

NAIST-IS-MT9451087

## 修士論文

ペトリネットを使ったシーケンス制御系の解析

西森 康博

1996年2月16日

奈良先端科学技術大学院大学  
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に  
修士(工学) 授与の要件として提出した修士論文である。

西森 康博

指導教官： 西谷 紘一 教授  
福嶋 雅夫 教授  
藤原 健史 助教授

# ペトリネットを使ったシーケンス制御系の解析\*

西森 康博

## 内容梗概

シーケンス制御は古くからある技術であり、生産現場においてはフィードバック制御と並んでよく用いられる自動制御である。現在、シーケンス制御を実装するプログラマブルコントローラ (PC) のプログラミングはラダー図にもとづいている。しかしながら、このラダー図は論理を表し、制御対象の動作順序を直接記述することができず、プログラムの設計や保全でのプログラムの解読に困難が指摘されている。このため、解読の容易な新しいプログラミングテクニックが要求され、この一つとして SFC (Sequential Function Chart) と呼ばれる記述法を用いたプログラミングテクニックが近年着目されている。SFC は制御対象の動作順や演算処理の順序を流れ図として表現するとともに、各処理における論理についてはラダー図プログラムで記述するため、順序と論理がよくわかる記述法となっている。

SFC が現在かかえている問題点の一つに、SFC を用いて作成したプログラムが正常に動作するか否かを検証するための手法が確立されていないことが挙げられる。本研究では、離散シミュレーションを行なうための一つの手法であるペトリネット (Petri Net) を利用してプログラムの検証を行うため、SFC を用いて作成したプログラムをペトリネットに自動変換する方法を開発した。更に、自動ポンプ運転システムと自動サイロ運転システムを、考案した変換手法によって、ペトリネットに表し、システムの解析およびシミュレーションを行った。

## キーワード

シーケンス制御, SFC, ペトリネット, シミュレーション

---

\*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9451087, 1996年2月16日.

# Analysis of Sequential Control System Using Petri Net\*

Yasuhiro Nishimori

## Abstract

At present, the most popular programming language for the programmable controller(PC) of sequential control system is based on the ladder diagram. However, it is pointed out that logic description of sequence of actions in the ladder diagram presents difficulties to program design and maintenance. A new programming technique has been desired for this purpose.

Recently, a graphical programming technique called SFC(sequential function chart) has been proposed as a solution. It can provide good grasp of the sequence. However, the verification method of the SFC program has not been established. In this paper, a new method to convert a SFC program to the corresponding Petri Net has been proposed for program verification of sequential control by simulation.

Some illustrative examples showed the usefulness of this method.

## Keywords:

Sequential Control, SFC, Petri Net, Simulation

---

\*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9451087, February 16, 1996.

# 目次

<b>1. 序論</b>	<b>1</b>
1.1 はじめに . . . . .	1
1.2 研究の目的と意義 . . . . .	2
<b>2. シーケンス制御系の設計法</b>	<b>3</b>
2.1 SFC 設計 . . . . .	3
2.2 SFC の要素の記述法 . . . . .	6
2.3 シーケンスの規則 . . . . .	9
<b>3. ペトリネット</b>	<b>12</b>
3.1 正規ペトリネットの定義 . . . . .	12
3.1.1 ペトリネットとその特徴 . . . . .	12
3.1.2 発火規則 . . . . .	13
3.1.3 状態ベクトル、発火ベクトル . . . . .	15
3.1.4 接続行列 . . . . .	16
3.1.5 状態方程式 . . . . .	18
3.2 正規ペトリネットによる解析法 . . . . .	19
3.2.1 発火可能ベクトル . . . . .	19
3.2.2 デッドロック . . . . .	20
3.2.3 発火系列を求める問題 (到達問題) . . . . .	21
3.3 時間ペトリネット . . . . .	23
<b>4. シーケンス制御系のペトリネットによる表現</b>	<b>27</b>
4.1 構成要素からペトリネットの表現 . . . . .	27
4.2 SFC 要素からペトリネットの表現 . . . . .	30
4.3 遅延時間を考慮した時間ペトリネット . . . . .	37
<b>5. シミュレーション</b>	<b>42</b>
5.1 自動ポンプ運転システム . . . . .	42

5.2 自動サイロ運転システム . . . . .	59
6. 手法の評価	77
7. 結論	80
謝辞	82
参考文献	83

## 目 次

1	ポンプのレイアウト	3
2	概念設計	4
3	機能レベルの SFC	5
4	ポンプの詳細レイアウト	6
5	詳細レベルの SFC	6
6	ステップの要素	7
7	トランジション条件	8
8	アクションブロック	8
9	アクションとステップの連結	9
10	直列シーケンス	10
11	選択シーケンス	10
12	並行シーケンス	11
13	ペトリネットグラフの例	13
14	$t_3$ 発火後のペトリネットグラフ	14
15	さらに $t_4$ 発火後のペトリネットグラフ	15
16	デッドロックの例	20
17	可達木の例	23
18	時間ペトリネットグラフの例	24
19	$t_1$ 発火後の時間ペトリネットグラフ	25
20	$t_4$ 発火後の時間ペトリネットグラフ	25
21	さらに $t_4$ 発火後の時間ペトリネットグラフ	26
22	$d_4+d_5$ 時刻経過後の時間ペトリネットグラフ	26
23	構成要素のペトリネット表現	28
24	計量タンクシステム	29
25	相互関係	29
26	計量タンクのレイアウト	38
27	ペトリネットグラフ	38
28	ペトリネットグラフ	39

29	$t_2$ 発火後のグラフ	39
30	$t_4$ 発火後のグラフ (従来の TPN)	39
31	$t_4$ 発火後のグラフ (特別 TPN)	39
32	タイムチャート (従来の TPN)	40
33	タイムチャート (特別 TPN)	41
34	詳細レベルの SFC	42
35	変換規則によるペトリネットグラフ	44
36	変換規則によるペトリネットグラフ～番号付け～	45
37	スキャン	47
38	ポンプの可達木	48
39	詳細レベルの SFC	50
40	訂正されたペトリネットグラフ	51
41	変換規則によるペトリネットグラフ～番号付け～	52
42	ポンプの可達木	53
43	詳細レベルの SFC	54
44	訂正されたペトリネットグラフ	55
45	変換規則によるペトリネットグラフ～番号付け～	56
46	ポンプの可達木	57
47	サイロのレイアウト	59
48	概念設計	60
49	機能レベルの SFC	61
50	サイロの詳細レイアウト	63
51	詳細レベルの SFC	64
52	詳細レベルの SFC	65
53	変換規則によるペトリネットグラフ	66
54	変換規則によるペトリネットグラフ～番号付け～	67
55	サイロの可達木	70
56	詳細レベルの SFC	72
57	訂正されたペトリネットグラフ	73



58	変換規則によるペトリネットグラフ～番号付け～ . . . . .	74
59	サイロの可達木 . . . . .	75

## 表 目 次

1	直列シーケンス . . . . .	30
2	誤ったアクションブロックの扱い (非保持) . . . . .	31
3	アクションの優先順位 . . . . .	32
4	選択シーケンス、並行シーケンス . . . . .	33
5	その他 . . . . .	34
6	ダミープレース . . . . .	35
7	誤ったアクションブロックの扱い (人間の介在) . . . . .	36
8	誤ったトランジションの扱い (セット/リセット) . . . . .	37
9	自動ポンプ運転システムの遅延時間 . . . . .	46
10	自動ポンプ運転システムの検証項目 . . . . .	49
11	自動ポンプ運転システムの検証結果 . . . . .	49
12	自動ポンプ運転システムの検証結果 . . . . .	54
13	自動ポンプ運転システムの検証結果 . . . . .	58
14	自動サイロ運転システムの遅延時間 . . . . .	69
15	自動サイロ運転システムの検証項目 . . . . .	71
16	自動サイロ運転システムの検証結果 . . . . .	71
17	自動サイロ運転システムの検証結果 . . . . .	76
18	SFC および PN の動作 (1) . . . . .	78
19	SFC および PN の動作 (2) . . . . .	79

# 1. 序論

## 1.1 はじめに

シーケンス制御は古くからある技術であり、生産現場においてはフィードバック制御と並んでよく用いられる技術である。

シーケンス制御とは、あらかじめ定められた順序に従って、制御の各段階を逐次進めていく制御である。現在、シーケンス制御においてはラダー図言語 (LD) と呼ばれるリレー回路向きプログラミング言語が主流である。

ラダー図言語は、有接点リレーの時代から現在の PC の時代まで使われており、信号の組合せの記述としては優れた表現方式である。このラダー図言語は、これまでも種々の改善がなされてきてはいるが、いまなお、制御の順序を表現することが困難であり、このことがラダー図言語の問題点、あるいは限界と指摘され始めている。

そのような背景から近年、ラダー図に変わって、SFC が普及し始めている。『順序』と『論理組合せ』の両方が記述できることが SFC の最大のメリットといえる。IEC 規格 (IEC 1131-3) による SFC の制定により、今後、この規格に準拠した PC が増加することが予想される。

一方、システムの構造が大規模かつ複雑になるいま、仕様、設計の信頼性を確かめるうえでシミュレーションの果たす役割は大きいと考えられる。有限な共有資源とその配分が与えられたとき、システムがどの状態にも遷移できないような状態 (デッドロック) に陥るようなことが起こらないだろうか、与えられた初期状態から出発して望ましい状態に到達できるであろうか、という問題を、シミュレーションによって洗い出して設計を見直すというプロセスが必要となってくる。具体的にいうと、原子力プラントで重大事故に至るようなすべての故障パス検出と、復旧対策戦略としてどのようなことが可能であるかなど。非決定的な並行動作を許す離散事象システムでは、このようなことが難解な問題となる。

## 1.2 研究の目的と意義

本研究では、シミュレーションを行うモデルとしてペトリネット (Petri Net) を使う。ペトリネットは、C.Petri の学位論文 (1962) で導入された、並列・同時進行性の複数のプロセスからなる離散事象システムを表現するモデルの1つである。ペトリネットは他のモデルと比較して、視覚的、数学的、シミュレーションに優れているという特徴を持っている。

最近になって、プロセスプラントに SFC を含んだ制御系の解析として、論文 [5] などが発表されている。しかし、この論文の解析法は、実際のシステムをシミュレーションするには不十分である。その理由として SFC におけるアクション、タイマ、無条件発火に対し、十分な考察がなされていないことが挙げられる。

本研究では、ペトリネットを拡張した時間ペトリネット (Timed Petri Net) を使う。また、システムの挙動が実際の物理現象を反映する為に、遅延要素を加える。が、従来の時間ペトリネットでは遅延要素を十分に表現することができない。従って、時間ペトリネットに新しく遅延のための規則を追加し、シミュレーションを行なう。

本論文では、第2章ではシーケンス制御系の基本的な設計法について述べ、第3章でペトリネットについて説明する。第4章でシステムの構成要素ならびに SFC からペトリネットに変換する方法を述べ、第5章で自動ポンプ運転システムと自動サイロ運転システムを考案した手法によってペトリネットに変換し、シミュレーションを行う。また、第6章で評価を、第7章で結論を述べる。

## 2. シーケンス制御系の設計法

ここでは、近年、着目されている SFC による設計法を述べる。次に SFC 要素の記述法を示した後、SFC 要素を並べたシーケンスの規則について述べる。

### 2.1 SFC 設計

SFC の設計は大きく分けて 3 つの段階を経ることに設計される。

1. 概念設計
2. 機能設計
3. 詳細設計

これを図 1 に示す簡単なプロセスを例にとって SFC の設計を行う。

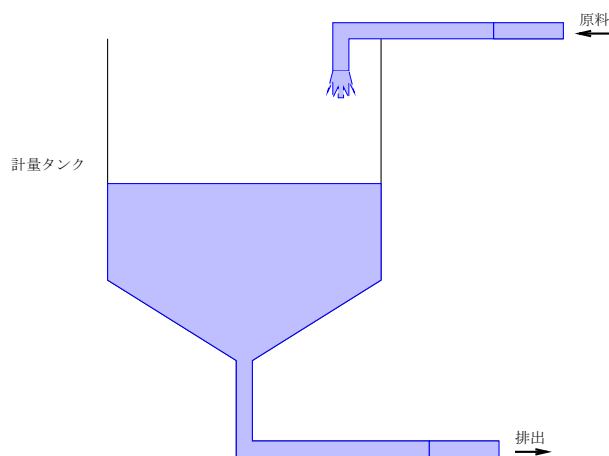


図 1 ポンプのレイアウト

#### 1. 概念設計レベル

このレベルでは、システムの目標を把握する。計量タンクシステムの目標は、

『操作者の指示に従い一定量の原料を計量する』  
である。

この目標から、大きく何をすべきかを機能系統図に示す(図 2(a))。得られた機能系統図から機械的にマクロレベルの SFC (図 2(b)) が作成される。

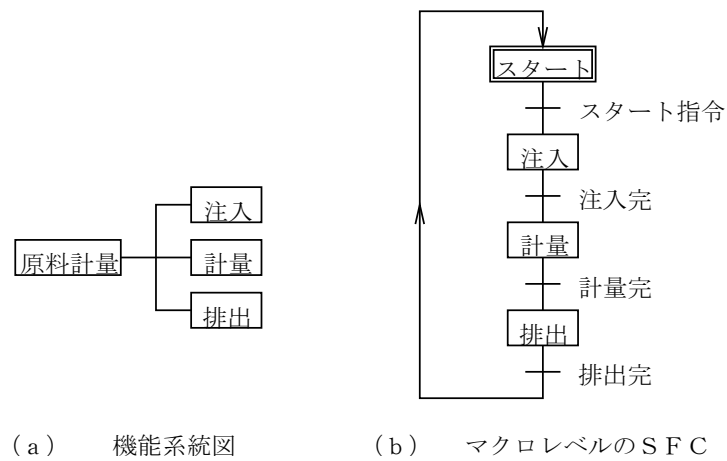


図 2 概念設計

## 2. 機能設計レベル

このレベルでは、制御対象に対する制御装置の動作を記述する。

前のレベルで作成されたマクロレベルの SFC に順序関係を与えると、機能レベルの SFC(図 3) が得られる。

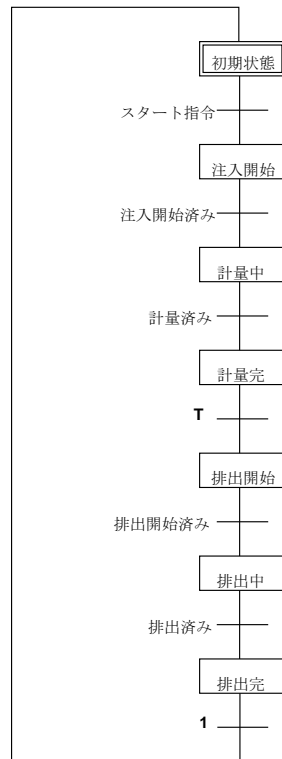


図 3 機能レベルの SFC

### 3. 詳細設計レベル

このレベルではまず、機器・設備の把握を行う(下記参照)。

- 制御対象            --- 計量タンク (液位により 3 値表現),  
                              --- 流入管・排出管 (2 値表現)
- 制御操作用機器    --- 電磁バルブ (2 値表現)
- 監視用機器         --- 表示ランプ (2 値表現)
- 検出用機器         --- レベルスイッチ (2 値表現),  
                              --- タイマ (2 値表現)
- 制御指令用機器    --- 押しボタンスイッチ (2 値表現)

制御対象の構成要素の各機能を機器と対応させながら展開することにより、詳細レベルのレイアウトを作成する (図 4)。

作成されたレイアウトと機能レベルの SFC より、詳細レベルの SFC(図 5)を作成する。

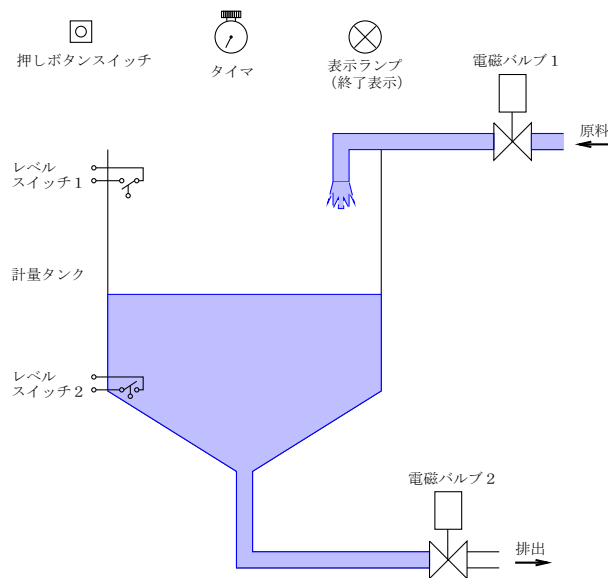


図 4 ポンプの詳細レイアウト

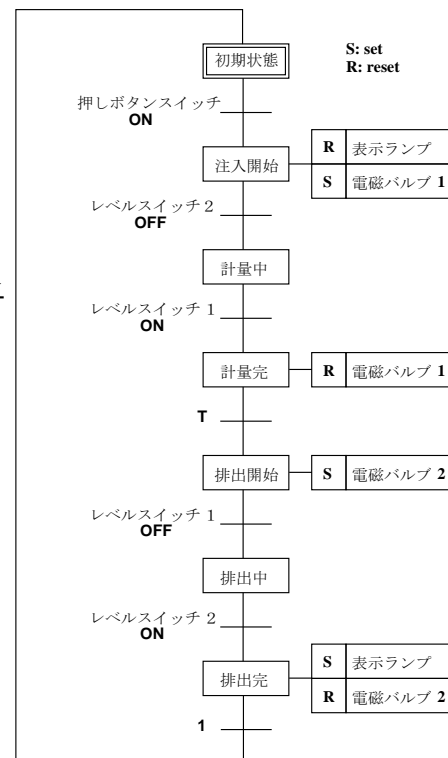


図 5 詳細レベルの SFC

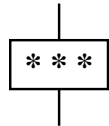
## 2.2 SFC の要素の記述法

SFC は、IEC 規格ではラダー図や IL のようなシーケンスを表記する独立した言語ではなく、順序制御における実行順序や処理内容を表現する規則として定義される。従って、グラフィック形の言語でもテキスト形の言語でも使用でき、多くの言語で使われることが期待される。

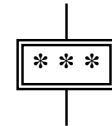


以下に、IEC 規格による SFC の構成要素（ステップ、トランジション条件、アクション、リンク）の記述方法を示す。

## 1. ステップ (図 6)



(a) ステップ



(b) 初期ステップ

図 6 ステップの要素

### ステップ

- 上下二つのリンクを持つ
- 『\*\*\*』はステップ名

### 初期ステップ

- 上下二つのリンクを持つ
- 先行のステップがない場合、上側のリンクは不要
- 『\*\*\*』は初期ステップ名

## 2. トランジション条件

トランジション条件は、先行するステップから後続のステップへリンクに沿って遷移する条件を表している。トランジションは水平の線で、リンクと直角に交差させて表示しなければならない (図 7)。トランジション条件は、あるステップから次のステップへ移行することを示している。

