて、木を次々と構成してゆく際、木の中に自分より小さなベクトルを持つ先祖があれば、真に大きくなっている成分は ω に書き換える (その間の発火列は繰り返し発火可能で、そうすれば、 ω で書き換えられた成分はいくらでも大きくできる)。そして、 ω を含むベクトルに書きかえた節点からも、発火可能な遷移があれば発火させて木を構成してゆく。 (ω はいくらでも大きくなる値を表わす記号なので $\omega \pm n = \omega, n < \omega$ (n は自然数) とする。) ある節点に対し、発火可能な遷移がひとつもないか、または同じマーク付けがすでに現れていて検査済みならば、その節点からはもう枝を伸ばさない。こうしてえられるのが図-11 である。

上述のように構成した木は、常に有限となることが証明されている¹⁶⁾。この木をペトリネットの到達可能木 (reachability tree) という。

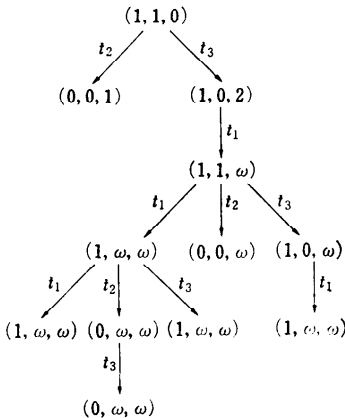


図-11 図-1のペトリネットの到達可能木

図-11には、例えばラベル $(1, 1, \omega)$ を持つ節点があるが、これは、任意の自然数 i に対して、 i より大きい自然数 j が存在して、マーク付け $(1, 1, j)$ に到達可能であることを意味している。したがって、ペトリネット N の到達可能木から、 N が安全か、有界か、与えられた x より大きなマーク付けに到達可能か (被覆可能問題) 等の問題に答えることができる。

5.2 線形代数的手法

ベクトルによるペトリネット N の表現を思い出そう。遷移 t_1, t_2, \dots, t_m の発火によるマーク付けの変化を表わす n 次元ベクトル v_1, v_2, \dots, v_m を縦に並べた、 $m \times n$ 次元の行列を V とする。この行列をペトリネットの付随行列 (incidence matrix) という。発火列 σ に対し、中の遷移 t_j の出現回数を第 j 成分とする m 次元ベクトルを $\phi(\sigma)$ と書く。(ϕ は Parikh 写像として知られている。) 例えば、 $\phi(t_1 t_2 t_1 t_4) = (2, 1, 0, 1)$ である。別の言い方をすると、 $\phi(t_j) = e_j$ (e_j は第 j 成分だけが1で他はすべて0の単位ベクトル) であり、 $\phi(t_{j_1} \dots t_{j_i}) = \sum_{k=1}^i \phi(t_{j_k}) = \sum_{k=1}^i e_{j_k}$ である。すると、 $x \xrightarrow{\sigma} y$ のとき、 $y = x + \phi(\sigma)V$ となる。

このことから、マーク付け y がマーク付け x から到達可能なら、ある m 次元ベクトル a が存在して、 $y = x + aV$ と書ける。したがって、ペトリネット N の初期マーク付けを x_0 とすると、 N の到達可能集合 $R(N)$ は $\{y \in N^* \mid \exists a \in N^m \ y = x_0 + aV\}$ の部分集合である。これによって到達可能集合の大まかな近似がえられる。例えば図-1 の例では、

$$\begin{aligned} & \left\{ y \in N^* \mid \exists a \in N^3 \ y = (1, 1, 0) + a \begin{pmatrix} 0 & 1 & -1 \\ -1 & -1 & 1 \\ 0 & -1 & 2 \end{pmatrix} \right\} \\ &= \{(1 - a_2, 1 + a_1 - a_2 - a_3, -a_1 + a_2 + 2a_3) \in N^3 \\ & \quad \mid a_1, a_2, a_3 \geq 0\} \\ &= \{(1, i, i') \mid i, i' \geq 0, i = 0 \text{ なら } i' \geq 2\} \\ & \quad \cup \{(0, j, j') \mid j, j' \geq 0\} \end{aligned}$$

