

制御システムのモデル化に適するように拡張した 論理カラーペトリネット*

和崎 克己** 不破 泰*** 江口 正義**** 中村 八束*****

(平成7年10月31日 受理)

Logical Coloured Petri Net expanded to be suitable for making the Control System Model

Katsumi WASAKI, Yasushi FUWA, Masayoshi EGUCHI
and Yatsuka NAKAMURA

In this paper, we propose an extension of Petri nets (LC-net) suitable for control systems design and discuss its methods of evaluation. This LC-net is a marked net with addition of the following features: data assignment of marks, representation of firing conditions as logic equations, coupling of output procedures with transition firing, etc. Also, since the concept of transition firing evaluation orders is omitted from the analysis of conventional Petri nets, we introduce this concept formally in our proposed net. Finally, in order to study the behavior of a system modelled with this net, we provide a means for searching the reachability tree of markings. This LC-net is a extended Petri nets which solve a problem of description from Place/Transition Petri nets (PNs) and Colored Petri nets (CPNs) before.

1 まえがき

並列システムの設計においては、ペトリネットをモデル化手法として採用したものが多く発表されている。これは、ペトリネットが並列システムが持つ特性の記述に適していること、一度ペトリネットモデル化すると解析が容易に行えることによる¹⁾²⁾。なかでも、プレース・トランジションペトリネット(以下、PNsと記す)は、特性が時間に依存しないシステムの設計に際して有効であることが知られている。また、対象とするシステムの複雑度が増してきた場合には、PNsよりハイレベルなペトリネットである、カラーペトリネット³⁾(以下、CPNsと記す)がよく用いられる。

* 平成5年5月 電子情報通信学会 ソフトウェアサイエンス研究会発表予稿に加筆

** 電子制御工学科 助手

*** 信州大学工学部 助教授

**** 東京商船大学商船学部 助教授

***** 信州大学工学部 教授

従来からの PNs および CPNs を，システムの設計と解析に適用した例としては，Jensen によるハードウェア設計への応用³⁾，Murata らによる Ada プログラムのデッドロック検証への適用⁴⁾，Nagao らによる FA 制御への応用⁵⁾，などが挙げられる。

PNs や CPNs を実際に制御システムの記述に適用した時，次の問題点がある．ネットの動作（トランジションの発火条件ならびに発火時のトークンの移動）が一意に固定されている．処理の分岐条件に応じて多くのトランジションやプレースを用いる必要がある．よって，ネットの規模が増大する．

本稿は，論理カラーペトリネット⁶⁾（Logical Coloured Petri net : 以下 LC-net と記す）を提案する．LC-net は，PNs に次の改良を行ったものである．トークンはデータを持つ．発火条件は，入力プレース内のトークンの有無や，トークンのデータ値を用いた任意の論理式で与える．発火時のトークンの出力は入力プレースのトークンのデータ値を用いた関数によって設定する．以上の改良により，条件分岐のトランジションやプレースの追加が不要で，ネット規模を縮小できる⁷⁾．

トランジション発火評価順位の概念を LC-net に導入する．ペトリネットをソフトウェアやハードウェア上へ実装した際には，トランジションの評価順序はそれぞれ，トランジションへ割りあてられた制御手続きの呼びだし順序，または動作用のクロックの配分方法から一意に定まる．しかし，従来の解析は，不確実な順番で起こる入力（刺激）に対する応答の解析時，トランジションの発火評価の順位付けを行わない．著者らは，既に発火評価順位という概念を導入し，順位を考慮したデッドロックおよびトラップ性を有するトランジション集合の検討を行い，重要性を指摘している⁸⁾⁹⁾．

LC-net により，実際の画像処理装置を制御するソフトウェアのモデル化を行う．その上で，発火評価の順位付けを考慮した可達木によるネット解析を行う．更に，システム制御用コンピュータへ作成したネットモデルを実装し，動作を確認する．

本稿では，2. で LC-net の定義と発火評価順位の概念について説明する．3. では，LC-net を実際の制御ソフトウェアの設計に適用する．更に 4. で，可達木によるネット解析と，制御用コンピュータへの実装による評価を行う．

2 論理カラーペトリネット (LC-net)