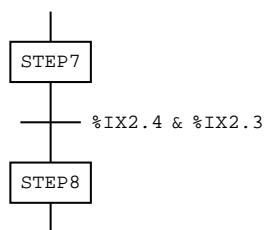


図7 トランジション条件

図7に表現されてあるトランジション条件は、検出信号 2.3 と 2.4 の両方も条件が満たされれば、次のステップへ移行することを示している。

### 3. アクションブロック

アクションブロックは、そのステップが活性の時に処理する内容を記述したもので、信号の論理組合せを図で示したものである (図8)。

アクションブロックのないステップは、待機 (WAIT) ファンクション、すなわち後続のトランジション条件が成立するまで”待つ”ステップとなる。

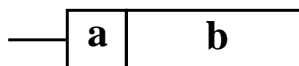


図8 アクションブロック

ただし、

- a : クオリファイア
  - 制御指令の内容を記述。以下にその内容を記述する。
    - S(セット) : 機器の状態を次の状態に進める
    - R(リセット) : 機器の状態を元に戻す
    - N(非保持) : 機器の状態を次の状態にするが、トランジションの条件成立後、元の状態に戻す。

- b : アクション名
  - 指令を与える機器名を記述

#### 4. リンク

リンクはステップ間を接続し、遷移（移行）の道筋を示す（図7）。方向は原則として上から下である。しかし、ループする場合や矢印によって示される場合は、下から上もある。

#### 5. アクションとステップの連結

アクションブロックは、必ずステップの右側に記述する（図9）。

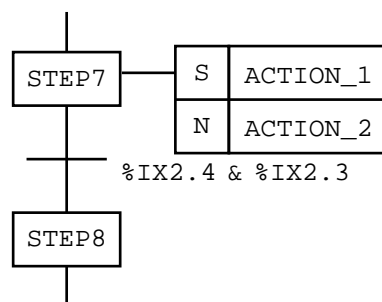


図9 アクションとステップの連結

## 2.3 シーケンスの規則

シーケンスの規則は、直列シーケンス、選択シーケンス、そして並行シーケンスに分けることができる。ここではシーケンスの動作について述べる。

### 1. 直列シーケンス (図10)

- 直列移行
  - シーケンスが直列に進行する直列移行を示す。

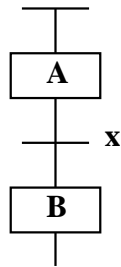


図 10 直列シーケンス

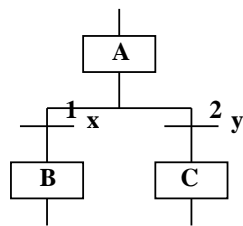
## 2. 選択シーケンス (図 11)

- 選択分岐

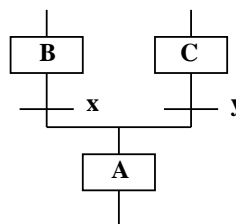
- 水平線下に複数のトランジションが接続されることで、シーケンスの始まりを表現
- トランジションの数字 1、2 はトランジション条件が同時に成立した時の優先度。数字が無いときは左側を優先

- 選択結合

- 複数のトランジションが下の水平線に接続されることで、シーケンス選択の終りを表現



(a) 選択分岐

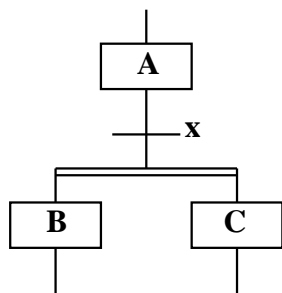


(b) 選択結合

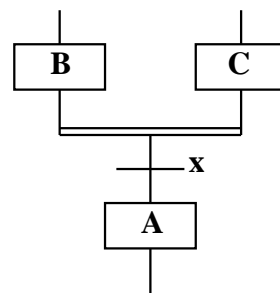
図 11 選択シーケンス

### 3. 並行シーケンス (図 12)

- 並行分岐
  - 二重水平線上の唯一のトランジションにより、並行シーケンスの最初のステップが同期して始まる
  - 以下のシーケンスは、独立して動く
- 並行結合
  - 二重水平線上の並行シーケンスの最後のステップがすべて活性化するのを待ち合わせる。
  - 二重水平線下の唯一のトランジションの条件が成立して並行シーケンスが終る



(a) 並列分岐



(b) 並列結合

図 12 並行シーケンス

## 3. ペトリネット

この章では、並列・同時進行性の複数のプロセスからなる離散事象システムを表現するモデルの1つであるペトリネットについて述べる。

ペトリネットのグラフの2種類のノード、プレースとトランジションに何を対応させるかはモデル作成者の考えによるものであり、プレースとトランジションに対する規定は特にはない。したがって、ペトリネットはプレースとトランジションへの対応づけの仕方により、種々のものが記述できる抽象度の高いグラフである。

### 3.1 正規ペトリネットの定義

#### 3.1.1 ペトリネットとその特徴

ペトリネット (Petri Net) のグラフ表現は、プレースおよびトランジションの2種類のノード(節)とアーク(有効枝)によって表現される。また、ペトリネットではシステムの遷移はトークンの移動によって表す。プレース、トランジション、アークおよびトークンは下記の記号を使用する。

○	:プレース (place)
	:トランジション (transition)
→ または ←	:アーク (arc)
•	:トークン (token)

離散事象システムをペトリネットを用いて表現する時、

1. 遷移の条件を示すトランジション
2. システムの状態を示すプレース
3. 状態遷移の方向を示すアーク
4. システムの状態、個数を示すトークン

と、対応づけることによってシステムを記述できる。

ペトリネットグラフでは、アークはプレースとトランジションを結合するものであり、始点がプレースの時は終点がトランジションであり、逆も同様である。また、同一のプレースとトランジションの組に対して、複数のアークを対応させることが許されている。

図 13 にペトリネットグラフの例を示す。同図において、 $p_1 \sim p_5$  はプレースを、 $t_1 \sim t_5$  はトランジションをそれぞれ示す。

ペトリネットは図 13 にみるように、下記の 4 項組  $(\mathbf{P}, \mathbf{T}, \mathbf{I}, \mathbf{O})$  で定義づけられる。

- $\mathbf{P}$  : プレース  $p_i$  の有限集合
- $\mathbf{T}$  : トランジション  $t_j$  の有限集合
- $\mathbf{I}$  :  $t_j$  から  $p_i$  へのアークに対応する関係を示す行列
- $\mathbf{O}$  :  $p_i$  から  $t_j$  へのアークに対応する関係を示す行列

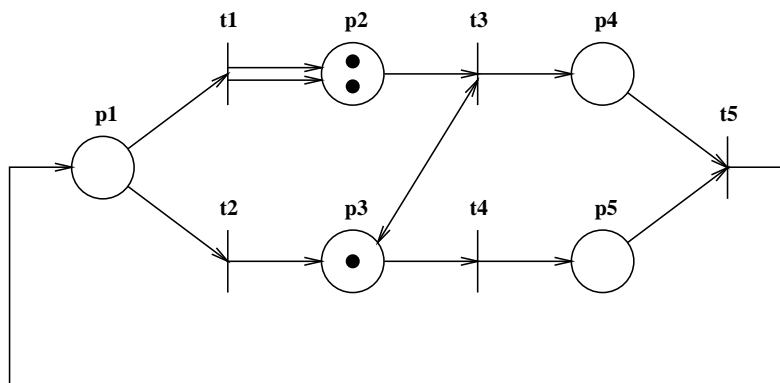


図 13 ペトリネットグラフの例

### 3.1.2 発火規則

システムの状態推移はトークンの移動によって表される。トークンの移動は下記の発火規則に従う。

**規則 1** 全ての入力プレース  $p_i$  から  $t_j$  へのアーク数以上のトークンが、 $p_i$  内に存在する時、 $t_j$  は発火可能である。

**規則 2**  $t_j$ の発火は瞬間的であり、異なる2個の  $t_j, t_l$ が同時に発火する確立は零である。

**規則 3**  $t_j$ の発火に伴い、全ての  $p_i$ から  $t_j$ へのアーク数に等しいトークンを除去する。

**規則 4**  $t_j$ の発火に伴い、出力プレース  $p_k$ に対して  $t_j$ から  $p_k$ へのアーク数に等しいトークンを割り当てる。

図13の例では、 $t_3$ と $t_4$ がともに発火可能状態になる。 $t_3$ 発火後のトークンの分布は図14に示すようになる。

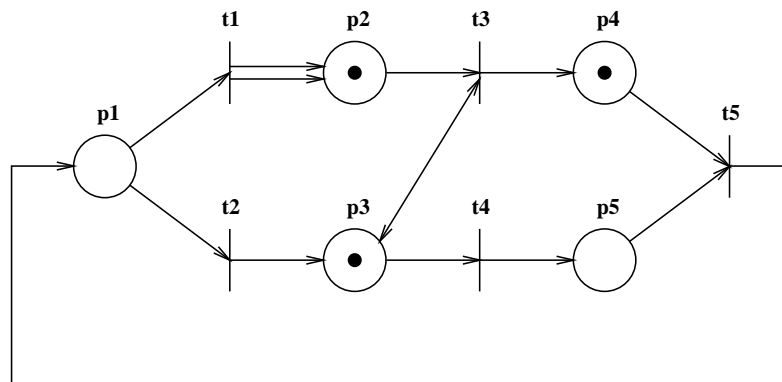


図14  $t_3$ 発火後のペトリネットグラフ

図14においても発火可能なトランジションは $t_3$ と $t_4$ である。ここで $t_4$ が発火すると図15に示す状態となって、 $t_3$ は発火可能でなくなる。このように、一方の発火によりもう一方のトランジションが発火可能でなくなることを実行競合という。

また2つ以上のトランジションが共通の入力プレースを持つことを構造的競合といい、 $\langle p_3, \{t_3, t_4\} \rangle$ のように表す。



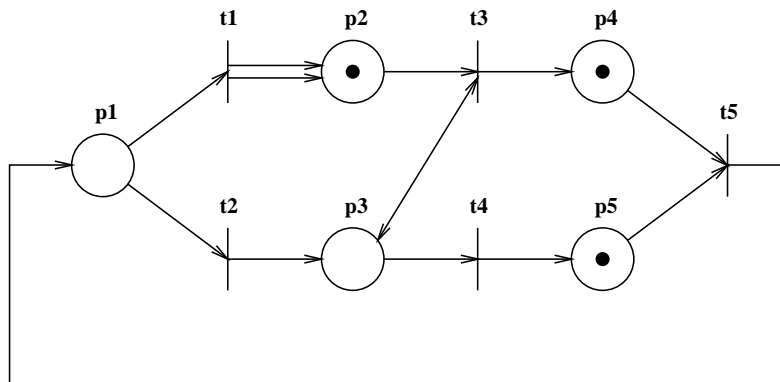


図 15 さらに  $t_4$  発火後のペトリネットグラフ

### 3.1.3 状態ベクトル、発火ベクトル

各プレース内のトークン数を示したものが状態ベクトルである。状態ベクトルはシステムの状態を示す。いまプレース  $p_i$  内のトークン数を  $\hat{p}_i$  で表す。

例 :状態ベクトル

$$\text{状態ベクトル } \hat{\mathbf{p}} = \begin{bmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \\ \hat{p}_5 \end{bmatrix} \quad (1)$$

$$= \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

どのトランジションが発火したかを示すのが発火ベクトルである。発火ベクトルの各要素はトランジションに対応し、値は発火の時、1、非発火の時、0をとる。

トランジション  $t_j$  に対応する変数を  $\hat{t}_j$  で表す。

例 :  $t_3$  が発火した場合、

$$\text{発火ベクトル } \hat{\mathbf{t}} = \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \\ \hat{t}_4 \\ \hat{t}_5 \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (4)$$

### 3.1.4 接続行列

システムの構造はプレースとトランジションおよびそれらの結合関係によって表現される。プレースとトランジションの結合関係は下記の2つの行列によって表現できる。

出力行列  $\mathbf{B}^+$  : トランジションからの出力アーク数の関係を表した行列

入力行列  $\mathbf{B}^-$  : トランジションへの入力アーク数 //

接続行列  $\mathbf{B}$  : トランジションの発火に伴うトークンの出入りを表した行列

- 各トランジションからの出力アーク数を求めると

出力プレース	出力アーク数				
	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
$p_1$	0	0	0	0	1
$p_2$	2	0	0	0	0
$p_3$	0	1	1	0	0
$p_4$	0	0	1	0	0
$p_5$	0	0	0	1	0

となり、トランジションの出力行列  $\mathbf{B}^+$  は次式となる。

$$\text{出力行列 } \mathbf{B}^+ = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5)$$

- 各トランジションからの入力アーク数を求めると

	入力アーク数				
入力プレース	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$
$p_1$	1	1	0	0	0
$p_2$	0	0	1	0	0
$p_3$	0	0	1	1	0
$p_4$	0	0	0	0	1
$p_5$	0	0	0	0	1

となり、トランジションの入力行列  $\mathbf{B}^-$  は次式となる。

$$\text{入力行列 } \mathbf{B}^- = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

- 接続行列  $\mathbf{B}$  は、トランジションの発火に伴うトークンの出入りを表したものであり、式 (7) で求めることができる。

$$\mathbf{B} = \mathbf{B}^+ - \mathbf{B}^- \quad (7)$$

$$= \begin{bmatrix} -1 & -1 & 0 & 0 & 1 \\ 2 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad (8)$$

### 3.1.5 状態方程式

システムの状態はペトリネットグラフのトークンの分布により表現される。現在（時刻  $k$ ）のシステムの状態から、次時刻のシステムを導出するのが状態方程式である。状態方程式はトランジションの発火により、トークンの分布がどのように変化するかを示すものである。

状態方程式は下記のように表すことができる。

$$\mathbf{P}(k+1) = \mathbf{P}(k) + \mathbf{B} \cdot \mathbf{t}(k) \quad (9)$$

ここで、

$\mathbf{P}(k)$  :時刻  $k$  のシステムの状態

$\mathbf{t}(k)$  :時刻  $k$  の発火ベクトル

$\mathbf{P}(k+1)$  :時刻  $k+1$  のシステムの状態

である。

例として、図 13 の状態を初期状態とし、次時刻の状態（図 14）を導出する。  
初期ベクトル  $\mathbf{P}(k)$ (式 2)、発火ベクトル  $\mathbf{t}(k)$ (式 4)、接続行列  $\mathbf{B}$ (式 8))

$$\mathbf{P}(k+1) = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & 0 & 0 & 1 \\ 2 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

$$= \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (11)$$

### 3.2 正規ペトリネットによる解析法

ここでは、ある時刻におけるシステムの状態から次にどの状態に遷移することが可能であるかを知るために必要な発火可能ベクトルについて述べる。次に、システムがどの状態にも遷移できないような状態(デッドロック)の検出、与えられた初期状態から出発して望ましい状態への到達(可達問題)について述べる。

#### 3.2.1 発火可能ベクトル

システムがある条件を満たして状態が次へ遷移することを、ペトリネットでは、ある状態ベクトルで発火可能なトランジションが発火して、次の状態ベクトルに移ったことを示す。

発火可能ベクトルは式(12)で求めることができる。

$$\mathbf{t}(k) = (\mathbf{B}^-)^t \diamond \mathbf{P}(k) \quad (12)$$

但し、

$$b_{ij}^- \diamond p_i(k) = \begin{cases} 1 & b_{ij}^- \leq p_i(k) \\ 0 & b_{ij}^- > p_i(k) \end{cases} \quad (13)$$

図13において発火可能ベクトルを求めると、初期状態ベクトル  $\mathbf{p}(0)=[0\ 2\ 1\ 0\ 0]$ , 入力行列  $\mathbf{B}^-$ (式6)より下記のようになる。

$$\mathbf{t}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \diamond \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (14)$$

$$= \begin{bmatrix} (1 \diamond 0) \cdot (0 \diamond 2) \cdot (0 \diamond 1) \cdot (0 \diamond 0) \cdot (0 \diamond 0) \\ (1 \diamond 0) \cdot (0 \diamond 2) \cdot (0 \diamond 1) \cdot (0 \diamond 0) \cdot (0 \diamond 0) \\ (0 \diamond 0) \cdot (1 \diamond 2) \cdot (1 \diamond 1) \cdot (0 \diamond 0) \cdot (0 \diamond 0) \\ (0 \diamond 0) \cdot (0 \diamond 2) \cdot (1 \diamond 1) \cdot (0 \diamond 0) \cdot (0 \diamond 0) \\ (0 \diamond 0) \cdot (0 \diamond 2) \cdot (0 \diamond 1) \cdot (1 \diamond 0) \cdot (1 \diamond 0) \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (16)$$

これより図 13において発火可能トランジションは、 $t_3$ と $t_4$ である。

### 3.2.2 デッドロック

トークンの推移が不可能となった状態をデッドロック状態と呼ぶ。デッドロック状態の例を図 16に示す。

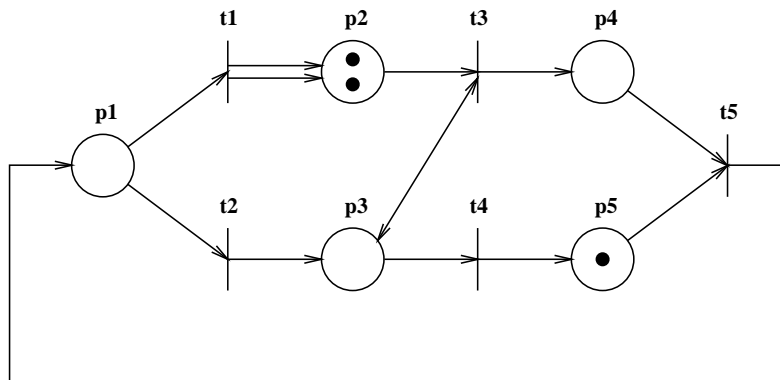


図 16 デッドロックの例

デッドロック状態では、発火可能ベクトル  $\mathbf{t}(\mathbf{k})$  は、

$$\mathbf{t}(\mathbf{k}) = (\mathbf{B}^-)^t \diamond \mathbf{P}(\mathbf{k}) = 0 \quad (17)$$

となる。

$$\begin{cases} t_1 \Rightarrow (1 \diamond \hat{p}_1(k)) \cdot (0 \diamond \hat{p}_2(k)) \cdots \cdot (0 \diamond \hat{p}_5(k)) = 0 \\ t_2 \Rightarrow (1 \diamond \hat{p}_1(k)) \cdot (0 \diamond \hat{p}_2(k)) \cdots \cdot (0 \diamond \hat{p}_5(k)) = 0 \\ t_3 \Rightarrow (0 \diamond \hat{p}_1(k)) \cdot (1 \diamond \hat{p}_2(k)) \cdots \cdot (0 \diamond \hat{p}_5(k)) = 0 \\ t_4 \Rightarrow (0 \diamond \hat{p}_1(k)) \cdot (0 \diamond \hat{p}_2(k)) \cdots \cdot (0 \diamond \hat{p}_5(k)) = 0 \\ t_5 \Rightarrow (0 \diamond \hat{p}_1(k)) \cdot (0 \diamond \hat{p}_2(k)) \cdots \cdot (1 \diamond \hat{p}_5(k)) = 0 \end{cases} \quad (18)$$