これは、実は N の到達可能集合と一致しているが、もちろん一般には一致するとは限らない。

ペトリネット N の付随行列 V の性質を調べることによって、初期マーク付けにはよらない、ペトリネットの構造的な性質が調べられる。例えば前節で挙げた、保存的という性質は、 $IV' = 0$ という式で表わせる (V' は V の転置行列、 $0 = (0, \dots, 0)$)。より一般的に、場所に適当な重みを付ける重みベクトル w で、 $wV' = 0$ となる w はペトリネットの解析に重要な役割を果たす¹⁹⁾。

もうひとつの例として、ペトリネットがすべての初期マーク付けに対して有界 (強有界 (strongly bounded) という) か否かを判定する問題を考えよう。ある $a \in N^m$ が存在して、 $aV \geq 0$ となれば、強有界ではない。($\phi(\sigma) = a$ である発火列 σ に対し、 σ が発火可能な初期マーク付けが必ず存在するので。) 逆に強有界でなければ $aV \geq 0$ なるベクトル a が存在する。したがって強有界か否かの判定問題は $aV \geq 0$ なる $a \in N^m$ が存在するか否かを判定する問題に帰着する。

6. 解析の複雑さ

この章では、4章で定義したいくつかの問題について、その解析の複雑さを概観する。たとえば、与えられたペトリネットが有界だとしても、マーク付けはどの位大きくなる可能性があるか、といった問題を考えて見よう。

6.1 弱計算機

ペトリネットに次のような意味で、関数の計算をさせることができる。場所 p_1, p_2, \dots, p_n のうち、 p_1, p_2, \dots, p_i に各々 x_1, x_2, \dots, x_i 個の石を置いてペトリネットを実行させたとき、 p_{i+1} が持つ石の最大数が $f(x_1, \dots, x_i)$ ならば、このペトリネットは i 変数関数 $f(x_1, \dots, x_i)$ を弱計算するという。

図-12、図-13 は各々加算、乗算の弱計算機である。図-12、図-13 は共に有界なペトリネットで、場所 p_3 の石の最大数が各々 $x + y, xy$ となる。これらを組

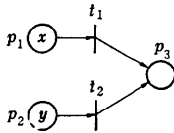


図-12 弱加算機

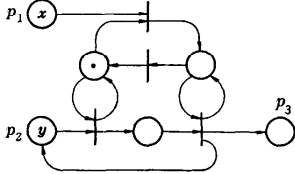


図-13 弱乗算機

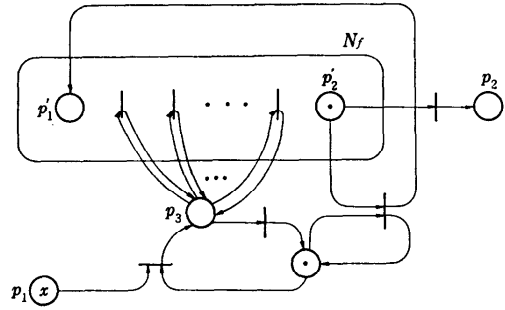


図-14 N_f から N_0 の構成

p'_1 と p'_2 は各々 N_f の入力用、出力用の場所である。
 p_3 は N_f に対するスイッチの役目を持つ。即ち N_f の遷移は p_3 に石があるときしか発火できない。

み合わせると、任意の多項式が計算できる。この事を利用して、Hack は Hilbert の第 10 問題をペトリネットの等価問題に帰着させ、等価問題が決定不能であることを示した¹³⁾。

多項式より複雑な関数をペトリネットでは弱計算できないのだろうか。実は $f(x)$ が単調増加 ($x < x'$ なら $f(x) < f(x')$) で弱計算可能なら、 $g(x) = f^*(1)$ で定義される関数も単調増加でかつ弱計算可能である。 $f(x)$ を弱計算するペトリネット N_f から $g(x)$ を弱計算する N_g を構成するには図-14 のようにする。

$f_1(x) = 2x$ (これは弱計算可能な関数) とすれば、
 $f_2(x) = f_1^*(1) = 2^2, f_3(x) = f_2^*(1) = 2^{2^2}$ } x 個、... も弱計算可能である。 $f_1(x) = 2x, f_{n+1}(x) = f_n^*(1), (n \geq 1)$ とし、 $A_n = f_n(1)$ と定義する。例えば、 $A_1 = 2, A_2 = 2, A_3 = 16$ 。すると数列 A_n はどんな原始帰納関数でも抑えられない。一方、任意の n に対し、 $f_n(x)$ を弱計算する、 n に比例するサイズのペトリネットが存在する。これはサイズが n (場所、遷移、初期マーク付けの石の数がすべて n 以下) の有界なペトリネットの到達可能なマーク付けの石の数が $A_n = f_n(1)$ のオーダーになりうることを示している。したがって、5章で解説した到達木のサイズのオーダーは原始帰納関数では抑えられない。