2-1 LC-net の定義

LC-net : $N_E = (S_E, T_E; F_E, M_E)$ は次の各条件 (i)~(v) を満たす

- (i) $S_E = \{s_1, s_2, \dots, s_n\}$ 及び $T_E = \{t_1, t_2, \dots, t_m\}$ は，それぞれプレース，トランジションの集合であり， $F_E = \{f_1, f_2, \dots, f_l\} \subseteq (S_E \times T_E) \cup (T_E \times S_E)$ はプレースからトランジション，もしくはトランジションからプレースへのアークの集合である．
- (ii) 各プレース $s_i \in S_E$ ($i = 1, 2, \dots, n$; $n = |S_E|$) のマークは，1 から N までの自然数の値をとることができる．これを $\mu(s_i)$ と書く．ただし， s_i にマークがないときは $\mu(s_i) = 0$ とする．

$$\mu : S_E \longrightarrow \{0, 1, 2, \dots, N\}$$

(iii) 各プレース s_i のマークの容量は 1.

(ii),(iii) から, 全てのプレースのあらゆるマーク値からなる集合 M_E は, S_E から $\{0, 1, 2, \dots, N\}$ への写像の作る集合で表す.

$$M_E \equiv \{0, 1, 2, \dots, N\}^{S_E}$$

(iv) $t_j \in T_E$ ($j = 1, 2, \dots, m; m = |T_E|$) へのアークを有する全てのプレースの集合を *t_j , t_j からの全てのアークが指すプレースの集合を t_j^* と書く.

$${}^*t_j = \{s \in S_E : (\exists f)(f \in F_E, f = (s, t_j))\}$$

$$t_j^* = \{s \in S_E : (\exists f)(f \in F_E, f = (t_j, s))\}$$

(v) 任意のマーキング $\mu \in M_E$ の下での t_j に関する発火評価は, *t_j に属するプレースの状態 $\mu|{}^*t_j$ の論理式で記述される発火条件 $\Phi^j(\mu|{}^*t_j)$ を調べる. Φ^j の真偽によって, 次段マーキング $\mu' \in M_E$ が変化する.

Φ^j が真ならば t_j は発火する. このときプレース ${}^*t_j - t_j^*$ はマークを取り除く. プレース t_j^* は, *t_j の状態に依存して次の C^j によって与えられる. つまり,

$$C^j : \{0, 1, 2, \dots, N\}^{t_j} \longrightarrow \{0, 1, 2, \dots, N\}^{t_j^*}$$

if $\Phi^j(\mu|{}^*t_j) = \text{True}$ then

$$\mu' = \begin{cases} 0 & : \text{on } {}^*t_j - t_j^* \\ C^j(\mu|{}^*t_j) & : \text{on } t_j^* \\ \mu & : \text{otherwise} \end{cases}$$

一方, Φ^j が偽ならば t_j は発火しない. このとき $\mu' = \mu$ である.

トランジション t_j の評価後の次段マーキングについては, ネットのプレースのマーク状態 $\mu \in M_E$ から $\mu' \in M_E$ への写像 f^j で表すことが出来る.

2-2 発火評価順序の導入

LC-net : $N_E = (S_E, T_E; F_E, M_E)$ に導入する, トランジションの発火評価順序の定義を以下に示す.

- (i) $G; \{1, 2, \dots, m\}$ ($m = |T_E|$) の置換群. トランジションの発火評価 $\{t_1, t_2, \dots, t_m\}$ の順番を決める. $g \in G$ に対して発火評価の無限系列 $t_{g(1)}, t_{g(2)}, \dots, t_{g(m)}, t_{g(1)}, t_{g(2)}, \dots, t_{g(m)}, \dots$ を対応させる.
- (ii) トランジション t_j ($j = 1, 2, \dots, m : m = |T_E|$) の発火評価は, プレースのマーク状態 $\mu \in M_E$ から $\mu' \in M_E$ への写像 f^j で表す.
- (iii) m 個のトランジションに関する発火評価順序は, $g \in G$ に基づく. このとき,

$$f^{g(1)g(2)\dots g(m)} = f^{g(m)} \times \dots \times f^{g(2)} \times f^{g(1)}$$