$\hat{p}_i(k) = 0, 1, 2, \dots$  であるから

$$(0 \diamond \hat{p}_i(k)) \equiv 1$$

である。従って、上記の条件は下記の条件を同時に満足させることと等価である。

$$\begin{cases} t_1, t_2 \text{ に関する条件} & \hat{p}_1(k) = 0 \\ t_3 \text{ に関する条件} & \hat{p}_2(k) = 0 \text{ または } \hat{p}_3(k) = 0 \\ t_4 \text{ に関する条件} & \hat{p}_3(k) = 0 \\ t_5 \text{ に関する条件} & \hat{p}_4(k) = 0 \text{ または } \hat{p}_5(k) = 0 \end{cases} \quad (19)$$

これより、条件 19 を満足する時、デッドロックの状態になる。

### 3.2.3 発火系列を求める問題 可達問題

システムが初期状態から出発して目的の状態に到達可能か否かを検討し、到達可能な場合にその発火系列を求める問題を**可達問題**という。

可達問題は下記の 2 段階の手順で解くことができる。

1. 初期状態  $P(0)$  から目標状態  $P(N)$  ( $N$  は必ずしも確定していない) への発火回数を求める。

$$\begin{aligned} \mathbf{t}(\mathbf{k}) &= \mathbf{B}^{-1}\{\mathbf{P}(N) - \mathbf{P}(0)\} \\ N &= \sum t_j(k) \end{aligned} \quad (20)$$

- (a)  $t_j(k)$  が零または正の整数値をとらないときは非可達。
  - (b)  $t_j(k)$  が零または正の整数値のとき手順 2 を実行する。
2. 全ての発火可能トランジションを発火させたときの可達木を作る。可達木を求める時の発火回数は、N 回以内とする。
- (a) 可達木に  $P(g)$  が含まれている :  $P(0)$  から  $P(g)$  は可達
  - (b)       "       が含まれていない :  $P(0)$  から  $P(g)$  は非可達

いま例として、図 13 の状態から  $\mathbf{P(N)}=[0\ 1\ 0\ 0\ 1]$  への可達問題を考えると、手順 1 より発火ベクトルの和を求めて、

$$\mathbf{t(k)} = \mathbf{B}^{-1}\{\mathbf{p(N)} - \mathbf{p(0)}\} \quad (21)$$

$$= [0\ 1\ 1\ 2\ 1] \quad (22)$$

$$N = \sum t_j(k) \quad (23)$$

$$= 5 \quad (24)$$

合計 5 回の発火に対する可達木を式 (9) の状態方程式を使って求めると図 17 が得られる。同図より、可達の発火系列が

$$t_3 \rightarrow t_4 \rightarrow t_5 \rightarrow t_2 \rightarrow t_4$$

であることがわかる。



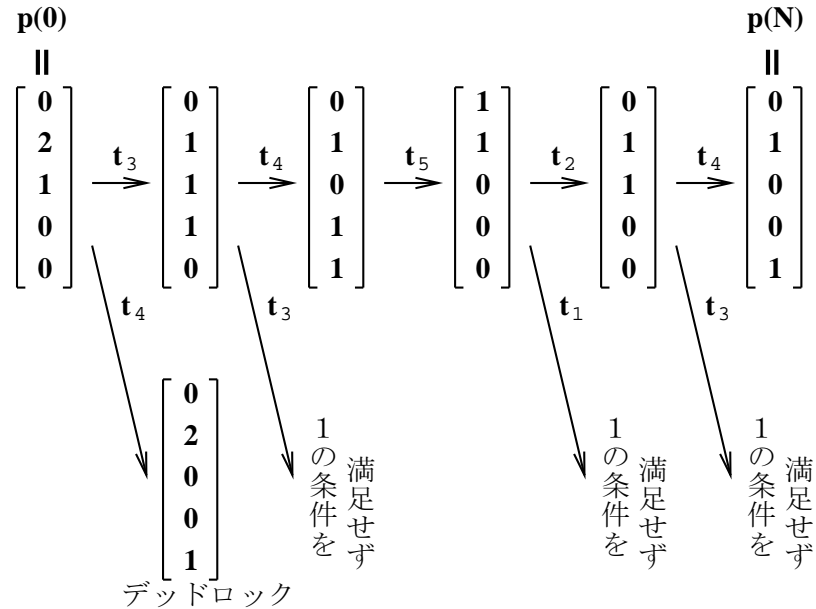


図 17 可達木の例

### 3.3 時間ペトリネット

タイミング、時間の概念は正規ペトリネットの定義にはない。しかし、動的なシステムの性能評価やスケジューリング問題に対して、時間の概念の導入は必要である。ここでは、**遅延時間 (delay time)** を扱うことができる時間ペトリネット (timed PN, TPN) について説明する。時間ペトリネットには、プレースに遅延時間を導入する P 時間 PN と、トランジションに遅延時間を導入する T 時間 PN がある。この 2 つは等価であり、相互変換が可能である。ここでは T 時間 PN について述べる。

#### ●トランジション TPN

各トランジション  $t_j$  に遅延時間  $d_j$  を考える。 $d_j$  は 0 を含む定数の場合を考えるが、一般には変数でもよい。また遅延時間  $d_j$  は、周期的な機能を扱えるよう通常は次のように有理数と定義しているが、実数と定義することもできる。

あるトランジションの発火により、トークンが  $p_i$  に置かれ、 $p_i$  の出力トランジション  $t_j$  の発火条件が満たされると、 $t_j$  は発火可能になる。発火条件を満たしてからある時間後に、発火が実行される。発火条件が満たされた時、この発火に必要なトークンは予約され、予約トークン (reserved token) [8] となる。予約されてから、 $d_j$  の遅延の後、 $t_j$  は実際に発火し、予約トークンは  $t_j$  から取り除かれる。

トランジション  $t_j$  の発火が決定した時刻を  $\tau$  とすると、 $\tau + d_j$  が実際に発火が実行される時刻である。発火は瞬間的に実行される。

T 時間 PN では、これまでの発火規則 1~4 に以下の規則を付け加える。

**規則 5** トランジション  $t_j$  が発火可能となった後、 $b_{ij}$  個のトークンが、トランジション  $t_j$  の発火によりプレース  $p_i$  から取り去られる前に、少なくとも  $d_j$  秒間プレース  $p_i$  に指定保留される。ここで、 $b_{ij}$  は  $p_i$  から  $t_j$  へのアークの重みである。

規則 5 を図 18 を例にとり説明する。遅延条件は、それぞれ  $d_4=2$ ,  $d_5=4$  である。何もついてないものについては、 $d_j=0$  とする。

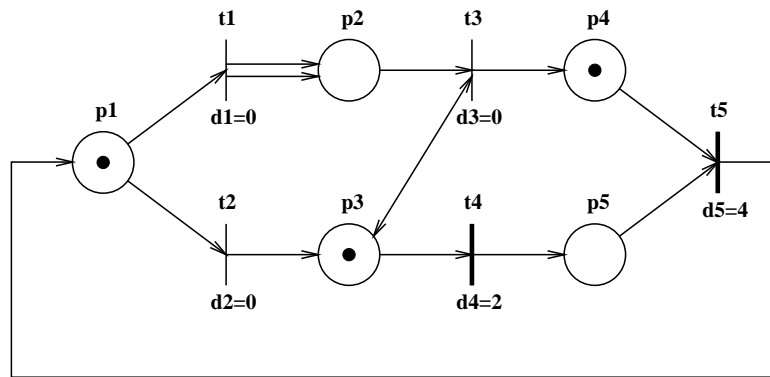


図 18 時間ペトリネットグラフの例

図 18 においては、発火可能トランジションは  $t_1$ ,  $t_2$  と  $t_4$  である。いま  $p_3$  内のトークンは  $t_4$  によって予約されたとする。しかし、 $t_4$  は 2 ステップ遅延が生じるため、この時点での発火可能トランジションは  $t_1$  と  $t_2$  である。

いま、トランジション  $t_1$  が発火すると仮定すると、図 19 のようになる。

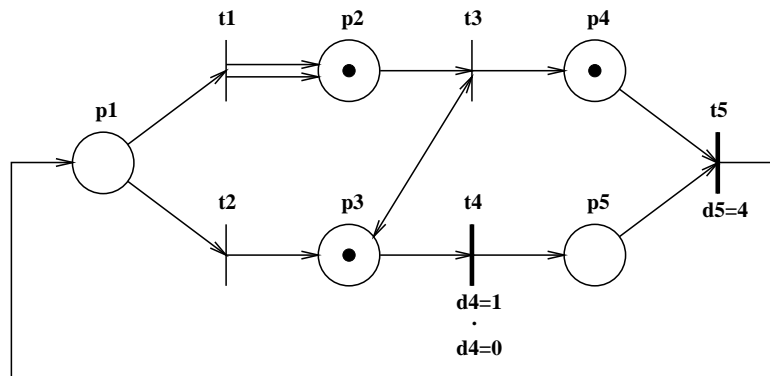


図 19  $t_1$ 発火後の時間ペトリネットグラフ

図 19においては、発火可能トランジションは  $t_3$  と  $t_4$  のように見える。しかし、 $p_3$ 内のトークンは  $t_4$ によって予約トークンとなっている。よって、 $t_3$ は発火可能ではない。 $d_4$ 時刻経過後、遅延時間が 0 になると、 $t_4$ は発火可能となる。

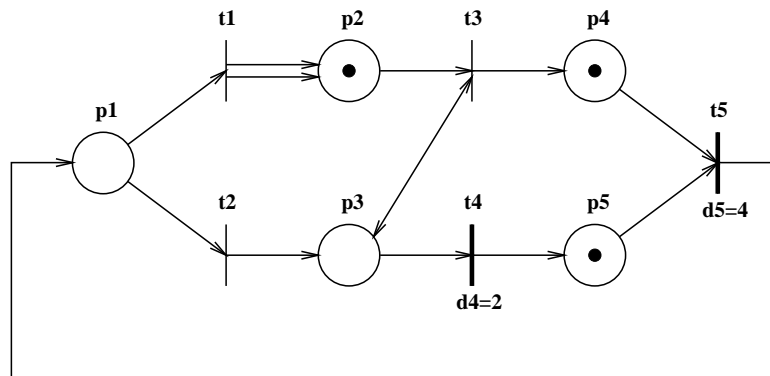


図 20  $t_4$ 発火後の時間ペトリネットグラフ

図 19において  $t_4$ を発火させると図 20になる。

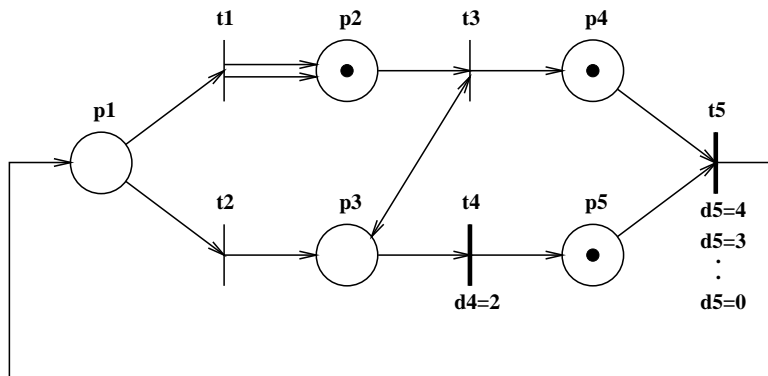


図 21 さらに  $t_4$  発火後の時間ペトリネットグラフ

図 20において規則 1~規則 4では  $t_5$  が発火条件を満たしているが、規則 5により  $d_5$  時刻経過していないので発火可能ではない。 $d_5$  時刻経過後、 $t_5$  が発火する。

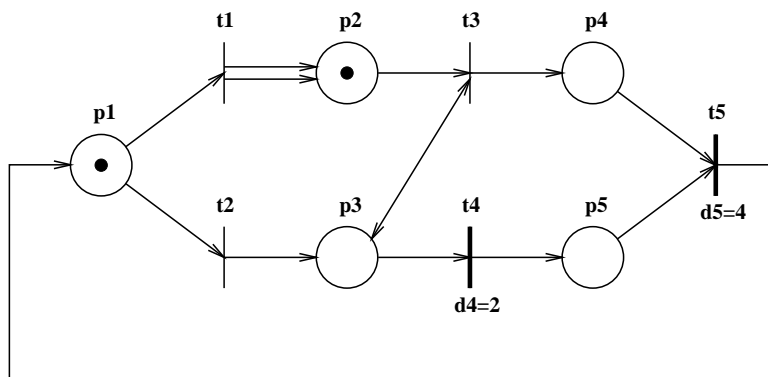


図 22  $d_4+d_5$  時刻経過後の時間ペトリネットグラフ

図 21において  $d_5$  時刻経過後、 $t_5$  が発火すると図 22になる。

## 4. シーケンス制御系のペトリネットによる表現

ここでは、シーケンス制御系を構成する要素およびSFCからペトリネットを表現する手法について述べる。

### 4.1 構成要素からペトリネットの表現

制御系では、部品・製品などの『物』の流れの制御が重要であり、システム中の設備・機械や搬送路等の資源の割り当て・有効利用を図り、並列同時進行性プロセスに特徴的なデッドロック、ボトルネック等を回避することが必要となる。

ペトリネットによるシステムの表現にはいくつかの利点がある。

#### 1. 視覚的表現

グラフ的な方法でモデルを記述できる。また、正確に、形式的に定義することができ、確かな意味を持たせることができる。加えて、イラスト的であるから理解するのが容易である。

#### 2. シミュレーション

得られたグラフを用いてシミュレーションが可能である。

#### 3. 数学的解析

デッドロックのような、システムにとって望ましくない性質を数学的に解析することが出来る。

システムの構成要素として以下のようなものが挙げられる。

- 制御対象            — — —   例 :計量タンク)
- 制御操作用機器   — — —   例 :電磁バルブ)
- 監視用機器       — — —   例 :表示ランプ)
- 検出用機器       — — —   例 :レベルスイッチ)
- 制御指令用機器   — — —   例 :押しボタンスイッチ)

これら構成要素のペトリネット表現例を示す。

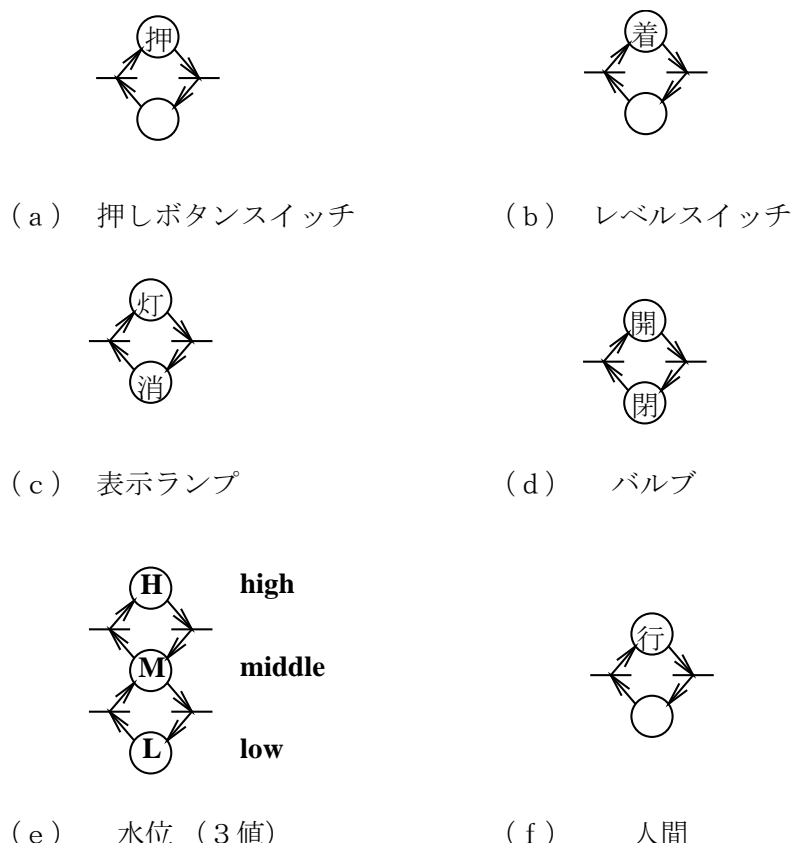


図 23 構成要素のペトリネット表現

スイッチは押されているか、押されていないかによって 2 値 (図 23(a)) で表現し、レベルスイッチはオン (着) の状態であるか、そうでないかによって 2 値 (図 23(b)) 表現する。表示ランプは点灯しているか、消えているかによって 2 値 (図 23(c))、バルブは開いているか、閉じているかによって 2 値 (図 23(d)) でそれぞれ表す。

タンクの表現は液位によりいろいろな表現が考えられるが、本研究では 3 値 (図 23(e)) で表現した。また、人間をシステムの一部とし、制御系を構成した。具体的には、行動を起こしているか、否かによって 2 値 (図 23(f)) 表現した。

制御装置に対する制御対象の動作は、プレースとトランジションをアークで結ぶことによって表現される。計量タンクの例を使って説明すると、図 24は、タンク内の液位が Low にあり、電磁バルブは開いた状態になっており、レベルスイッチはオンを示している。

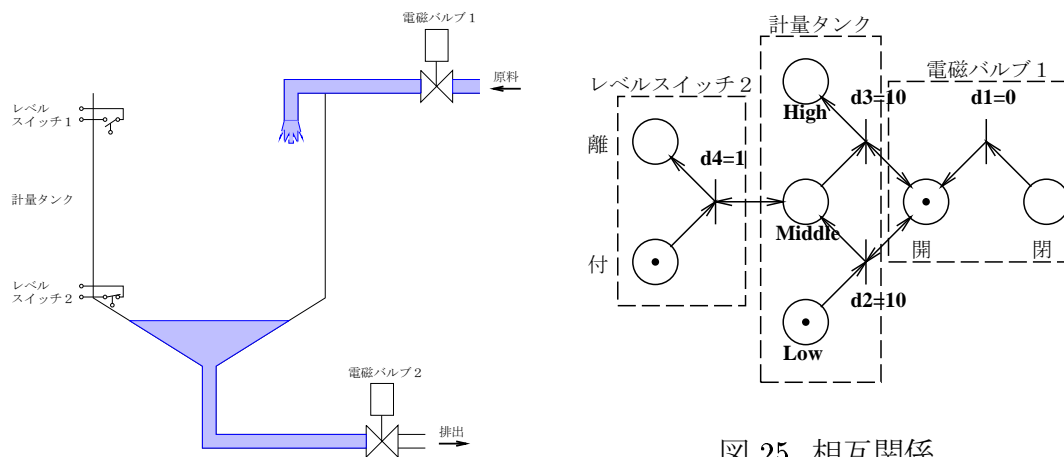


図 24 計量タンクシステム

この状態をペトリネットで表現すると図 25 のようになる。

## 4.2 SFC 要素からペトリネットの表現

ここでは、直列シーケンス、選択シーケンス、並行シーケンスと特別な変換を行うトランジション条件について変換方法を示す。

### 1. 直列シーケンス (表 1)

	SFC	PN
直列移行		
セット/リセット		
非保持 (セット&リセット)		