7. ペトリネット言語

この章では、ペトリネットの言語理論的な研究を紹介しよう。これはペトリネットではモデル化したシステムにおいて、起りうる動作列全体を、そのペトリネットの生成する言語 (記号列の集合) として扱えようとするものである。

ペトリネット言語を考える際には、その遷移にラベルを付けるのが一般的である。ラベル付け関数 $\gamma: T \rightarrow \Sigma \cup \{\lambda\}$ としてどのような関数を考えるかによって、いくつかの異なるペトリネット言語が定義できる。

(1) 1対1ラベル付け関数 $\gamma: T \rightarrow \Sigma$ で $t_i \neq t_j$ ならば $\sigma(t_i) \neq \sigma(t_j)$

これは、異なる遷移はすべて異なる動作に対応すると考える場合である。

(2) λ なしラベル付け関数 $\gamma: T \rightarrow \Sigma$

これは、同一の動作が異なる環境のもとで起りうるので、それらは異なる遷移でモデル化してよいという考えに基づいている。したがって異なる遷移が同じラベル (動作名) を持つことがある。

(3) λ を許したラベル付け関数 $\gamma: T \rightarrow \Sigma \cup \{\lambda\}$

さらに、より一般的に、モデル化の過程でシステムの具体的な動作には対応しない余分な遷移をペトリネットに導入してしまうこともありうる。そこでそのような遷移には空列 λ をラベル付けすることも許す。

一方、ペトリネット言語を定義するもうひとつの要素、受理状態 (受理マーク付け) の集合 F の定義にもいくつかの種類がある。

(1) L-型: F は有限集合

(2) G-型: 有限集合 F' が与えられ、 $F = \{x \mid \exists y \in F', y \leq x\}$

(3) T-型: F はデッドロック (どの遷移も発火可能でないマーク付け) の集合

(4) P-型: $F = N^*$, すなわち、 F はすべてのマーク付けの集合

ペトリネット N とそのラベル付け関数 γ , 受理マーク付けの集合 F が与えられると、これらが定義する

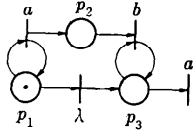


図-15 ペトリネット
(遷移の名前 (a, ...) は省略した. a, b, λ 等はラベルである.)

1対1ラベル付け関数	$P_f \subseteq G_f$	$L_f \subseteq T_f$
λなしラベル付け関数	$P \subseteq G \subseteq L \supseteq T$	
λも許したラベル付け関数	$P_\lambda \subseteq G_\lambda \subseteq L_\lambda = T_\lambda$	

図-16 ペトリネット言語のクラスの包含関係

言語は, $\{\gamma(\sigma) \exists y \in F, x_0 \rightarrow y\}$ である. 例として図-15のペトリネットで, 4通りの受理マーク付け集合の違いによるペトリネット言語の違いを見てみよう.

$F = \{(0, 0, 1)\}$ とする L-型言語は $\{a^n b^n | n \geq 0\}$, $F = \{x | x \geq (0, 0, 1)\}$ とする G-型言語は $\{a^m b^n | m \geq n \geq 0\}$, T-型言語は $\{a^m b^n a | m \geq n \geq 0\}$ ($F = \{(0, n, 0) | n \geq 0\}$ である), 最後に P-型言語は $\{a^m b^n, a^m b^n a | m \geq n \geq 0\}$ である.

さて, ラベル付け関数で3通り, 受理マーク付けの集合で4通り, 計12通りの異なるペトリネット言語のクラスの定義がえられた. これらのクラス間の包含関係を図-16に, 種々の演算のもとでの閉包性を図-17に示した^{6), 24)}.

ペトリネット言語の各クラスはまだ十分分っていない部分が多い. 一方このようなやり方は, もともと並行的に動くシステムのモデルであるペトリネットに対し, すべての動作を順次的に表現してしまうことにより, 全順序を不正確に導入してしまうおそれのあることにも注意しなければならない²²⁾.