(iv) m 個のトランジションに対する 1 サイクルの評価順序 g が与えられたとき,

$$g : t_{g(1)} \rightarrow t_{g(2)} \rightarrow \cdots \rightarrow t_{g(m)}$$

と書く。また、このことを略して次のように書く。

$$g = (g(1), g(2), \cdots, g(m))$$

3 制御ソフトウェアの LC-net によるモデル化

3-1 画像処理装置制御システムの概要

画像処理システムの機器構成を、図 1(a) に示す。この制御システムは、画像処理装置、XY ステージ、ステージコントローラ、CCD カメラ、及びシステム制御用コンピュータから構成されている。

おおまかな機能は以下の通りである。XY ステージ上に置かれたプリント基板 (PCB) などの対象物の表面を、セルと呼ぶ小さな四角形の領域に分割しておく。CCD カメラでこの各セルを原画像として次々に取り込み、取り込みが終了した原画像をあらかじめ用意しておいたいくつかの基準画像と比較することにより、対象物の品質検査を行う。例えば対象物がプリントパターンである場合、正常なパターンを基準画像とし、取りこんだ各セルの画像との比較を行い、その不一致な箇所を調べることでパターンの欠陥が判る。欠陥の程度は不一致な画素数で制御用コンピュータへ通知する。

画像処理装置の内部ハードウェア構成について説明する。構成図を、図 1(b) に示す。この画像処理装置は、CCD カメラから取りこんだ各セルの画像を格納するための A, B 2 つの原画像メモリ (Raw Image Memory)、マッチング処理フィルタ (Matching Filter)、ピクセル数カウンタ (Pixel Counter)、A, B 2 つの基準画像メモリ (Template Image Memory)、および複数枚の基準画像を蓄積しておくための基準画像ストレージ (Template Image Storage) から構成される。原画像メモリと基準画像メモリを 2 つ持つことにより、CCD カメラからある原画像メモリへ取込処理を行っている間に、他方の原画像メモリと基準画像とのマッチング処理を同時に行うことが可能である。また、基準画像ストレージと基準画像メモリとの間は、独立したデータバスによって接続されているので、ストレージからメモリへの転送は独立して動作させることが出来る。これらの並列処理の制御は、システム制御用コンピュータに実装されるソフトウェアが行う。

3-2 LC-net による制御システムの設計例

画像処理シーケンスに基づいた、LC-net による制御システムの設計例を図 2 に示す。

画像処理シーケンスのうち、原画像の取込処理、基準画像の転送処理、およびマッチング処理などの各処理は、それぞれ図中の Subnet-1, 2, 3 で示すサブネットでは表現されている。各サブネットの入口である、プレース s_1, s_4, s_6 へマークを送出することで、それぞれのサブネットの動作が開始される¹。

¹各サブネットの動作や詳細定義は紙面の都合により省略

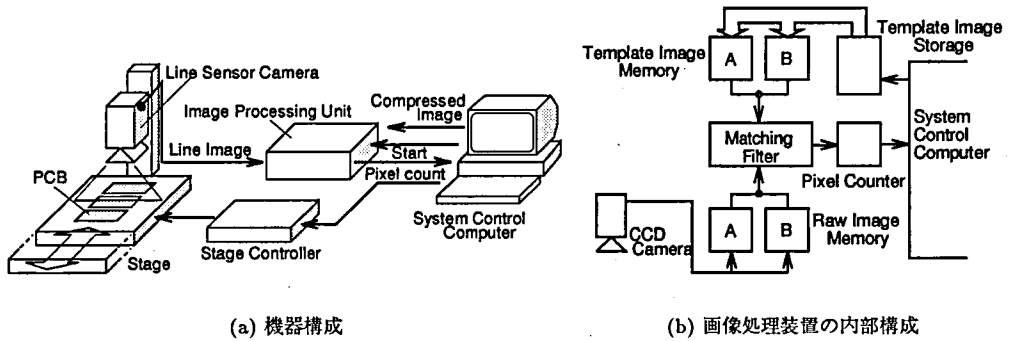


図 1 画像処理システム

Fig 1: An Image Processing System.