表 1 直列シーケンス

- アクションブロックが非保持(N)の場合、tr-n にダミープレース (D-n) より入力アークを引き、tr-b にアクションブロックの元の状態より入



出力1のアーキを引く。

- 備考 :もし、セット(S)/リセット(R)と同様の扱いをすると、トランジションの条件と接続した際に、tr-nが再度発火可能になる(表2)。またtr-bにアクションブロックの元の状態より入出力1のアーキが引かれてなければ、元の状態に戻る前に次状態に進むことが可能になってしまう。

	SFC	PN
非保持		

表2 誤ったアクションブロックの扱い (非保持)

表1のアクションブロックは一つであるが、一つのステップに対してだされる作業命令が2つ以上の場合、上側のブロックを優先して行う。このSFCをグラフ表現すると表3で表される。

	SFC	PN
4. ロボットの制御		

表 3 アクションの優先順位

2. 選択シーケンス、並行シーケンス (表 4)

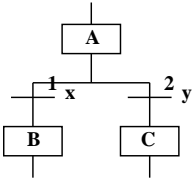
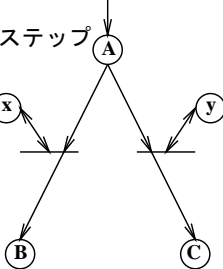
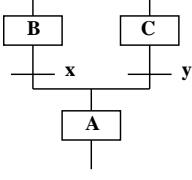
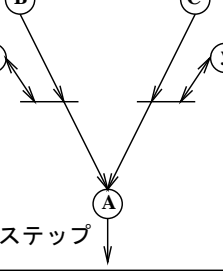
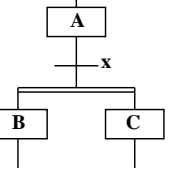
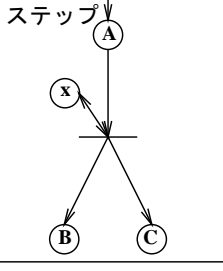
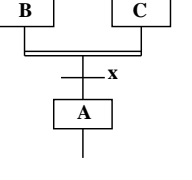
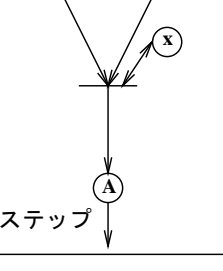
	SFC	PN
選択分岐		
選択結合		
並列分岐		
並列結合		

表 4 選択シーケンス、並行シーケンス

### 3. その他(表5)

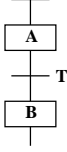
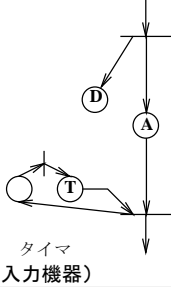
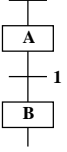
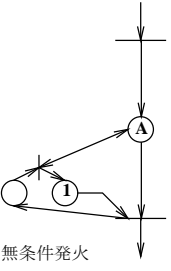
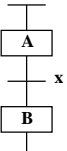
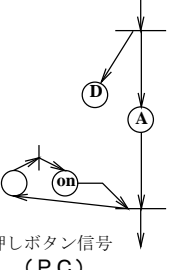
	SFC	PN
タイマ		 <p>タイマ (入力機器)</p>
無条件発火		 <p>無条件発火</p>
押しボタン		 <p>押しボタン信号 (PC)</p>

表5 その他

- タイマ
  - － 条件成立後、次のステップへ移行と同時にリセットされる。
- 無条件発火 (1)
  - － 条件成立後、次のステップへ移行と同時にリセットされる。  
ステップとトランジションの間をアークで結ぶ。

- 押しボタン
  - 条件成立後、次のステップへ移行と同時にリセットされる。

4. ダミープレース (表 6)

	SFC	PN
人間の介在		
セット/リセット&タイム		
非保持&タイム		

表 6 ダミープレース

- 人間の介在
  - SFC を含む制御系には 『アナンシエータで人間に知らせて対応を待つ』とある。しかし、押しボタンスイッチ、選択スイッチは 『

間の介入』なくして次の状態に遷移することはできない。本研究では、『人間の介入』をシステムの一部と考える。もし、『人間の介入』について、ダミープレースを用いずに **tr-h** よりステップ A に入出力 1 のアークを結ぶ (表 7) と、

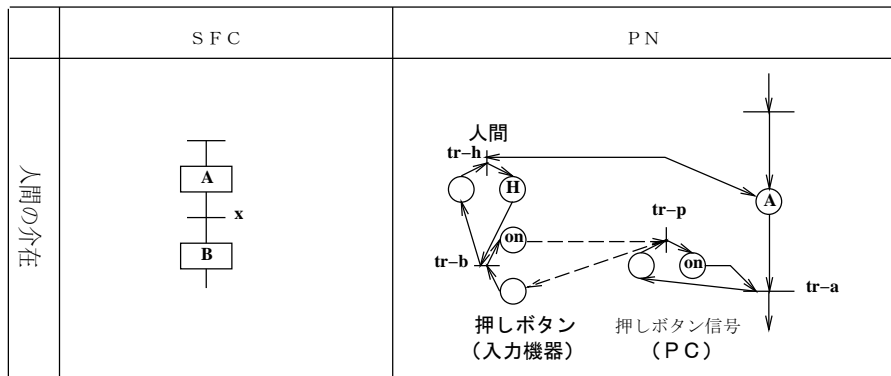


表 7 誤ったアクションブロックの扱い (『人間の介入』)

**tr-b**, **tr-p** の発火後、本来次のステップへ移動しなければいけない (**tr-a** のみ発火可能の) はずが、人間がもう一度介入することが可能 (**tr-h** が発火可能) となる。これを回避するため、ダミープレースを用いる (表 6)。

- タイマ & S

- タイマの条件成立がセット (S)/リセット (R) のアクションブロックの作業命令直後の時は、**tr-t** にダミープレース (D) より入力アークを引く (表 6)。

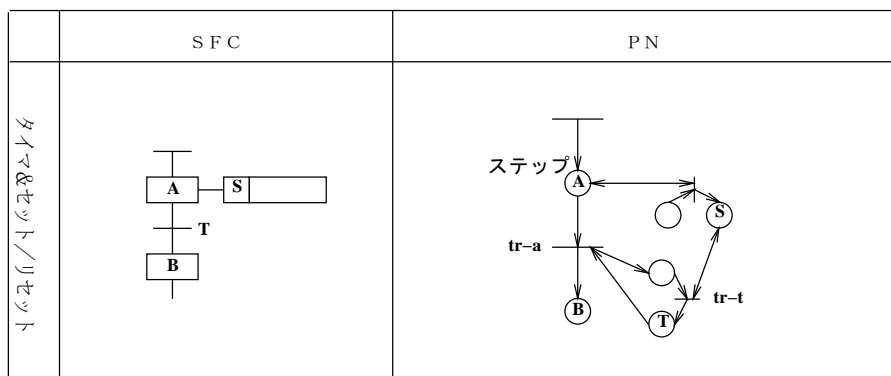


表 8 誤ったトランジションの扱い (セット/リセット)

もし、このダミープレースがないと、tr-a の発火の後に、ステップ B におけるアクションと tr-t の 2 つが発火可能になってしまう (表 8。このことより、ダミープレース (D) が必要となる。

- タイマ & N
  - タイマの条件成立が、非保持 (N) のアクションブロックの時は、タイマの条件成立と同時に次の状態へ移動させる。ダミープレースは不要。

### 4.3 遅延時間を考慮した時間ペトリネット

今まで述べてきた変換規則に従ってペトリネットグラフを構成し、状態を遷移させていくと、設計者が意図しない状態へ遷移してしまうことがある。また、時間ペトリネットをシーケンス制御系に用いたという例はほとんどない。

そこで本研究では、実際の物理現象をうまく反映させるために、従来の時間ペトリネットに新たな規則を追加する。

実際のシステムでは、制御対象、タイマについて遅延時間が存在する。制御対象について一例を挙げると、タンク内の液位が Low から Middle に変わるまでの時間経過などが考えられる。タイマはいうまでもなく、ある一定の時間を計測するものである。

制御対象、タイマに関して不都合が起こる原因は、時間ペトリネットでは規則5によってトランジションにおいて、発火が決定された時、発火に必要なトークンが予約されるためである。

実際のシステムを例にとって説明する。図26は計量タンクのレイアウトを示したものである。タンクの液位がLowで、バルブは開いた状態である。

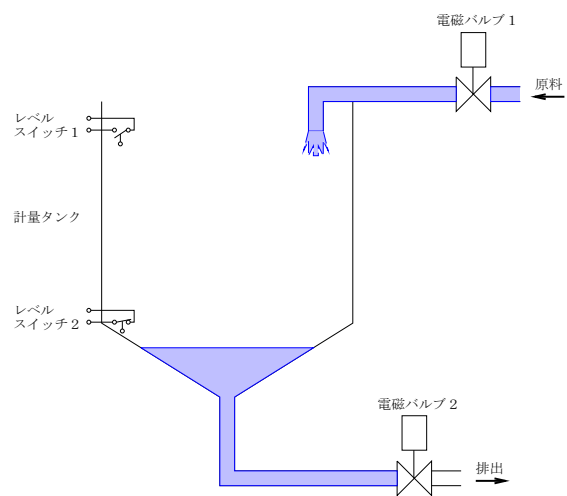


図26 計量タンクのレイアウト

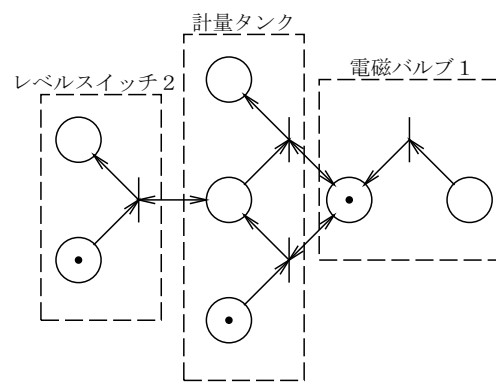


図27 ペトリネットグラフ

図26をグラフ表現したものが図27である。



図 28は、図 27にプレースとトランジションに番号を割り振ったものである。

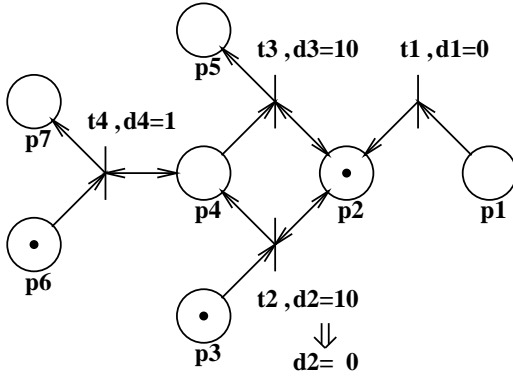


図 28 ペトリネットグラフ

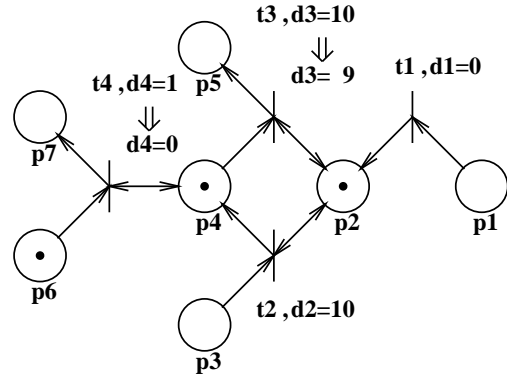


図 29  $t_2$ 発火後のグラフ

図 28において発火可能トランジションは $t_2$ である。10 時刻経過後、 $t_2$ は発火する。発火して得られたグラフが図 29である。

図 29において発火するトランジションは、1 時刻経過後に $t_4$ が発火する。 $t_4$ 発火後に得られたグラフが図 30である。

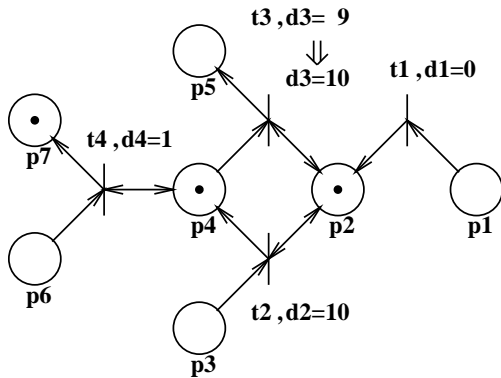


図 30  $t_4$ 発火後のグラフ (従来の TPN)

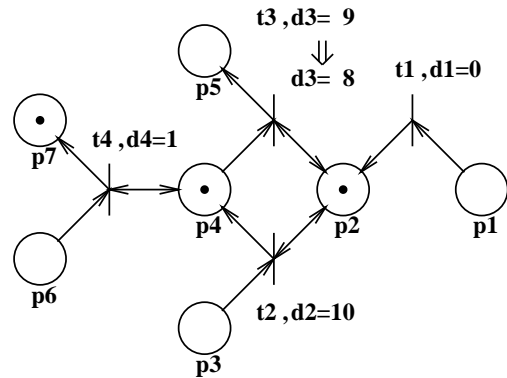


図 31  $t_4$ 発火後のグラフ (特別 TPN)

その際に、図 30の $d_3$ が示すように、遅延時間が 9 から 10 に戻される。これは

$p_4$ 内のトークンが、一度  $t_4$ の発火により取り去られ、その後もう一度、 $p_4$ にトークンが配分されるためである。

そのため、 $t_3$ が発火するのは(1+10)時刻の11時刻後である(図32)。

このことは実際の現象では、10時刻後に Low から High になるはずが、11時刻後に液位が変わることを示す。これは、始めに定めた設定を満足していない。

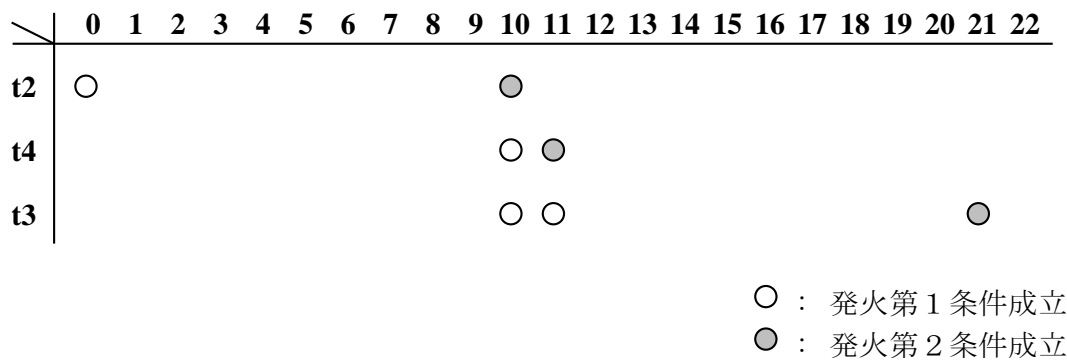


図32 タイムチャート(従来の TPN)

図32で発火第1条件というのは、特別規則1を満たしたことを示し、発火第2条件というのは、規則1を満たしたことを示す。

この問題を解決するために本論文では、規則1~5に以下の特別規則1を付け加える。

**特別規則1** あるプロセス  $p_i$ に対して、実行競合を起こしているトランジション  $(t_j, t_k)$ があるとする。 $t_k$ の遅延時間が先に0になるとき、 $t_j$ の遅延時間は下記条件を満たす時、遅延時間を元の値に戻さなくてよいものとする。

$$\hat{p}_i + b_{ik}^+ - b_{ik}^- \geq b_{ij}^- \quad (25)$$

この規則を追加して  $t_4$ 発火後に得られたグラフが図31である。

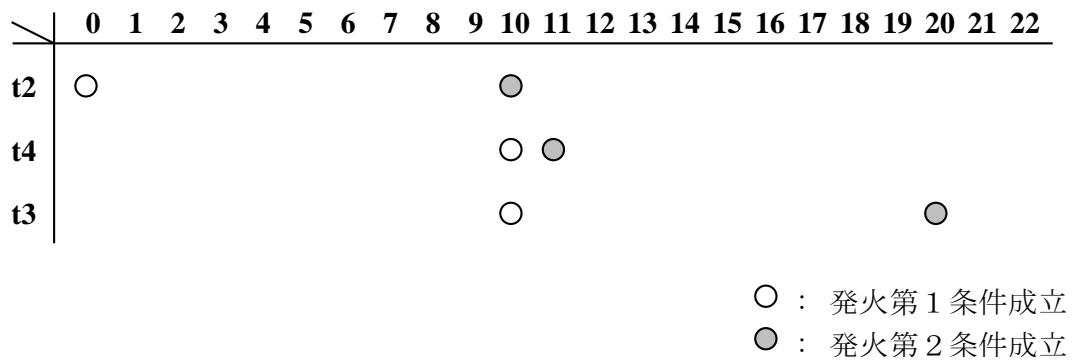


図 33 タイムチャート (特別 TPN)

これにより、 $t_3$ は9 から 8 に時刻はそのまま経過し、10 時刻後に  $t_3$ は発火する (図 33)。

## 5. シミュレーション

第4章で述べた手法を使ってペトリネットに変換し、シミュレーションを行う。そのことにより設計のチェックを行い、設計のバグをとることにかつようすることにより、提案した手法の有効性を示す。