正則表現を拡張して, 並行的システムの動作列を記述するために, フロー表現が Shaw²⁷⁾ によって提案された. フロー表現とペトリネット言語との関連も調べられている²⁾.

8. ペトリネットの部分クラスと Presburger 式

ペトリネットの解析の問題, 特に等価問題が決定不能であるところから, 実用的には適当なモデル化の能力を持ち, かつ解析がより易くなるような, ペトリネットの部分クラスが考えられている.

	U	∩	C	·	Δ	R
$L_\lambda = T_\lambda$	○	○	×	○	○	○
G_λ	○	○	×	○	○	×
P_λ	○	○	×	○	○	×
L	○	○	○	○	○	○
G	○	○	×	○	○	×
T	×	○	×	○	○	○
P	○	○	×	○	○	×
L_f	×	○	×	×	×	×
G_f	×	○	×	×	×	×
T_f	×	○	×	×	×	×
P_f	×	○	×	×	×	×

図-17 種々の演算のもとでの閉包性

U: 和集合, ∩: 共通部分, C: 補集合
 ·: 連接, $A \cdot B = \{xy | x \in A, y \in B\}$
 Δ: シャッフル $A \Delta B = \{x_1 y_1 \dots x_n y_n | x_1 \dots x_n \in A, y_1 \dots y_n \in B\}$
 R: 逆転 $A^R = \{a_n \dots a_1 | a_1 \dots a_n \in A\}$

○印はそのクラスがその演算に関して閉じていることを×印は閉じていないことを表す. クラスLが補集合で閉じているか否かはまだわかっていない.

これらの部分クラスでは, 到達可能集合と, 発火可能列の集合の Parikh 写像による像を Presburger 式と呼ばれる整数上の1階述語論理式で定義できる事が多く, それゆえに種々の決定問題が決定可能になることが多い. この章ではまず Presburger 式を紹介したあと, ペトリネットの種々の部分クラスを紹介しよう.

8.1 Presburger 式

整数の集合 Z 上加算+, 減算-, 等号=, そして大小関係≤が与えられているものとする. (乗算や除算は除く.) これに整数を値域とする変数 x, y, z, \dots , 変数に対する量化記号 (\forall, \exists), 論理記号 \wedge (and), \vee (or), \neg (not) 等を許した1階述語論理の式を Presburger 式という. 例えば $p_1 \equiv \forall x \exists y \exists z (x \leq y) \wedge (x \leq z) \wedge (2y = 3z)$ や $p_2(x_1, x_2) \equiv \exists y (x_1 = 2y) \vee (x_2 = 3y + 1)$ 等は Presburger 式である. (簡単のため, $y + y$ を $2y$ と書いた. $3y, 3x$ も同様である.) 式 p_1 のように変数がすべて量化記号で束縛されている式の真偽は決定可能である⁷⁾. 一方式 $p_2(x_1, x_2)$ のように自由な(即ち束縛されていない)変数 x_1, \dots, x_n を持つ式 $p(x_1, \dots, x_n)$ の真偽は一意に定まらない. 以後簡単のため $p(x_1, \dots, x_n)$ を $p(x)$ と書こう. ($x = (x_1, \dots, x_n)$) 式 $p(x)$ を成り立たせる $x \in Z^n$ の集合を p が定義する集合といい, $T(p(x))$ と書く. 例えば $T(p_2(x_1, x_2)) = \{(0, 1), (2, 4), (-2, -2), \dots\}$ である.

自由変数を含まない Presburger 式の真偽が決定可

能なので、自由変数を含む式 $p(x)$, $p'(x)$, $\alpha \in \mathbb{Z}^n$ に対し $T(p(x))$ は ϕ か (これは $\exists x p(x)$ が偽と同値), $T(p(x))=T(p'(x))$ か (これは $\forall x p(x) \leftrightarrow p'(x)$ が真と同値), $\alpha \in T(p(x))$ かといった問題はすべて決定可能である。

さて、ペトリネット N の到達可能集合 $R(N)$ は次の式をみたす最小の集合に等しい。

$$x_0 \in R \wedge \forall x \geq 0 \bigwedge_{i=1}^m [x \in R \wedge x \geq u_i \rightarrow x + v_i \in R] \tag{1}$$

(N の初期マーク付けは x_0 , 遷移は t_1, \dots, t_m .)