Subnet-1~3 の同期化のために、各サブネットの入口と出口にあたるプレースを、同期用トランジション t_1 および t_2 によって連結する。プレースに格納されるトークンの値は $1 \sim N$ で、各サブネットで処理したセル番号を示す。

トランジション t_1 は、原画像の取込中オーバーラップ点に到達したとき、次のセルの原画像取込処理を開始させるために用意する。また、全ての検査が完了した際の終了処理も含む。発火条件は $\Phi_{overlap}^1, \Phi_{exit}^1$ の 2 つを用意し、それぞれ次のセルに対する取り込み処理の開始、および終了処理の役割を持つ。 t_1 の定義は次のように与える。

$$\begin{aligned}
 {}^*t_1 &= \{s_2, s_5, s_7\}, t_1^* = \{s_1, s_8\} \\
 \Phi_{overlap}^1(\mu | {}^*t_1) &\equiv (0 < \mu(s_2) < N) \text{ and } (\mu(s_5) \neq 0) \text{ and } (\mu(s_7) \neq 0) \\
 C_{overlap}^1(\mu | {}^*t_1) &\equiv \begin{cases} \mu(s_2) + 1 & : \text{on } s_1 \\ 0 & : \text{on } s_8 \end{cases} \\
 \Phi_{exit}^1(\mu | {}^*t_1) &\equiv (\mu(s_2) = N) \text{ and } (\mu(s_7) \neq 0) \\
 C_{exit}^1(\mu | {}^*t_1) &\equiv \begin{cases} 0 & : \text{on } s_1 \\ \mu(s_2) & : \text{on } s_8 \end{cases}
 \end{aligned}$$

一方、トランジション t_2 は、原画像の取込処理が終了した後、基準画像転送とマッチング処理を行わせるために用意する。ただし、最後の画像取り込み処理の終了後であれば、基準画像の転送は必要なく、マッチング処理の開始のみ行う。発火条件は $\Phi_{normal}^2, \Phi_{final}^2$ の 2 つを用意し、それぞれ基準画像転送とマッチング処理の開始、およびマッチング処理のみの開始の役割を持つ。 t_2 の定義は次のように与える。

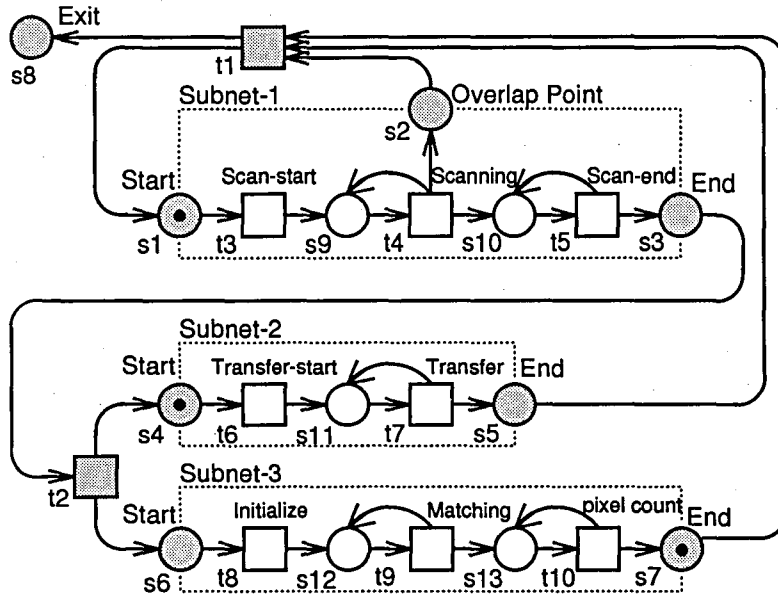


図 2 画像処理のための LC-net 図

Fig 2: LC-net figure for image processing.

$$\begin{aligned}
 {}^*t_2 &= \{s_3\}, t_2^* = \{s_4, s_6\} \\
 \Phi_{normal}^2(\mu | {}^*t_2) &\equiv (0 < \mu(s_3) < N) \\
 C_{normal}^2(\mu | {}^*t_2) &\equiv \begin{cases} \mu(s_3) + 1 & : on\ s_4 \\ \mu(s_3) & : on\ s_6 \end{cases} \\
 \Phi_{final}^2(\mu | {}^*t_2) &\equiv (\mu(s_3) = N) \\
 C_{final}^2(\mu | {}^*t_2) &\equiv \begin{cases} 0 & : on\ s_4 \\ \mu(s_3) & : on\ s_6 \end{cases}
 \end{aligned}$$

本制御システムを実現するために作成したネットの規模は、図 2 に示すように、サブネット 3, トランジション 10, プレース 13 であった。

4 解析と動作試験

4-1 LC-net の解析

図 2 の LC-net モデルを解析する。本システムは対象となるプリント基板等が搬送される毎に、処理を繰り返し安定して行う必要がある。このためには、システム中に発火できないトランジションがないこと、ある一定の繰り返し処理を行う際に、繰り返し現れるマーキングの状態