今回提案する手法を適用した2つのシステムは、自動ポンプ運転システムと自動サイロ運転システムである。

### 5.1 自動ポンプ運転システム

1. SFC 設計 (第2章 2.1節参照)
2. 問題提起

ここでは、詳細レベルの SFC 設計時に 2箇所の設計ミスがあったと仮定する (図 34)。

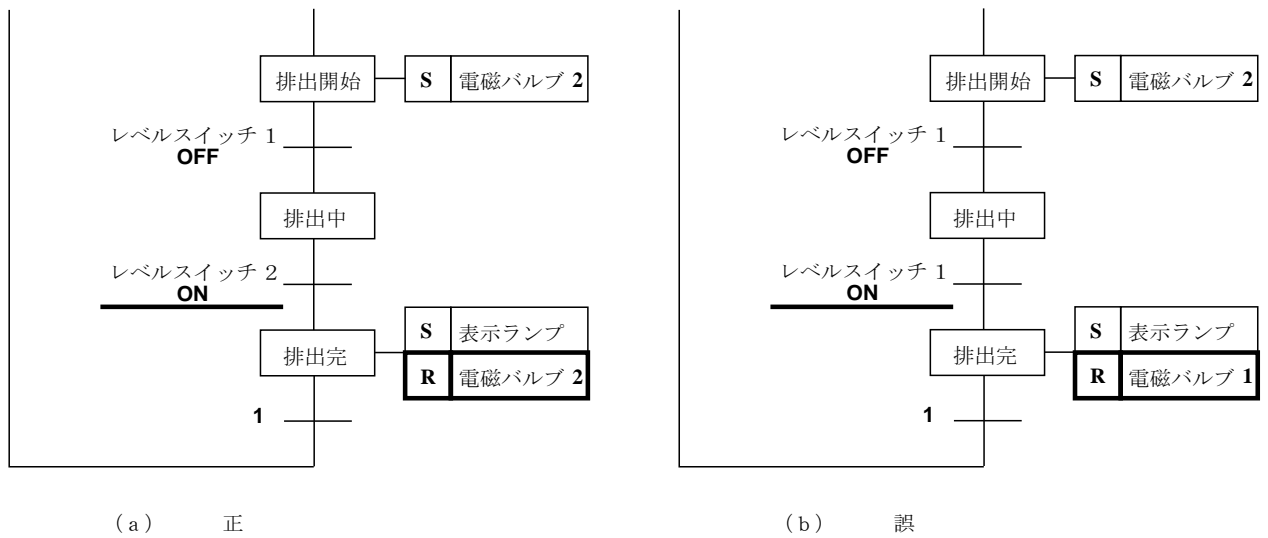


図 34 詳細レベルの SFC

『レベルスイッチ 2.ON』と 『電磁バルブ 2.R』に設計ミスがあったと仮定す

る。それぞれ『レベルスイッチ 2.ON』を『レベルスイッチ 1.ON』に、『電磁バルブ 1.R』を『電磁バルブ 1.R』に変える。

この誤った SFC を含む制御系について解析を行う。

### 3. 手順

SFC を含む制御系の解析を以下の手順で行う。

解析手順：

- (a) 詳細レイアウトおよび SFC 設計
- (b) SFC を含む制御系をペトリネット表現
- (c) 状態方程式
- (d) 可達木作成
- (e) 解析
- (f) 設計ミスがあれば、箇所を特定
- (g) SFC 再設計
- (h) (b) に戻る

まず、詳細レイアウトおよび SFC を作成する。作成されたレイアウトおよび SFC を第 4 章の変換手法に従い、ペトリネット表現する。グラフに対してはプレースとトランジションに固有の番号をつける。グラフから接続行列、状態ベクトルを求め、状態方程式をつくる。状態方程式に対して、発火可能トランジションを求め、可達木を作成する。作成された可達木に対し、解析を行う。

解析を行った結果、設計ミスがなければここで終了する。もし、設計ミスが見つければ、その箇所を特定し、再設計を行う。そして、また (b) に戻り、解析を行う。

#### 4. シーケンス制御系のペトリネットによる表現

誤った詳細レベルの SFC(図 5(b)) とシステムの構成要素をグラフ表現する (図 35)。

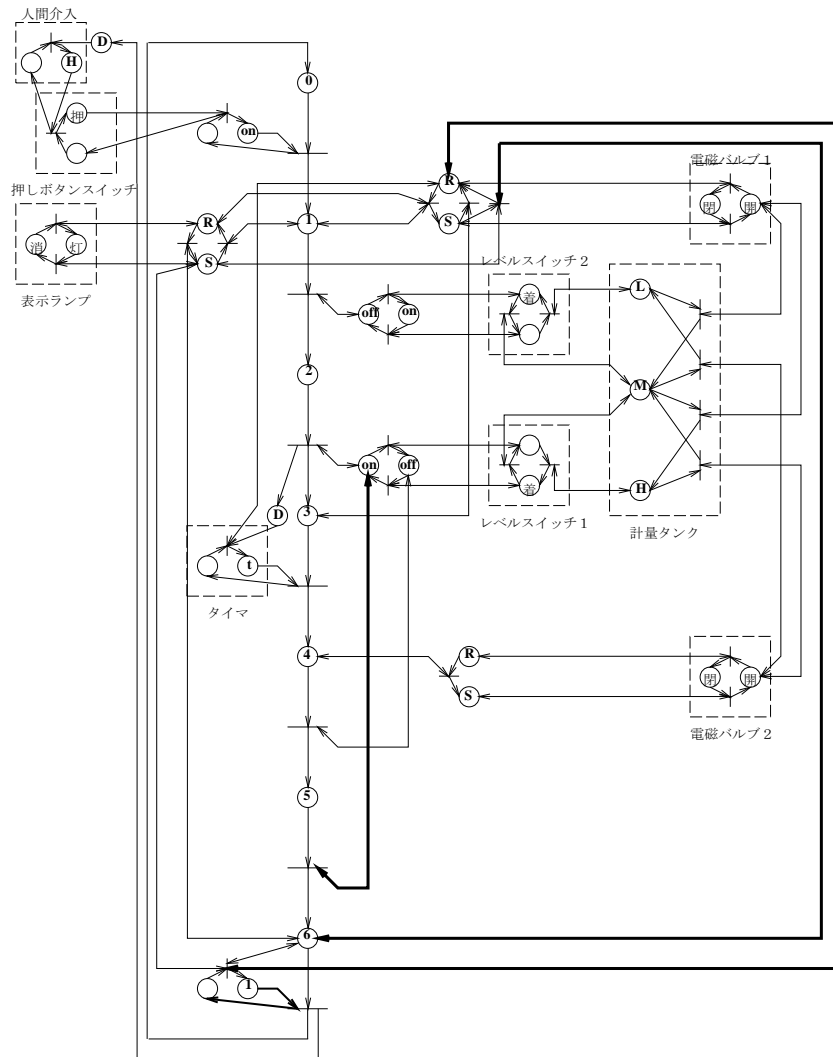


図 35 変換規則によるペトリネットグラフ

図 35に対して、プレースおよびトランジションに番号をつける (図 36)。

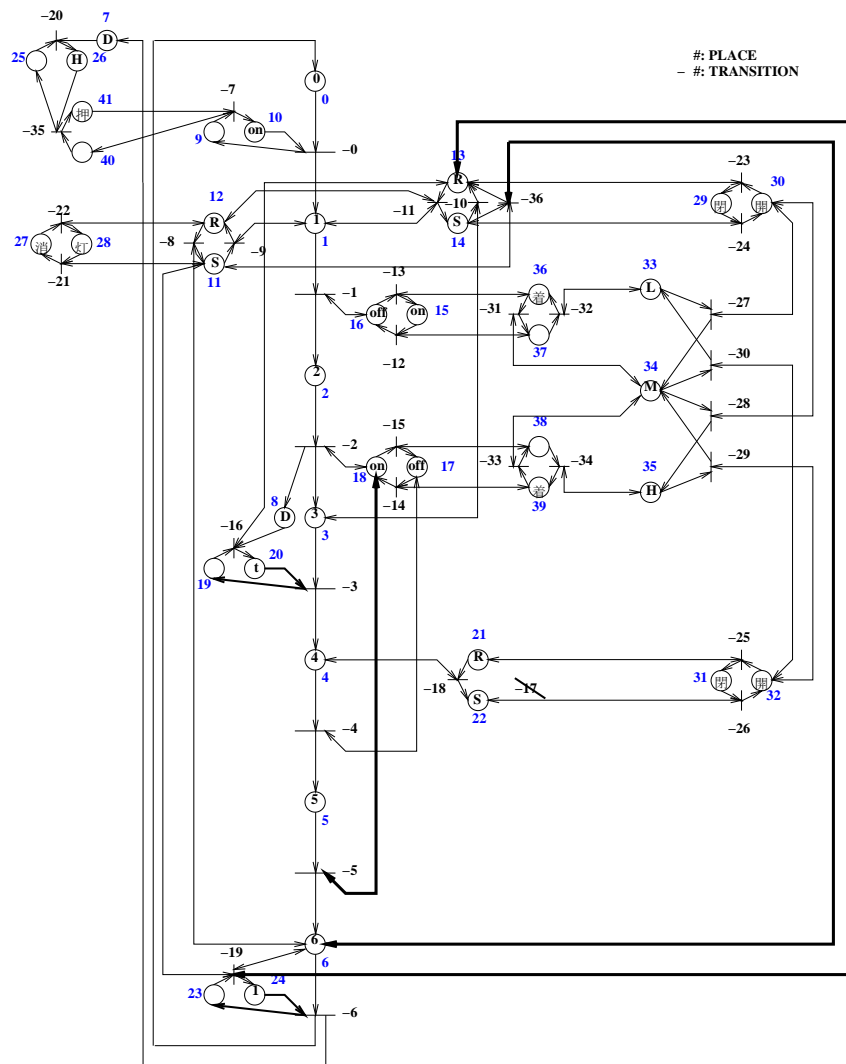


図 36 変換規則によるペトリネットグラフ～番号付け～

作成されたグラフにおいて、システムの時間経過はトランジションの遅延時間によって表現される。図 36 に遅延要素を加えるトランジションおよび時間を表 9 に示す。

tr-	遅延要素	遅延時間 (msec)	スキャン 回数(回)
20	人間が介入するまでの時間	500	500
21	表示ランプが点灯するまでの時間	10	10
22	〃 が消灯するまでの時間	10	10
23	電磁バルブ 1 が開くまでの時間	20	20
24	〃 が閉じるまでの時間	20	20
25	電磁バルブ 2 が開くまでの時間	20	20
26	〃 が閉じるまでの時間	20	20
27	水位が Low から Middle になるまでの時間	1000	1000
28	水位が Middle から High になるまでの時間	1000	1000
29	水位が High から Middle になるまでの時間	1000	1000
30	水位が Middle から Low になるまでの時間	1000	1000
31	水位が変わってからレベルスイッチ 2 が離れるまでの時間	10	10
32	〃 が入るまでの時間	10	10
33	水位が変わってからレベルスイッチ 2 が入るまでの時間	10	10
34	〃 が離れるまでの時間	10	10
35	押しボタンが押されるまでの時間	10	10

表 9 自動ポンプ運転システムの遅延時間

今回のシミュレーションでは、遅延時間の扱いは時間の代わりにスキャンの回数を用いる。



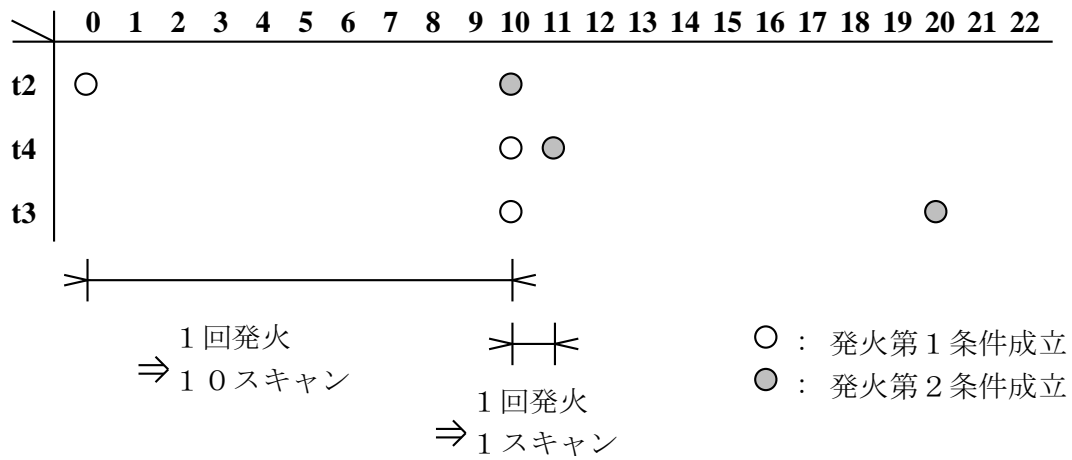


図 37 スキャン

1回のスキャンは、ペトリネットグラフでトランジションの1回の発火を表す。ある時刻に発火可能なトランジションが存在せず、遅延時間が0になれば発火可能という場合には、その時刻でスキャン回数が一番少ないものが0になるまで空打ちをする。

#### 5. システムの検証

作成されたペトリネットグラフを状態方程式に表し、可達木を求めると、図 38が得られる。

状態数 = 36

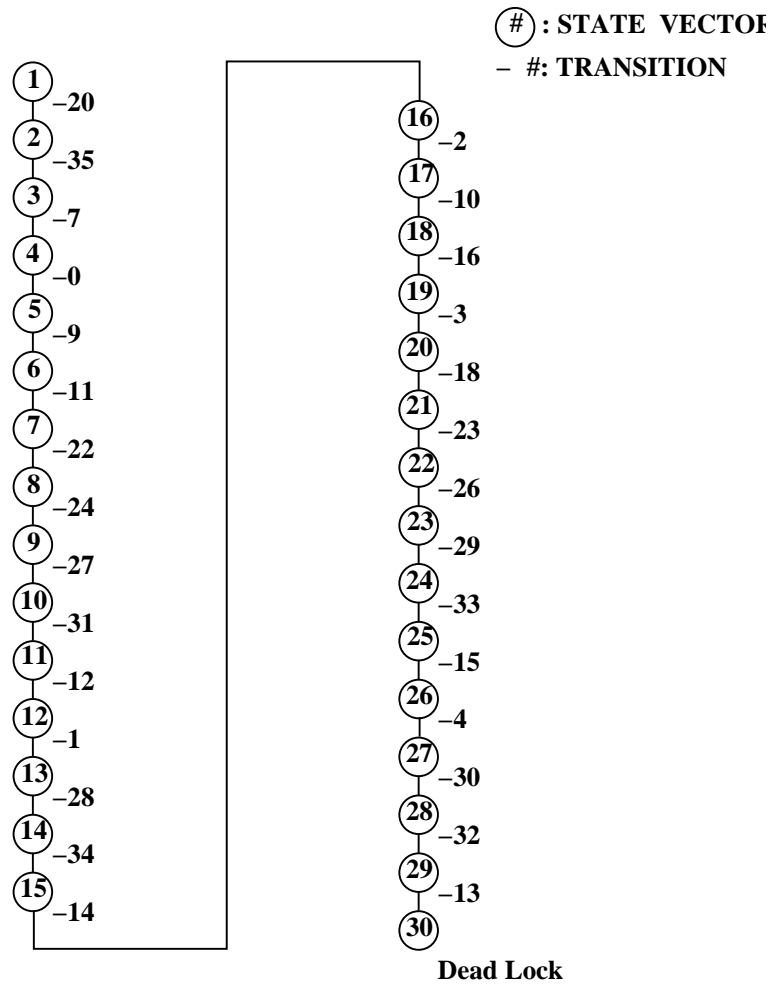


図 38 ポンプの可達木

図 38で ○の中に数字があるのは、状態ベクトル(State Vector)、つまりある時刻におけるシステムの状態を表している。同様に、状態ベクトル間にある ”-” の後にうたれている数字は発火トランジションを表している。

図 38において、システムの検証を行う。

検証 No.	検証項目	P[1]
1	液位が High の時、レベルスイッチ 1 が OFF になっている。	35,38
2	液位が High の時、レベルスイッチ 2 が ON になっている。	35,36
3	液位が Middle の時、レベルスイッチ 1 が ON になっている。	34,39
4	液位が Middle の時、レベルスイッチ 2 が ON になっている。	34,36
5	液位が Law の時、レベルスイッチ 1 が ON になっている。	33,39
6	液位が Law の時、レベルスイッチ 2 が OFF になっている。	33,37
7	電磁バルブ 1 と 2 が同時に開いている。	30,32
8	初期状態の時、表示ランプがついている。	0,28
9	SFC が定められた順序に従ってステップが進行していない。	

表 10 自動ポンプ運転システムの検証項目

表 10 については、状態が変わってから遅延時間経過後も検証項目に挙げる状態が存在すれば、異常とみなす。表 10 で P[1] は、状態ベクトルの中で 1 をとる要素を表す。

一つ例を挙げれば、『液位が Middle の時、レベルスイッチ 1 が ON になっている』は、『液位が Middle になって (遅延時間経過後も)、レベルスイッチ 1 が ON になっている』という意味である。