N の発火可能な遷移列の Parikh 写像による像全体の集合を $W(N)$ としよう。すなわち, $W(N) = \{\phi(\sigma) \mid \exists x, x_0 \rightarrow x\}$ とする。すると到達可能集合 $R(N)$ は, $R(N) = \{x_0 + tV \mid t \in W(N)\}$ (V は N の付随行列) と書けるので, $W(N)$ もペトリネットの解析に重要な役割を果たす。

ところで $W(N)$ は次の式をみたす集合として定義できる。

$$\begin{aligned} 0 \in W(N) \wedge \forall x \geq 0 [x \in W(N) \\ \leftrightarrow \exists y \in W(N) \bigvee_{i=1}^m [y + e_i = x \wedge x_0 + yV + u_i \geq 0]] \end{aligned} \tag{2}$$

(1), (2)式は Presburger 式ではないが, R や $W(N)$ を Presburger 式 $p(x)$ でおきかえた式は Presburger 式となり決定可能である。このことから, 任意の Presburger 式 $p(x)$ に対し, $R(N) \subseteq p(x)$ か否か, $W(N) = p(x)$ か否かという問題は決定可能であることが言える。

したがって, 例えば $W(N)$ が Presburger 式で定義可能な事が証明できれば (具体的に式が求まらなくても), すべての Presburger 式に対し, (2) が成り立つか否かを片っ端から調べることによって, $W(N)$ を定義する Presburger 式が具体的に求まる。

なお, Presburger 式で定義可能な自然数上のベクトルの集合のなすクラスは準線形集合 (semilinear set) のなすクラスと一致していることが知られている⁶⁾。したがって“Presburger 式で定義可能な集合”という文は“準線形集合”と読み換えてもよい。また $W(N)$ が Presburger 式で定義可能なら, 到達可能集合 $R(N)$ も Presburger 式で定義可能である。

8.2 5次元ループフリーネット^{14), 29)}

場所の数 n が 5 以下で, ループ(図-18)を持たないペトリネット N の到達可能集合 $R(N)$ は Presburger



図-18 ペトリネットのループ

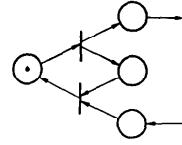
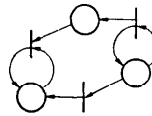
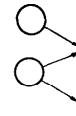


図-19 マーク付きグラフ



(a) 自由選択ネット



(b) 許されない形

図-20

式で定義可能であり, また等価問題も含めて, 種々の問題が決定可能であることを Hopcroft と Pansiot が示した¹⁴⁾。しかし場所の数が 6 以上だと, $R(N)$ が Presburger 式で定義できない例が知られている。

8.3 マーク付きグラフ⁴⁾

各場所 p が p へ入る枝と p から出る枝を 1 本ずつ持つペトリネットをマーク付きグラフ (marked directed graph) という。(図-19) マーク付きグラフ N では遷移間の衝突は決して起きないので, $R(N)$ や $W(N)$ は Presburger 式で定義可能である (8.4 参照)。また, マーク付きグラフのサイクルの中にある石の総数は, 遷移の発火によって変わらない。さらにマーク付け x と y において, 各サイクル中の石の総数が正で各々一致しているなら, y は x から到達可能であり逆に x が y から到達可能でもある。これはマーク付きグラフでは, x が各サイクル中に石がある初期マーク付け x_0 から到達可能なら, x_0 は x から到達可能であることを意味する。

8.4 持続的ペトリネット^{9), 18), 30)}

ペトリネット N が持続的なとき, $R(N)$ や $W(N)$ は Presburger 式で定義可能である。このことは, $f = (f_1, f_2, \dots, f_m)$ $g = (g_1, g_2, \dots, g_m)$ が共に $W(N)$ に属していれば, f と g の各成分の大きい方を取ったベクトル $(\max(f_1, g_1), \max(f_2, g_2), \dots, \max(f_m, g_m))$ も $W(N)$ に属していることから証明できる。なおペトリネットが持続的か否かの判定問題も, 決定可能で