が常に保たれていることが必須である。そのため、設定した初期マーキングに基づいた可達木を生成し、ネット解析を行う。

なお、以後の説明では、同期用および終了用トランジションに連結されている、各サブネットの入出力部分のプレース ($s_1 \sim s_8$) に関するマーキング状態 (μ) のみを記述する。

$$\mu = (s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8)$$

4-1-1 初期マーキングの設定

ネット解析を行うために、図2に示したLC-net図に対して、初期マーキングの設定を行う。初期マーキング μ_0 は、検査を開始する最初のセル番号が1であることから、 $\mu_0 = (1, 0, 0, 1, 0, 0, 1, 0)$ と定義する。この初期マーキングのもとでは、最初の1ステップ目で原画像の取り込み処理と、基準画像の転送処理が開始され、比較処理は行われない。

4-1-2 可達木によるネット解析

初期マーキング μ_0 の下で、図2から生成した可達木の一部を図3に示す。なお、可達木中の各マーキング状態 M_i ($i \in \{1, 2, \dots, 17\}$) の要素 $|s_i|$ は、各プレース s_i に収容されるトークンの数 (0 または 1) で表した。

$$M_i = (|s_1|, |s_2|, |s_3|, |s_4|, |s_5|, |s_6|, |s_7|, |s_8|)$$

まず、マーキング状態の全てのパスを調べると図2の全てのトランジションが発火することがわかる。また、状態 $M_2 = (1, 0, 0, 0, 1, 0, 1, 0)$ に示されるマーキング状態で、各サイクル終了後の状態が常に保たれていることもわかる。よって、この設計例は上述の初期マーキングのもとで、並列動作の安定性が保証されている。

次に状態 M_1, M_7, M_{10} について検討する。これらは、原画像取り込み処理において、次のセルのオーバーラップ点に到達しているにも関わらず、基準画像の転送処理または画素数カウンタ処理が終了していない状態である。つまり、原画像取り込みを行っている時間内に、基準画像の転送処理、マッチング処理および画素数カウンタ処理のいずれかが、間に合っていないということであり、システムとして異常である。この状態を検出するため、 t_1 を変更し、新たなプレース s_{14} を追加する。異常時は、 s_2 内にトークンがあつて、 s_5 または s_7 内にトークンが無い状態なので、 t_1 の異常時の発火条件と次段マーキングを次のように追加すればよい。

$$\Phi_{error}^1(\mu |^* t_1) \equiv (\mu(s_2) \neq 0) \text{ and } ((\mu(s_5) = 0) \text{ or } (\mu(s_7) = 0))$$

$$C_{error}^1(\mu |^* t_1) \equiv \begin{cases} 0 & : \text{on } s_{11}, s_8 \\ \mu(s_2) & : \text{on } s_{14} \end{cases}$$

以上の修正を行ったネットに対して可達木を生成したところ、上述の異常が起きた場合においても、異常状態を検出して処理が停止することを確かめた。

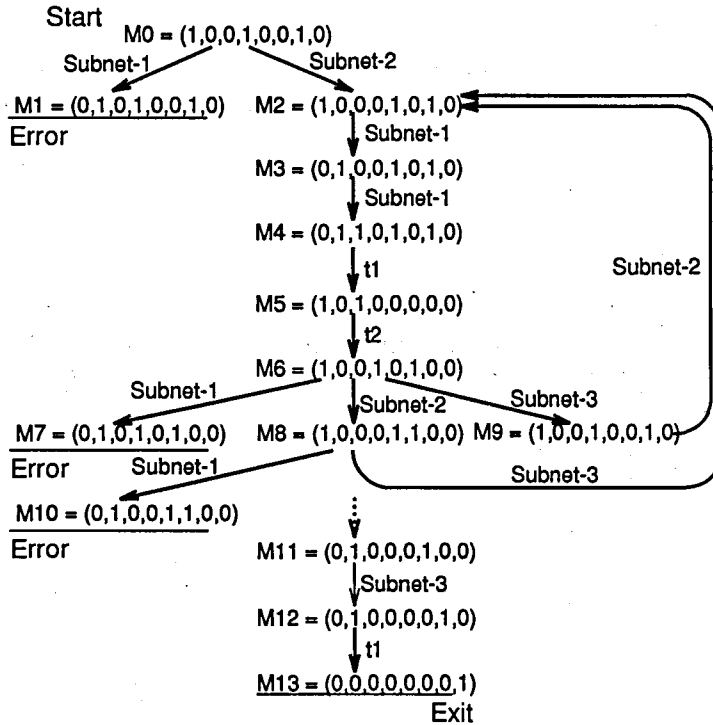


図3 LC-net 図から生成した可達木 (一部分)