表 10 に挙げている項目について解析を行ったところ、表 11 に示す結果が得られた。

異常検証 No.	発火 tr-	状態	発火可能 tr-
9	13	30	dead lock

表 11 自動ポンプ運転システムの検証結果

概念設計レベルでは、デッドロックが起こることはないように SFC は設計されている。しかし、tr-13 を発火させた後にデッドロックが起きている。

これは、『排出完』というステップが行われる前で起こっている。これより、それ以前の SFC を含む制御系について誤りがないかを調べる。

その結果、『レベルスイッチ 1.ON』に問題があることがわかる。次に本来どうあるべきかをもう一度考え、『レベルスイッチ 2.ON』に修正する。

修正された SFC を下記に示す(図 39(b))。

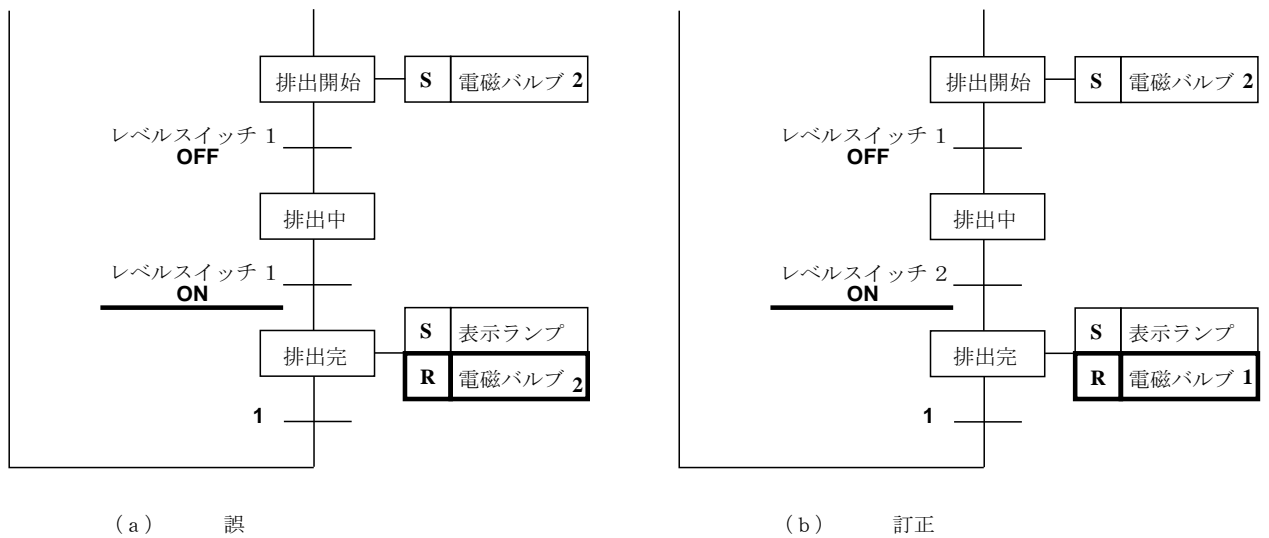


図 39 詳細レベルの SFC

この訂正された SFC を含む制御系について解析を先程同様の手順で行う。

訂正された SFC とシステムの構成要素をグラフ表現する(図 40)。

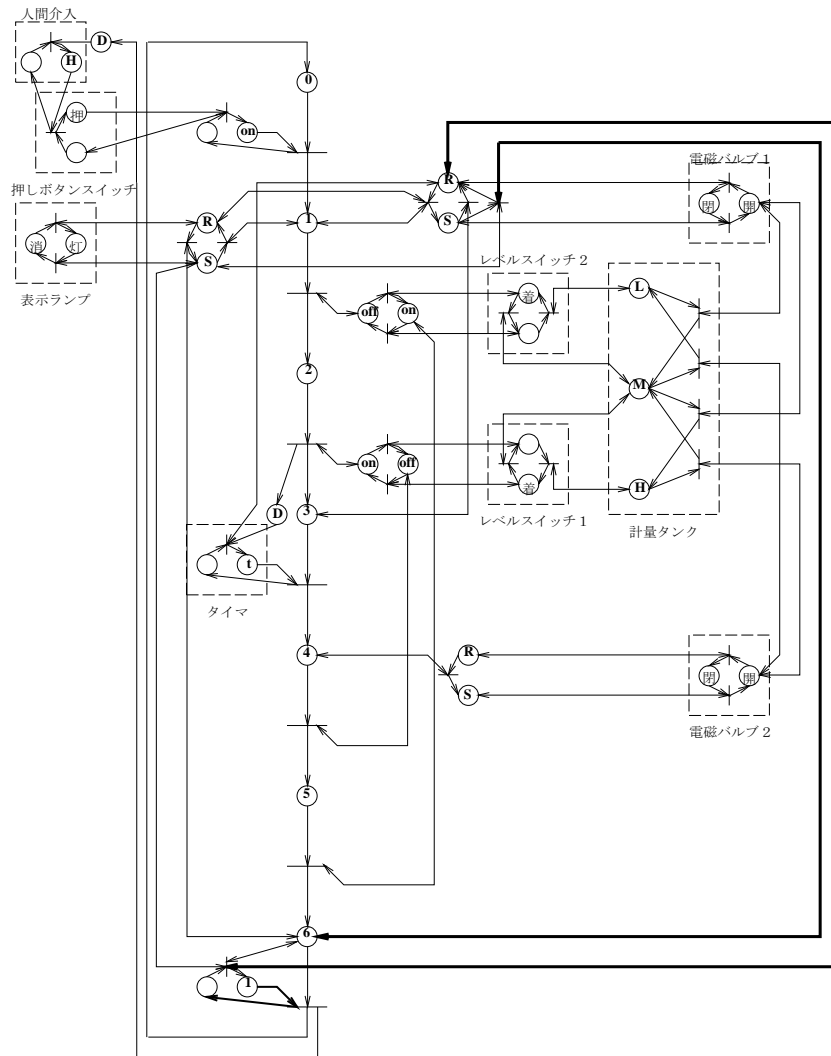


図 40 訂正されたペトリネットグラフ

図 40に対して、プレースおよびトランジションに番号をつける (図 36)。

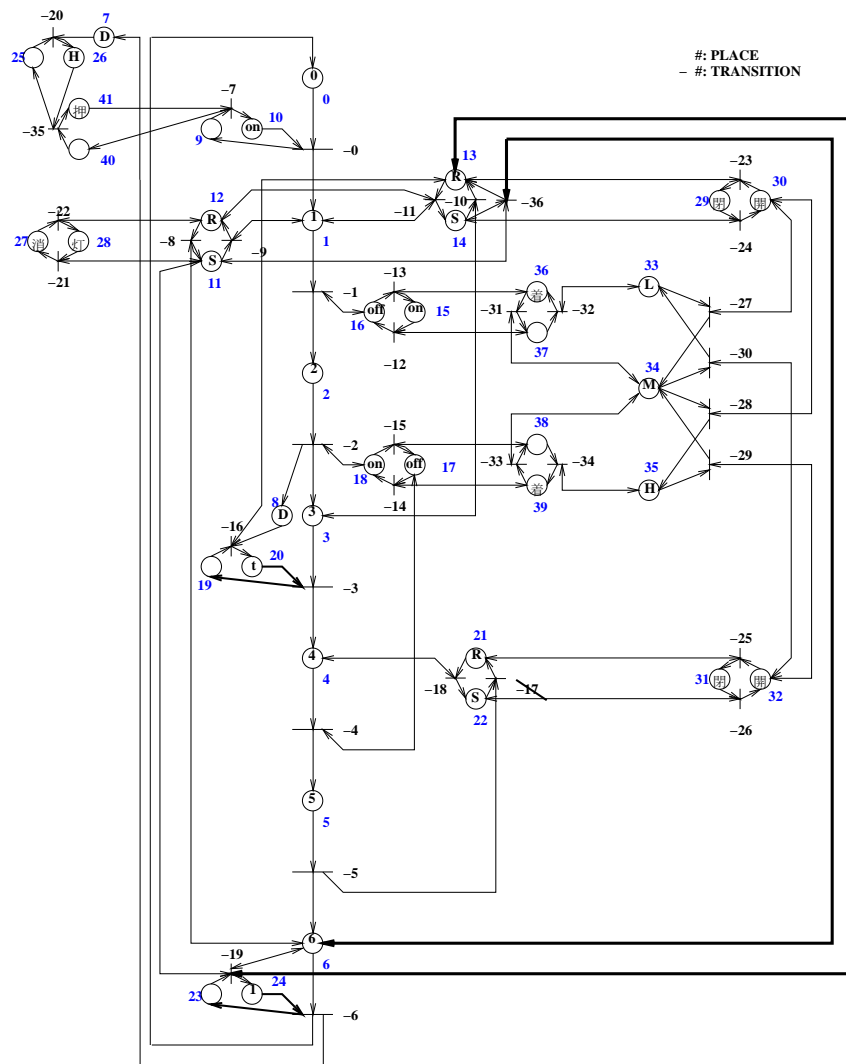


図 41 変換規則によるペトリネットグラフ～番号付け～

作成されたグラフにおいて、遅延要素を加える。加えるトランジションは先程示した表 9 と同様である。

### 6. システムの検証

作成されたペトリネットグラフを状態方程式に表し、可達木を求めると、図

42が得られる。

状態数 = 59

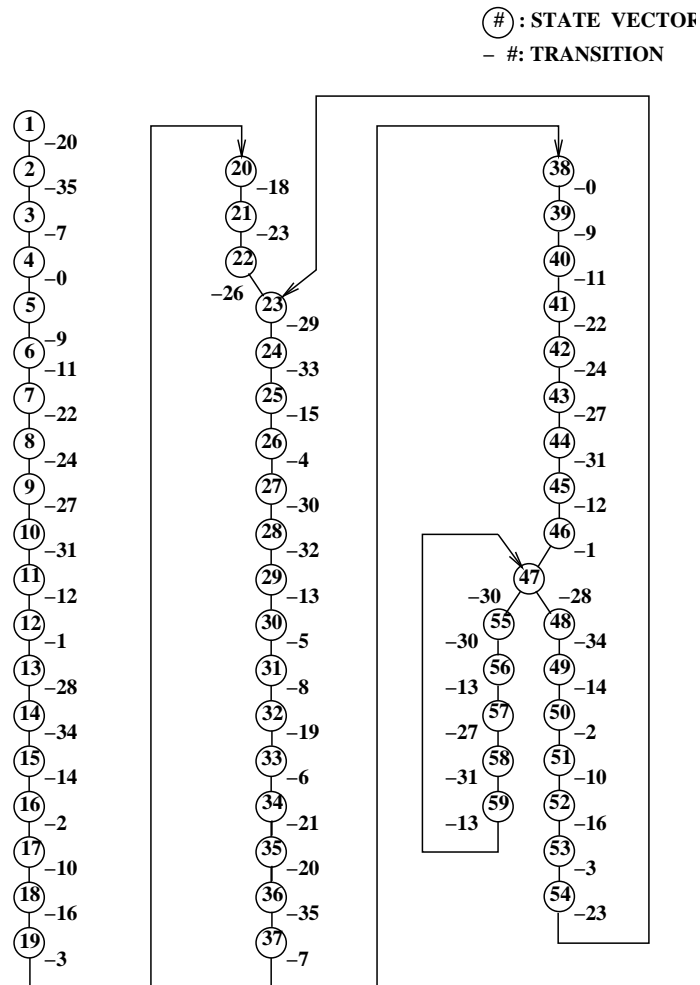


図 42 ポンプの可達木

図 42において、システムの検証を行う。検証項目は表 10に示した通りである。

表 10に挙げている項目について解析を行ったところ、表 12に示す結果が得られた。

異常検証 No.	発火 tr-	状態	発火可能 tr-
7	-	42~59	-

表 12 自動ポンプ運転システムの検証結果

設計仕様では、注入・排出をわけているため電磁バルブの1, 2 が同時に開くことはない。しかし、2 つのバルブが同時に開く現象が起きている。この2 つのバルブの作業命令のどちらかに問題があることを発見する。そうして『電磁バルブ 1.R』に問題あることを見つけ、『電磁バルブ 2.R』に修正する。