ある。

8.5 自由選択ネット¹⁰⁾

自由選択ネット (free choice net) は、衝突は許すけれども、その起り方に制限を付けたペトリネットである。多重枝のないペトリネットで p_i から t_j への枝があるとき、場所から t_j への枝がそれ1本しかないか、 p_i から遷移への枝がそれ1本しかないならば、そのペトリネットを自由選択ネットという。(図-20) 互に衝突する遷移同志は、唯一つの共通の場所以外からの枝は持たず、したがってそれらは同時に発火可能になる。自由選択ペトリネットではすべての遷移が生きているための簡単な必要十分条件がわかっている、その問題は決定可能である。しかしその到達可能集合は Presburger 式で定義できるとは限らず、まだよく調べられていない。

8.6 リバーシブルネット¹¹⁾

ペトリネット N がリバーシブル (reversible) であるというのは、初期マーク付けが、任意の到達可能なマーク付けから到達可能なときである。したがって、リバーシブルネットの到達可能集合では、常に一方から他方へ到達可能である。システムにこのような制限を付けるのは実用上も意味のあることと思われる。

リバーシブルネットの到達可能集合を定義する Presburger 式は具体的に求まる。したがってこのクラスに関する種々の解析問題はリバーシブルか否かの判定問題も含めて決定可能である。

上の結果は一般のペトリネット N に対して、初期マーク付け x_0 からそれ自身への発火系列上に現れるマーク付の集合 $C(N)$ 、すなわち

$$C(N) = \{x \mid \exists \sigma, \exists \sigma' \quad x_0 \xrightarrow{\sigma} x \xrightarrow{\sigma'} x_0\}$$

も Presburger 式で定義可能であることを用いて証明される。 $C(N)$ が到達可能集合 $R(N)$ と等しければ、ペトリネットはリバーシブルである。また、一般のペトリネットに対し、 $C(N) = R(N')$ となるリバーシブルペトリネット N' を求める手続きも知られている¹²⁾。

9. ペトリネットの拡張

3.4 節のモデル化の例で示したように、ペトリネットのモデル化の能力に限界があった。この章ではペトリネットを拡張してそのモデル化能力を高めたものとして禁止枝付きのペトリネットを石に色を付けたペトリネットを紹介する。

ペトリネットをモデル化能力の点から見た場合の基本的な問題は、“場所に石がない”ということが調べ

られないことである。このことからペトリネットに禁止枝 (inhibitor arc) を付けたモデルが考えられた²⁴⁾。場所 p_i からの禁止枝を持つ遷移は、通常の条件の他に、禁止枝のある場所 p_i の石が空のときのみ発火可能である。

禁止枝付きのペトリネットは任意のチューリング機械を模倣できてしまう。したがって、解析の問題の多くは決定不能である。

一方、石に色を付けることによって、モデル化が易しくなる場合も多い。このようなペトリネットをカラーペトリネット (colored Petri net) と言う^{15), 23)}。カラーの数が有限の場合、カラーペトリネットは一般のペトリネットで模倣できる。したがって、モデル化能力の点から言えば、一般のペトリネットと変わらないが、より簡潔な表現が可能になる点で有用であろう。

参 考 文 献

以下に上げた参考文献のうち、3), 22), 24) は、ペトリネットの理論の全体を眺めるうえで参考になる。またこの文献表は全体を網羅したものではなく、選択にも偏りがあると思われるので、上に挙げた文献の文献表等も参考に頂ければ幸いである。

- 1) Araki, T. and Kasami, T.: Decidable Problems on the Strong Connectivity of Petri Net Reachability Sets, *Theor. Comput. Sci.*, Vol. 4, No. 1, pp. 99-119 (1977).
- 2) Araki, T., Kagimasa, T. and Tokula, N.: Relations of Flow Languages to Petri Net Languages, *Theor. Comp. Sci.*, Vol. 15, No. 1, pp. 51-75 (1981).
- 3) Brauer, W. (ed.): *Net Theory and Applications*, Lecture Notes in Comput. Sci., Vol. 84, Springer-Verlag (1980).
- 4) Commoner, F. and Holt, A.W.: Marked Directed Graphs, *J. Comput. Sys. Sci.*, Vol. 5, pp. 511-523 (1971).
- 5) Dijkstra, E.: Cooperating Sequential Processes, in F. Genuys (ed.), *Programming Languages*, Academic Press, New York (1968).
- 6) Fujikawa, W.: Some Decision Problem for Vector Addition Systems and Petri Nets, 修士論文, 東工大 (1979).
- 7) Ginsburg, S. and Spanier, E.H.: Semigroups, Presburger Formulas and Languages, *Pacific Journ. Math.*, Vol. 16, pp. 285-296 (1966).
- 8) Girault, C. and Reisig, W. (ed.): *Application and Theory of Petri Nets*, Informatik-Fachberichte, Vol. 52, Springer-Verlag (1982).
- 9) Grabowski, J.: The Decidability of Persistence for Vector Addition Systems, *Inform.*

