

# ペトリネットによる構造解析に関する研究

○奥田 誠<sup>\*1)</sup>、長尾 達明<sup>\*1)</sup>、水矢 亨<sup>\*1)</sup>、宮澤 以鋼<sup>\*1)</sup>

## 1. はじめに

近年、生産システムの要求事項の増加に伴って制御システムは複雑化しており、システムの挙動や制御構造を把握して不具合を早期発見することが重要である。これまで制御システムに対して、主に挙動解析が行われてきた。筆者らは、実用上要請の強いシステムの制御構造の解析について研究しており、制御システムの典型的な処理に対応した構造を導出するアルゴリズムを開発中である。システムの制御構造が把握できれば、設計仕様と比較することで不具合を減少させ、生産性の向上が期待できる。

## 2. 制御システムのペトリネット (Petri-net) による表現

制御システムは離散事象システムであり、Petri-net による表現に適している。制御システムの典型的な処理として、繰り返し実行と分岐処理が挙げられる。これらの処理は Petri-net のサイクル構造と並列構造に相当する。Petri-net は、place (状態)・transition (遷移条件) とそれらを接続する arc から構成される (図 1)。サイクル構造とはある状態から元の状態に遷移する経路のことで (図 2 左)、並列構造とはある状態から別の状態に遷移する 2 つ以上の経路から成る (図 2 右)。

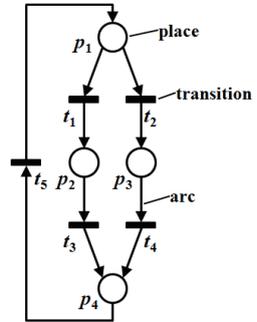


図 1. Petri-net モデルのグラフ表現

## 3. Petri-net による構造解析

Petri-net の構造は、連立方程式  $BT_x=0$  を解くことによって、サイクル構造と並列構造に分類可能である。 $B$  は接続行列といい、行は place、列は transition に対応している。接続行列の各要素の値  $B_{ij}$  は、 $t_j$  から  $p_i$  に入力される arc の本数である。ただし、 $p_i$  から  $t_j$  に出力される場合は負値を取る。例として図 3 に、図 1 の接続行列を示す。 $BT_x=0$  の解  $T_x$  は transition に対応しており、その条件によって Petri-net の構造を以下のように分類できる。

【条件 1】 非零の全成分が同符号であるとき、サイクル構造  
 【条件 2】 非零の成分が異なる符号を持つとき、並列構造  
 ただし、【条件 2】を満たす解には並列構造を表さない場合がある。【条件 2】を満たす解は、成分の負値に対応する transition に接続される arc の向きを反転させるとサイクル構造を示す。このため、負値に対応するすべての transition が place を介して連続であれば並列構造であり、不連続ならば並列構造でないことが分かる。【条件 2】を満たす解が並列構造か否かを判別する方法を提案し、サイクル構造と並列構造を導出するアルゴリズムを開発する。

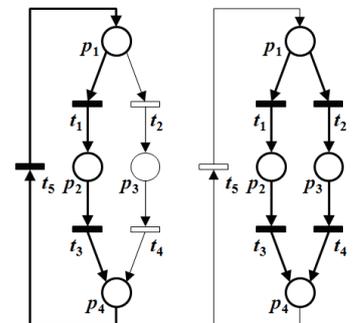


図 2. サイクル構造 (左) と並列構造 (右)

$$B = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 \end{matrix} \\ \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{matrix} & \begin{bmatrix} -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix} \end{matrix}$$

図 3. 図 1. の接続行列

## 4. 解析例

図 1 について  $BT_x=0$  を解くと、 $T_x$  の基本解として、 $(-1, 1, -1, 1, 0) \dots (1)$  と  $(1, 0, 1, 0, 1) \dots (2)$  が得られる。解 (1) は、【条件 2】を満たし、負値である  $t_1, t_3$  は図 1 より連続していることが分かるため、 $t_1, t_3$  と  $t_2, t_4$  の並列構造を表す。解 (2) は、【条件 1】をみたすため、 $t_1, t_3, t_5$  のサイクル構造を表す。これらの導出した構造は、図 2 より明らかである。また、解 (1) に解 (2) を加えると  $(0, 1, 0, 1, 1)$  が得られ、これは【条件 1】をみたすため、 $t_2, t_4, t_5$  のサイクル構造を表す。

## 5. まとめ

制御システムの典型的な処理である繰り返し実行と分岐処理の解析手法について述べ、簡単な解析例を示した。しかし本来は、挙動解析を行う前に、設計仕様と制御構造が一致しているか検証する必要があると考える。本解析手法を用いれば、繰り返し実行と分岐処理に対して解析が可能となり、不具合の早期発見に貢献できると考えている。

\*1)神奈川県産業技術センター