修正した SFC を下記に示す (図 43(b))。

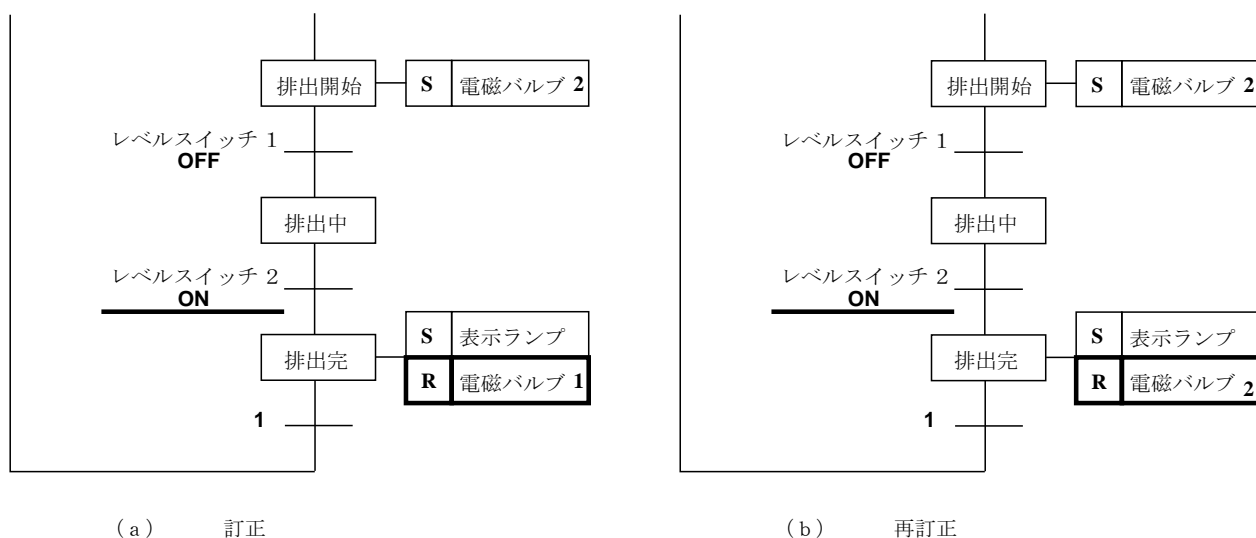


図 43 詳細レベルの SFC

この訂正された SFC を含む制御系について解析を先程同様の手順で行う。

訂正された SFC とシステムの構成要素をグラフ表現する (図 44)。



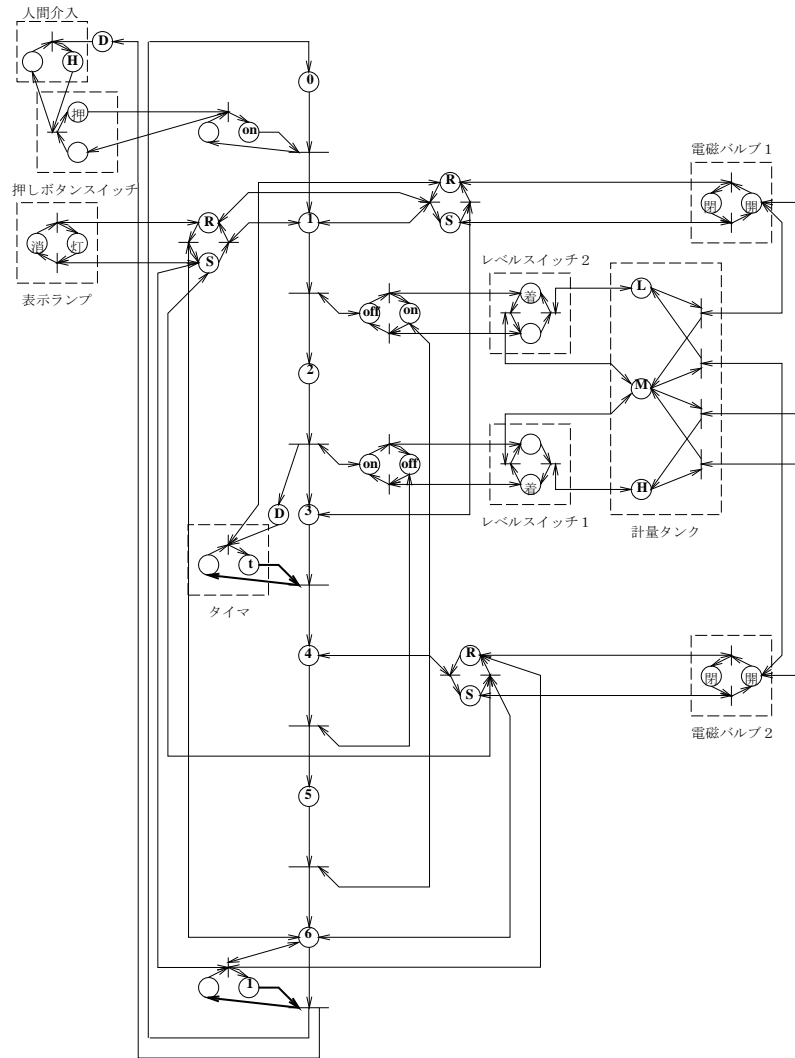


図 44 訂正されたペトリネットグラフ

図 44 に対して、プレースおよびトランジションに番号をつける (図 45)。

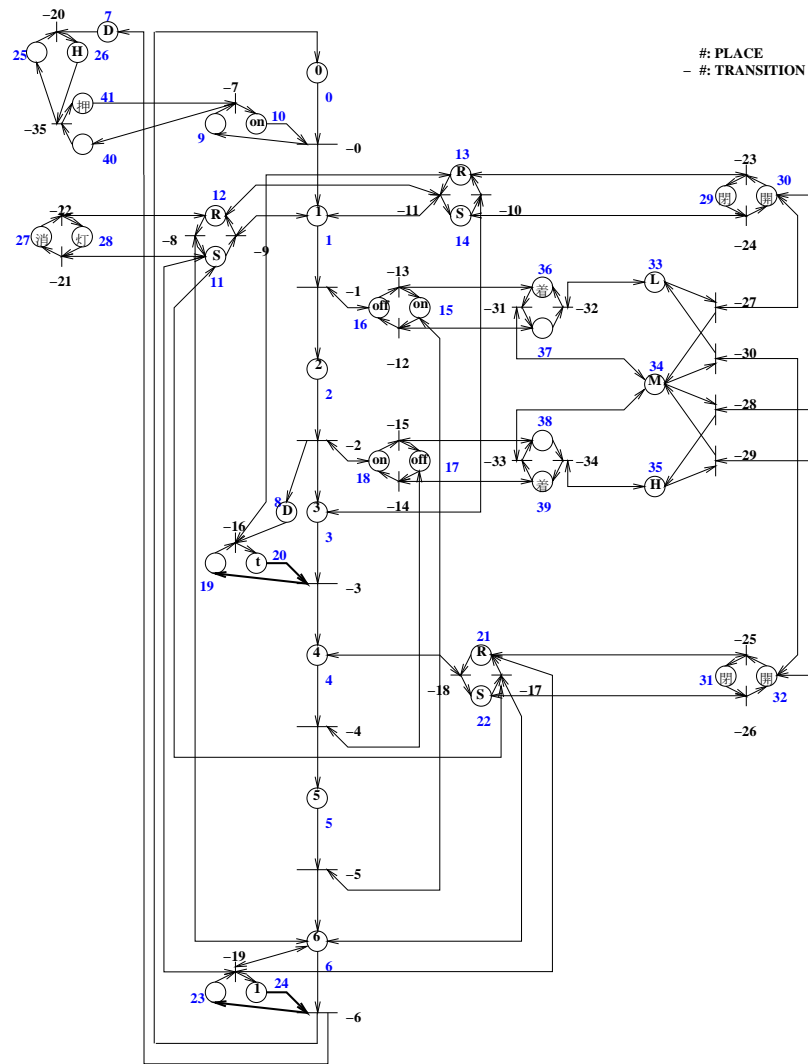


図 45 変換規則によるペトリネットグラフ～番号付け～

作成されたグラフにおいて、遅延要素を加える。加えるトランジションは先程示した表 9 と同様である。

### 7. システムの検証

作成されたペトリネットグラフを状態方程式に表し、可達木を求めると、図

46が得られる。

状態数 = 30

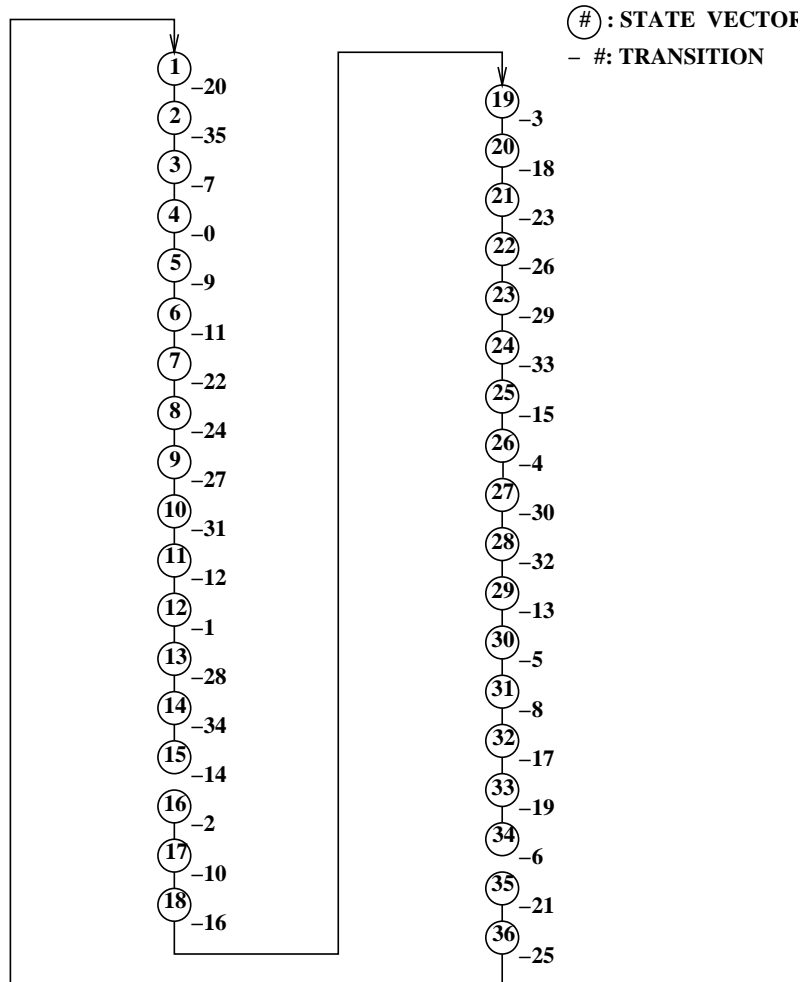


図 46 ポンプの可達木

図 46において、システムの検証を行う。検証項目は表 10に示した通りである。

表 10に挙げている項目について解析を行ったところ、表 13に示す結果が得られた。

異常検証 No.	発火 tr-	異常状態ベクトル	発火可能 tr-
-	-	-	-

表 13 自動ポンプ運転システムの検証結果

今回修正された SFC を含む制御系には、チェック項目すべてにパスして問題がないことがわかった。これで、SFC を含む制御系の設計を終了する。

## 5.2 自動サイロ運転システム

### 1. SFC 設計

図 47 に示す簡単なプロセスを例にとって SFC の設計を行う。

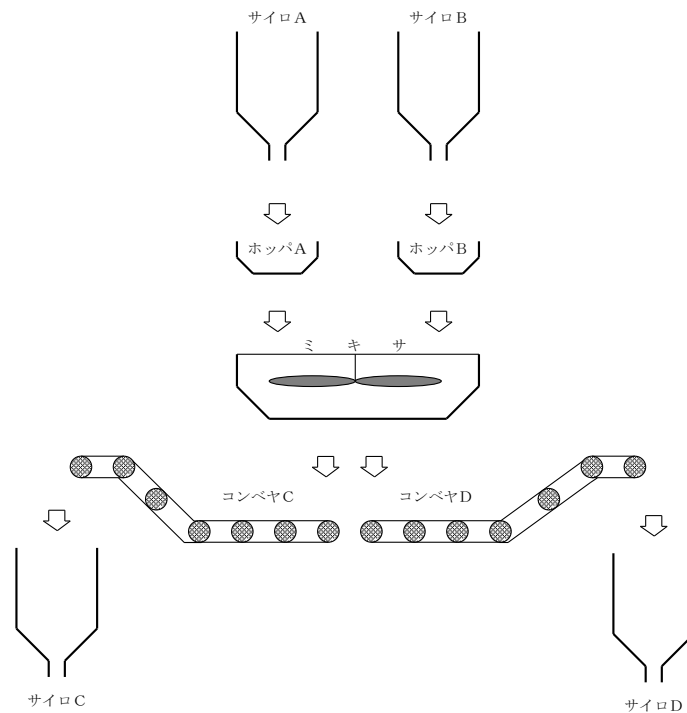


図 47 サイロのレイアウト

#### (a) 概念設計レベル

このレベルでは、システムの目標を把握する。サイロシステムの目標は、

『システムの目標 :サイロに貯蔵された2種の原料から混合した原料を作り、2つのサイロに貯蔵する』

ことである。

この目標から、大きく何をすべきかを機能系統図に示す(図 48(a))。得られた機能系統図から機械的にマクロレベルの SFC (図 48(b)) が作成される。

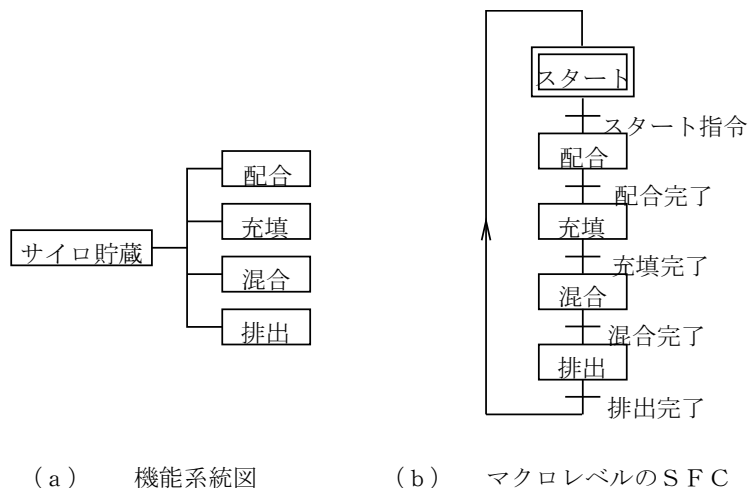


図 48 概念設計

(b) 機能設計レベル

このレベルでは、制御対象に対する制御装置の動作を記述する。  
 前のレベルで作成されたマクロレベルの SFC に順序関係を与えると、  
 機能レベルの SFC(図 49) が得られる。

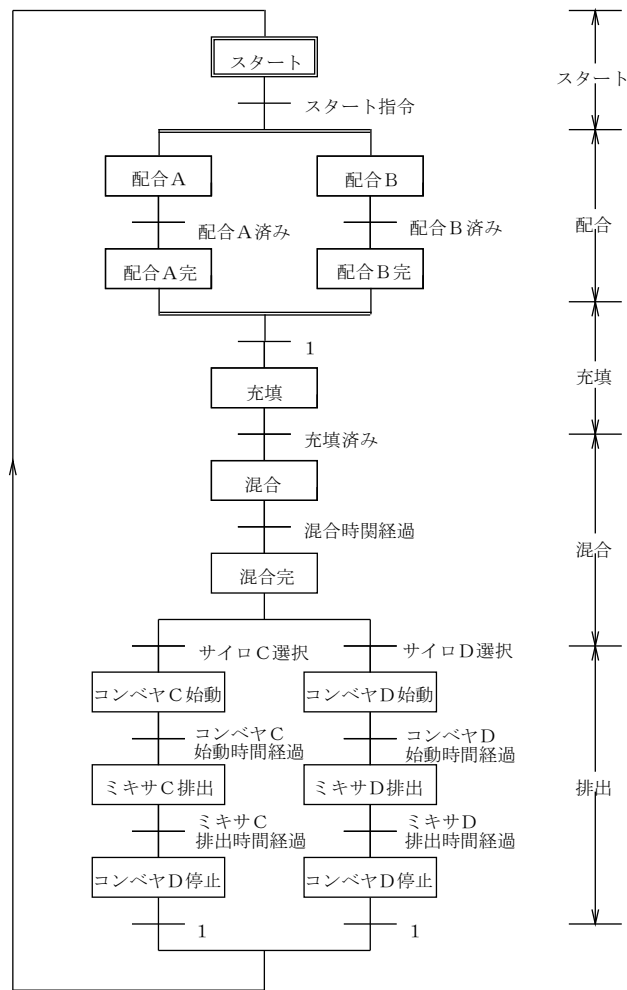


図 49 機能レベルの SFC

(c) 詳細設計レベル

このレベルではまず、機器・設備の把握を行う(下記参照)。

- 制御対象            --- サイロA, サイロB (2値表現),  
                              ホッパA, ホッパB (2値表現),  
                              ミキサ (2値表現),  
                              コンベヤC, コンベヤD (2値表現),  
                              サイロC, サイロD (2値表現),
- 制御操作用機器 --- サイロバルブA, サイロバルブB (2値表現)  
                              ホッパバルブA, ホッパバルブB (2値表現)  
                              ミキサモータ (2値表現)  
                              コンベヤモータMC, コンベヤモータMD (2値表現)  
                              ミキサバルブC, ミキサバルブD (2値表現)
- 監視用機器        --- なし
- 検出用機器        --- 重量計A, 重量計B (2値表現)  
                              ミキサタイマ (2値表現)  
                              移送時間タイマC, 移送時間タイマD (2値表現)  
                              コンベヤC始動タイマ (2値表現)  
                              コンベヤD始動タイマ (2値表現)
- 制御指令用機器 --- 押しボタンスイッチ (2値表現)  
                              サイロCD選択スイッチ (3値表現)

制御対象の構成要素の各機能を機器と対応させながら展開することにより、詳細レベルのレイアウトを作成する(図50)。

作成されたレイアウトと機能レベルのSFCより、詳細レベルのSFC(図51)を作成する。



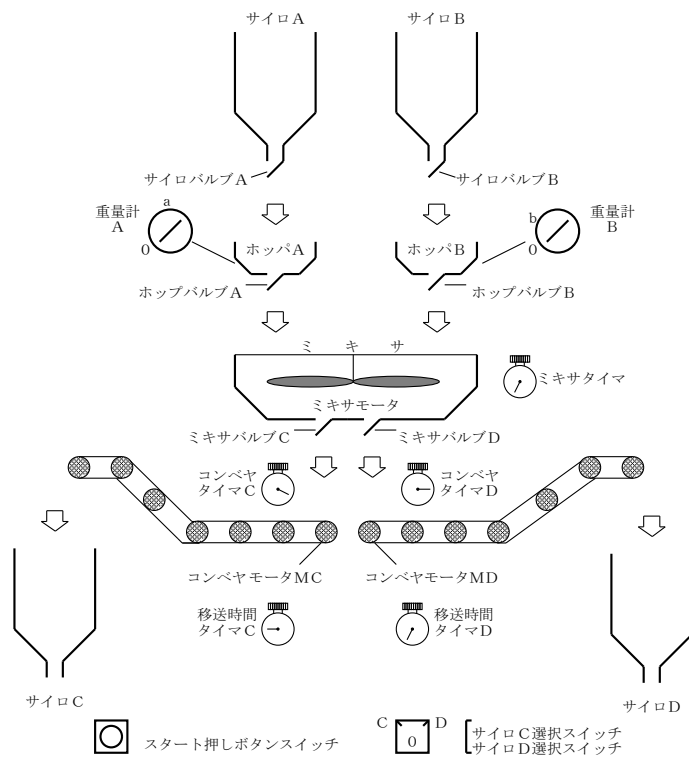


図 50 サイロの詳細レイアウト

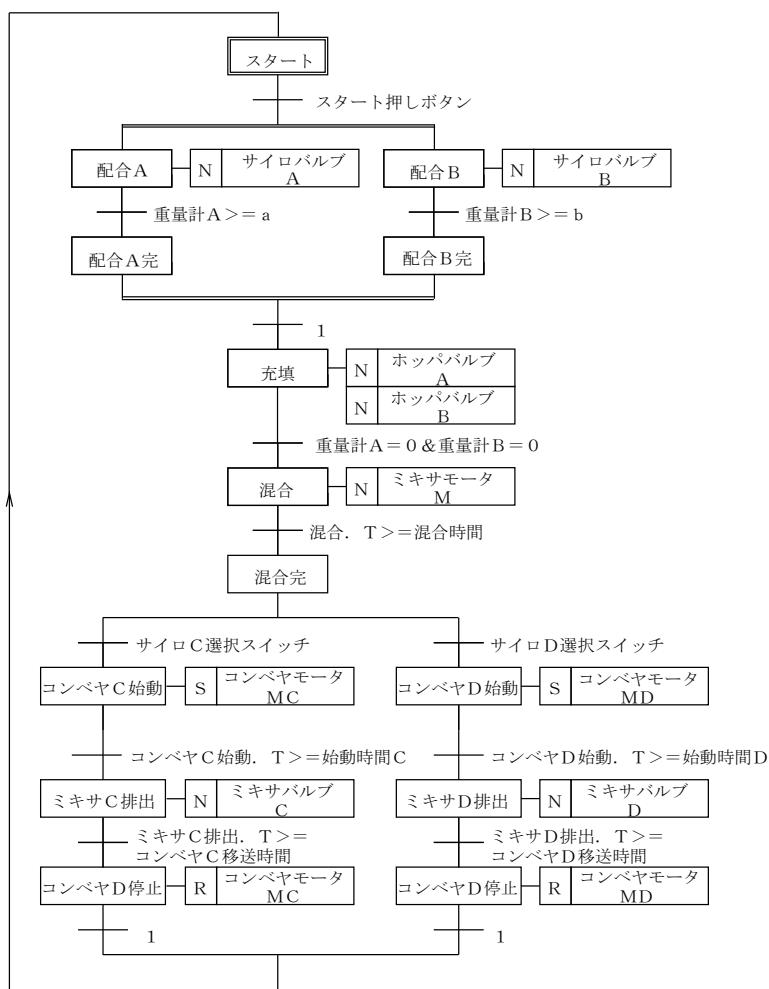
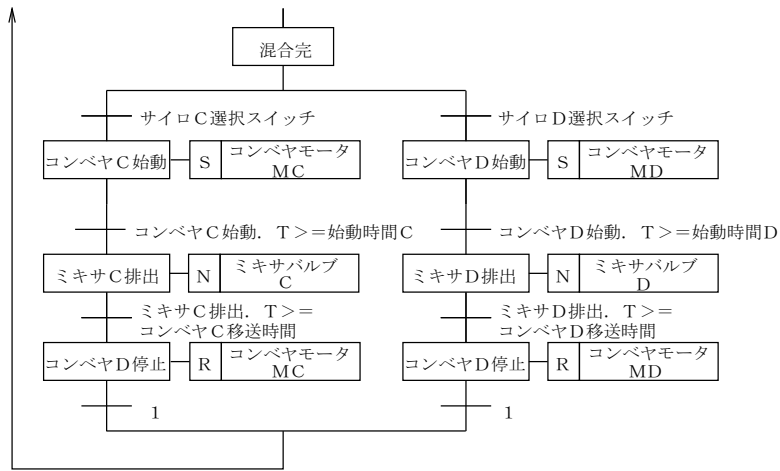