- Process. Lett., Vol. 11, pp. 20-23 (1980).
- 10) Hack, M.: Decision Problems for Petri Nets and Vector Addition Systems, Computation Structures Group Memo 95, Project MAC, MIT (1974).
 - 11) Hack, M.: The Recursive Equivalence of the Reachability Problem and the Liveness Problem for Petri Nets and Vector Addition Systems, Proc. 15th IEEE Symposium on Switching and Automata, pp. 156-164 (1974).
 - 12) Hack, M.: Petri Net Languages, Computation Structure Group Memo 121, Project MAC, MIT (1975).
 - 13) Hack, M.: The Equality Problem for Vector Addition Systems is Undecidable, Theor. Comput. Sci., Vol. 2, No. 1, pp. 77-96 (1976).
 - 14) Hopcroft, J. and Pansiot, J.: On the Reachability Problem for 5-dimensional Vector Addition Systems, Theor. Comp. Sci., Vol. 8, No. 1, pp. 135-159 (1979).
 - 15) Jensen, K.: Colored Petri Nets and the Invariant-method, Theor. Comp. Sci., Vol. 14, No. 3, pp. 317-336 (1981).
 - 16) Karp, R. and Miller, R.: Parallel Program Schemata, J. Comput. Syst. Sci., Vol. 3, No. 4, pp. 167-195 (1969).
 - 17) Kosaraju, S. R.: Decidability of Reachability in Vector Addition Systems, Proc. 14th ACM Symposium on Theory of Computing, pp. 267-281 (1982).
 - 18) Landweber, L. and Robertson, E.: Properties of Conflict-free and Persistent Petri Nets, J. ACM, Vol. 25, No. 3, pp. 352-364 (1978).
 - 19) Martinez, J. and Silver, M.: A Simple and First Algorithm to Obtain All Invariants of a Generalized Petri Net, in Application and Theory of Petri Nets, Informatik-Fachberichte, Vol. 52, Springer-Verlag (1982).
 - 20) Mayr, E. W.: An Algorithm for the General Petri Net Reachability Problem, Proc. 13th ACM Symposium on Theory of Computing, pp. 238-246 (1977).
 - 21) Peterson, J.: Computation Sequence Sets, J. Comput. Syst. Sci., Vol. 13, No. 1, pp. 1-24 (1976).
 - 22) Peterson, J.: Petri Nets, Comput. Surv., Vol. 9, No. 3, pp. 223-252 (1977). 訳: 齊藤信男, bit 臨時増刊, 12月, pp. 99-128 (1978).
 - 23) Peterson, J.: A Note on Colored Petri Nets, Inform. Process. Lett., Vol. 11, No. 1, pp. 40-43 (1980).
 - 24) Peterson, J.: Petri Net Theory and the Modeling of Systems, Prentice-Hall (1981).
 - 25) Petri, C.: Kommunikation mit Automaten, Ph. D. dissertation, University of Bonn, West Germany (1962).
 - 26) Sacerdote, S. and Tenney, R.: The Decidability of the Reachability Problem for Vector Addition Systems, Proc. 9th ACM Symposium on Theory of Computing, pp. 61-76 (1977).
 - 27) Shaw, A. C.: Software Descriptions with Flow Expressions, IEEE Trans. Softw. Eng., Vol. 4, No. 3, pp. 242-254 (1978).
 - 28) Stark, H. P.: Processes in Petri Nets, Fundamentals of Computation Theory, Lecture Notes in Comput. Sci., Vol. 117, Springer-Verlag (1981).
 - 29) Van Leeuwen, J.: A Partial Solution to the Reachability Problem for Vector Addition Systems, Proc. 6th ACM Symposium on Theory of Computing, pp. 303-309 (1974).
 - 30) Yamasaki, H.: On Weak Persistency of Petri Nets, Inform. Process. Lett., Vol. 13, No. 3, pp. 94-97 (1981).

(昭和58年8月22日受付)