Fig 3: Reachability Tree by generate from LC-net figure (Subset).

4-2 制御用コンピュータへの実装と動作試験

LC-net モデルによる並列処理システムを、制御用コンピュータ上へ実装し、動作の確認を行った。

本装置の機能の中で、画像処理シーケンスの部分はペトリネットで記述したが、数値演算処理を伴う比較処理や、高速な処理が必要な XY ステージコントローラとの通信機能、および画面表示等のマンマシンインターフェース機能などの基本プロシージャは、従来の C 言語で実現した。開発時から実地試験時まで、制御用コンピュータには市販のパーソナルコンピュータを用い、LC-net によるモデル化、ネットの検証、基本プロシージャとの結合、および現場の実行マシンでのテスト運用までを行い、正しく動作することを実証した。

LC-net の利用により、この画像処理システムのソフトウェア開発は、特に従来の開発方式では十分な検証なしに実装していた並列処理部分の作成に関して、大幅なコスト削減が図られた。これは、設計時にシステム仕様を LC-net で簡素に記述できたこと、システムの安定性をネット解析によって十分に検証できたこと、基本ワークプロシージャとの結合で制御プログラムを

生成できたこと、更に現場のテスト段階で、制御プログラムのリアルタイムな動作をネットの形で視覚的に確認しながら行えたことの結果である。

5 まとめ

既存 PNs や CPNs の記述性を改善する、論理カラーベトリネット (LC-net) を提案し定義を行った。トランジション発火評価順位の概念を LC-net に導入し、定義を行った。LC-net を用いて、画像処理システムを制御するソフトウェアをモデル化し、可達木によるネット解析を行った。安定性検証済みのネットから生成した制御ソフトウェアを、制御用コンピュータ上へ実装し、その動作を確認した。

今後は、LC-net の動作に関する考察を深めると共に、制御システムを対象とした CASE ツールへの応用を行いながら LC-net の改良を行っていく予定である。

参考文献

- 1) Murata, T. : "Petri Nets: Properties, Analysis and Applications", Proc. IEEE, Vol.77, No.4, pp.541-580, 1989.
- 2) Peterson, J.L. : "Petri Net theory and the Modelling of Systems", Prentice-Hall, Inc., 1981.
- 3) Jensen, K. : "Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use", Vol.1, Basic Concepts, Springer-Verlag, 1992.
- 4) Murata, T., Shenker, B. and Shatz, S.M. : "Detection of Ada Static Deadlocks Using Petri Net Invariants" IEEE Trans. Softw. Eng., Vol.15, No.3, pp.314-326, 1989.
- 5) Nagao, Y., Ohta, H., Urabe, H. and Kumagai, S. : "Petri Net Based Programming System for FMS" IEICE Trans. Fundamentals., Vol.E75-A, No.10, pp.1326-1334, 1992.
- 6) 和崎 克己, 不破 泰, 江口 正義, 中村 八束 : "制御ソフトウェアに適したベトリネットの拡張", 信学技報, COMP93-12, SS93-6, pp.37-44, 1993.
- 7) 和崎 克己, 不破 泰, 江口 正義, 中村 八束 : "CASE ツールとしての論理カラーベトリネット (LC-net) の能力", 信学技報, CAS93-69, pp.101-108, 1993.
- 8) Kawamoto, P., Fuwa, Y. and Nakamura, Y. : "Basic Concepts for Petri Nets with Boolean Markings", Journal of Formalized Mathematics, Vol.4, No.1, 1993.
- 9) 川本 奈緒美, 江口 正義, 不破 泰, 中村 八束 : "The Detection of Deadlocks in Petri Nets with Ordered Evaluation Sequences", 信学技報, SS92-29, KBSE92-50, pp.45-52, 1993.