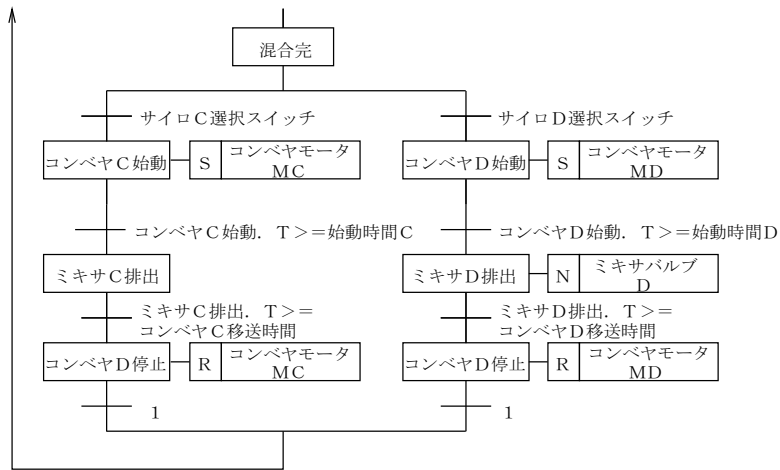
図 51 詳細レベルの SFC

## 2. 問題提起

ここでは、詳細レベルの SFC 設計時に設計ミスがあったと仮定する (図 52)。



(a) 正



(b) 誤

図 52 詳細レベルの SFC

『ミキサバルブ C.N』の作業命令を設計するのを忘れたと仮定する。この誤った SFC を含む制御系について解析を行う。

### 3. シーケンス制御系のペトリネットによる表現

誤った詳細レベルの SFC(図 51(b)) とシステムの構成要素をグラフ表現する (図 53)。

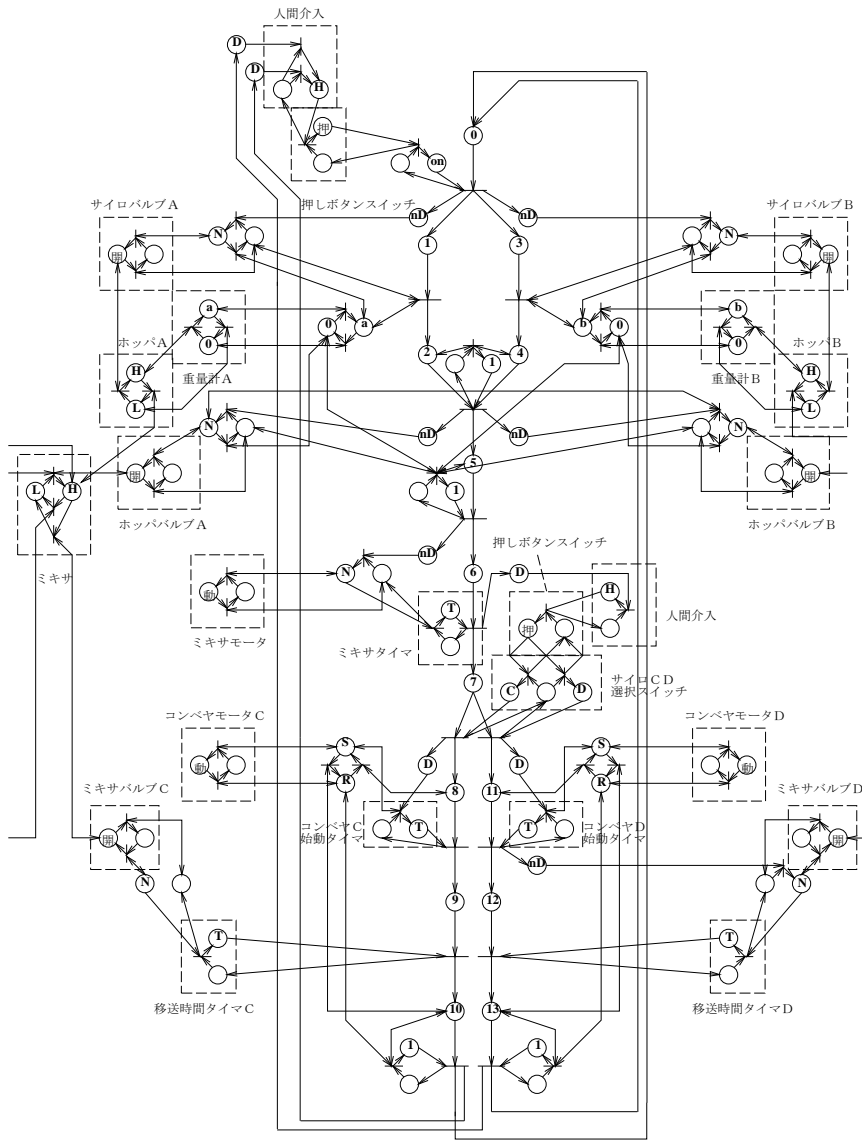


図 53 変換規則によるペトリネットグラフ

図 53 に対して、プレースおよびトランジションに番号をつける (図 54)。

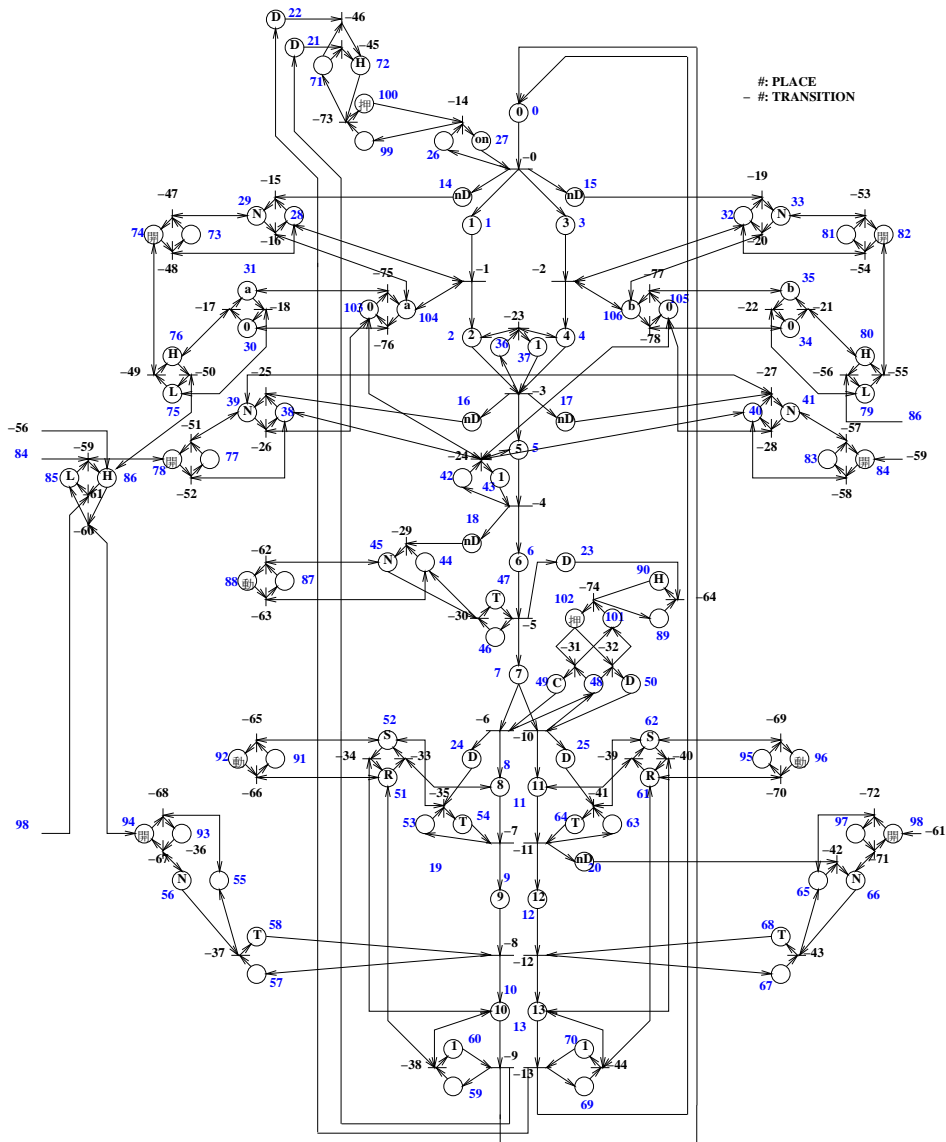


図 54 変換規則によるペトリネットグラフ～番号付け～

作成されたグラフにおいて、システムの時間経過はトランジションの遅延時間によって表現される。図 58 に遅延要素を加えるトランジションおよび

時間を表 14に示す。

tr-	遅延要素	遅延時間 ( $\mu$ sec)	スキャン 回数 (回)
45	人間が介入するまでの時間	500	500
46	〃	500	500
64	〃	500	500
17	原料が a の重さになり、重量計が値を示すまでの時間	10	10
18	原料が 0 の重さになり、	〃	10
21	原料が b の重さになり、	〃	10
22	原料が 0 の重さになり、	〃	10
47	サイロバルブ A が開くまでの時間	20	20
48	〃 が閉じるまでの時間	20	20
51	サイロバルブ B が開くまでの時間	20	20
52	〃 が閉じるまでの時間	20	20
53	ホッパバルブ A が開くまでの時間	20	20
54	〃 が閉じるまでの時間	20	20
57	ホッパバルブ B が開くまでの時間	20	20
58	〃 が閉じるまでの時間	20	20
67	ミキサバルブ C が開くまでの時間	20	20
68	〃 が閉じるまでの時間	20	20
71	ミキサバルブ D が開くまでの時間	20	20
72	〃 が閉じるまでの時間	20	20

65	コンベヤモータCが動くまでの時間	20	20
66	〃 が止まるまでの時間	20	20
69	コンベヤモータDが動くまでの時間	20	20
70	〃 が止まるまでの時間	20	20
62	ミキサモータが動くまでの時間	100	100
63	〃 が止まるまでの時間	100	100
49	サイロAの液位が Low から High (目標値) になるまでの時間	500	500
50	〃 が High から Low になるまでの時間	500	500
55	サイロBの液位が Low から High (目標値) になるまでの時間	800	800
56	〃 が High から Low になるまでの時間	800	800
59	ミキサの液位が Low から High (目標値) になるまでの時間	1000	1000
60	〃 が High から Low になるまでの時間	1000	1000
61	〃 が High から Low になるまでの時間	1000	1000
30	ミキサモータを動かしておく時間	1000	1000
35	コンベヤモータCを動かしておく時間	1000	1000
41	コンベヤモータDを 〃	1000	1000
37	ミキサバルブCを開いておく時間	2000	2000
43	ミキサバルブDを 〃	2000	2000
73	押しボタンが押されるまでの時間	10	10
74	選択ボタンが押されるまでの時間	10	10

表 14 自動サイロ運転システムの遅延時間

#### 4. システムの検証

作成されたペトリネットグラフを状態方程式に表し、可達木を求めると、図 55が得られる。

状態数 = 70

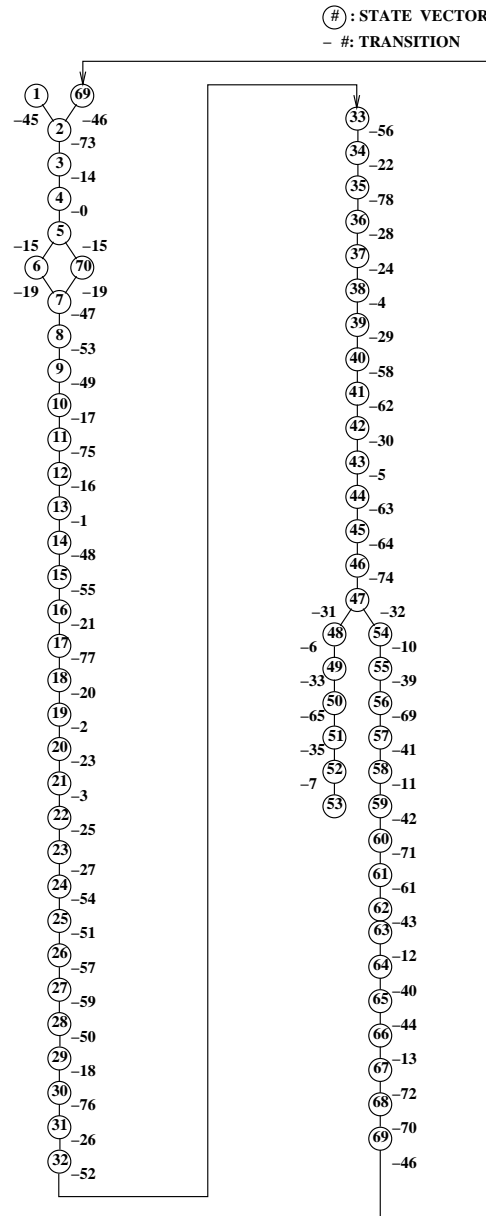


図 55 サイロの可達木

図 55において、システムの検証を行う。



検証 No.	検証項目	P[1]
1	ホップバルブ A(B) が開く時、ミキサモータが動いている。	78,88 (84,88)
2	ホップバルブ A(B) が開く時、ミキサバルブ C(D) が開いている。	78,94 (84,98)
3	コンベヤモータ C(D) が ON になる時、ミキサバルブ C(D) が開いている。	91,94 (95,98)
4	重量計 A(B) が a,b でないのに、ホップバルブ A(B) を開く。	30,78 (34,84)
5	ミキサモータが稼働中、ミキサバルブ C(D) が開いている。	88,94 (88,98)
6	ミキサモータが稼働中、重量計 A(B) が 0 でない。	88,30 (88,34)
7	コンベヤモータ C(D) が同時刻に稼働している。	92,96
8	ミキサバルブ C(D) が同時刻に開いている。	94,98
9	SFC が定められた順序に従ってステップが進行していない。	

表 15 自動サイロ運転システムの検証項目

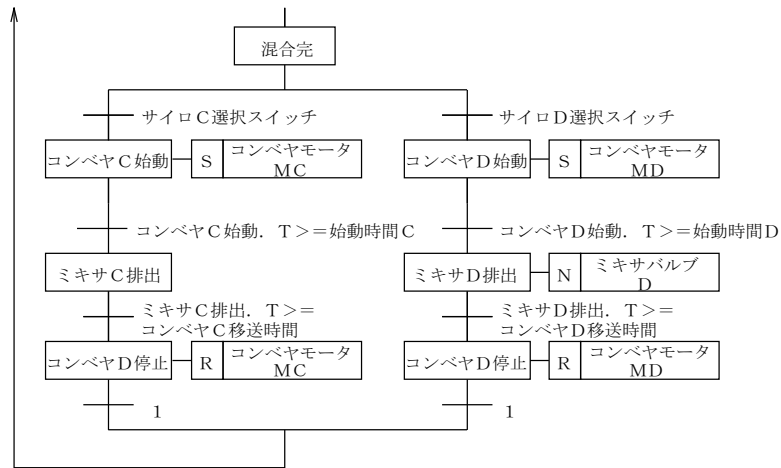
表 15に挙げている項目について解析を行ったところ、表 16に示す結果が得られた。

異常検証 No.	発火 tr-	異常状態ベクトル	発火可能 tr-
9	7	53	dead lock

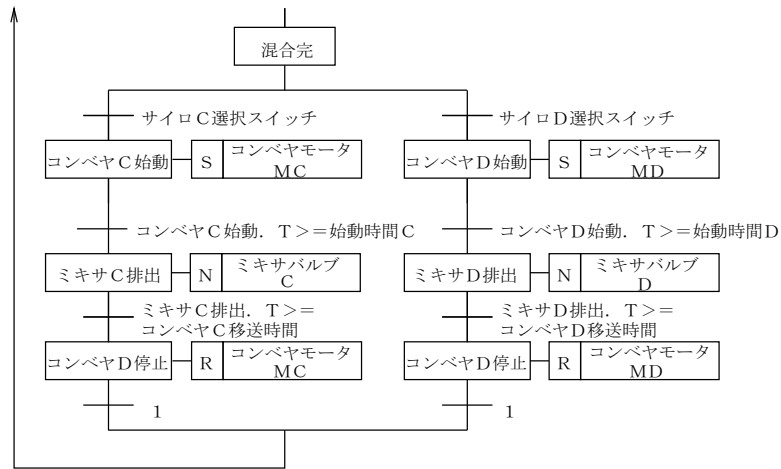
表 16 自動サイロ運転システムの検証結果

概念設計レベルでは、デッドロックが起こることはないように SFC は設計されている。しかし、tr-13 を発火させた後にデッドロックが起きている。これは、『排出完』というステップが行われる前で起きている。これより、それ以前の SFC を含む制御系について誤りがないかを調べる。

修正した SFC を下記に示す (図 56(b))。



(a) 誤



(b) 訂正

図 56 詳細レベルの SFC

この訂正された SFC を含む制御系について解析を先程同様の手順で行う。

訂正された SFC とシステムの構成要素をグラフ表現する (図 57)。

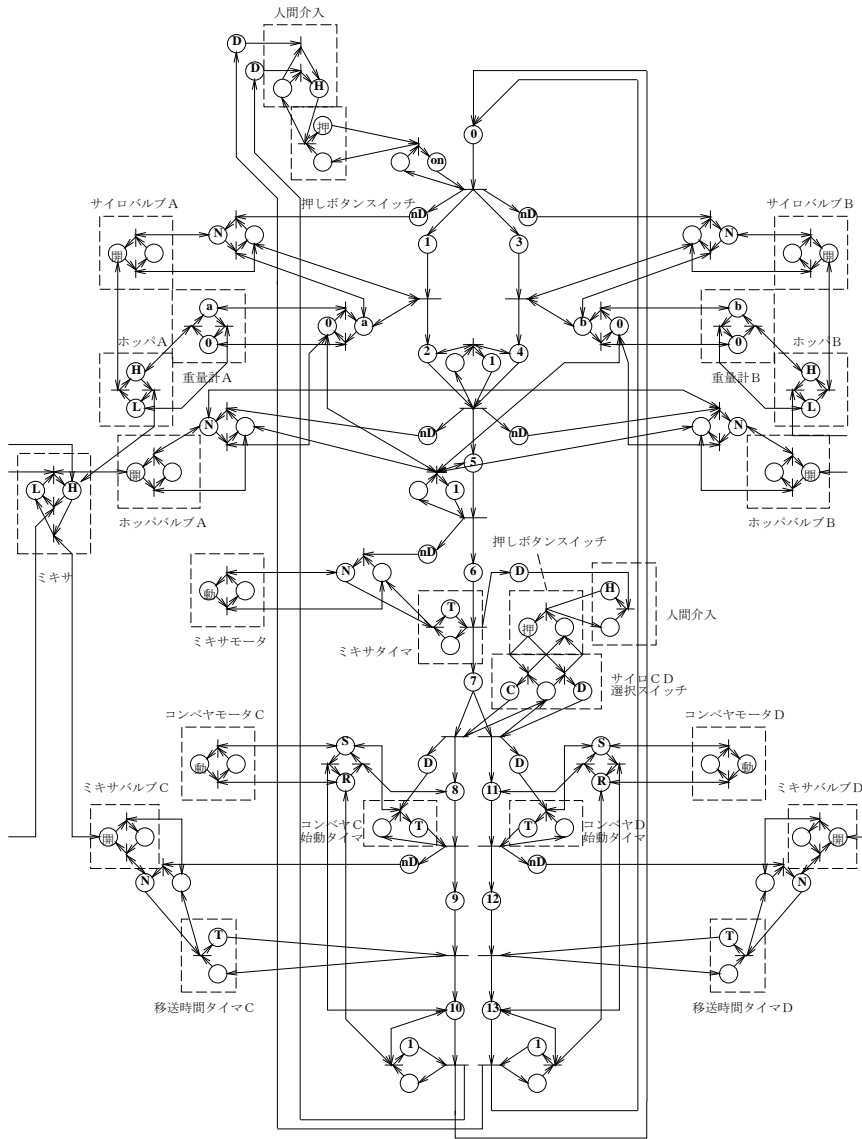


図 57 訂正されたペトリネットグラフ

図 57に対して、プレースおよびトランジションに番号をつける (図 54)。

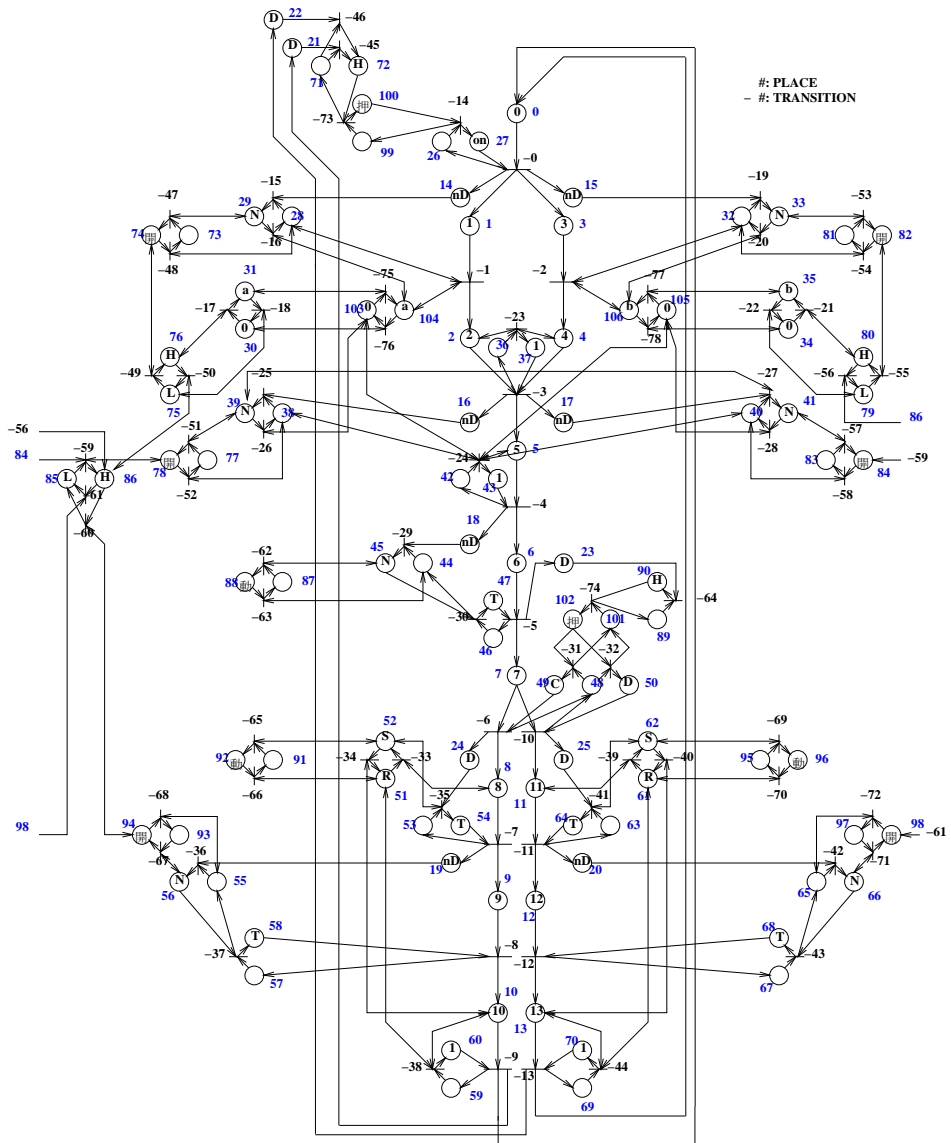


図 58 変換規則によるペトリネットグラフ～番号付け～

作成されたグラフにおいて、遅延要素を加える。加えるトランジションは先程示した表 14と同様である。

### 5. システムの検証

作成されたペトリネットグラフを状態方程式に表し、可達木を求めると、図 59 が得られる。

状態数 = 79

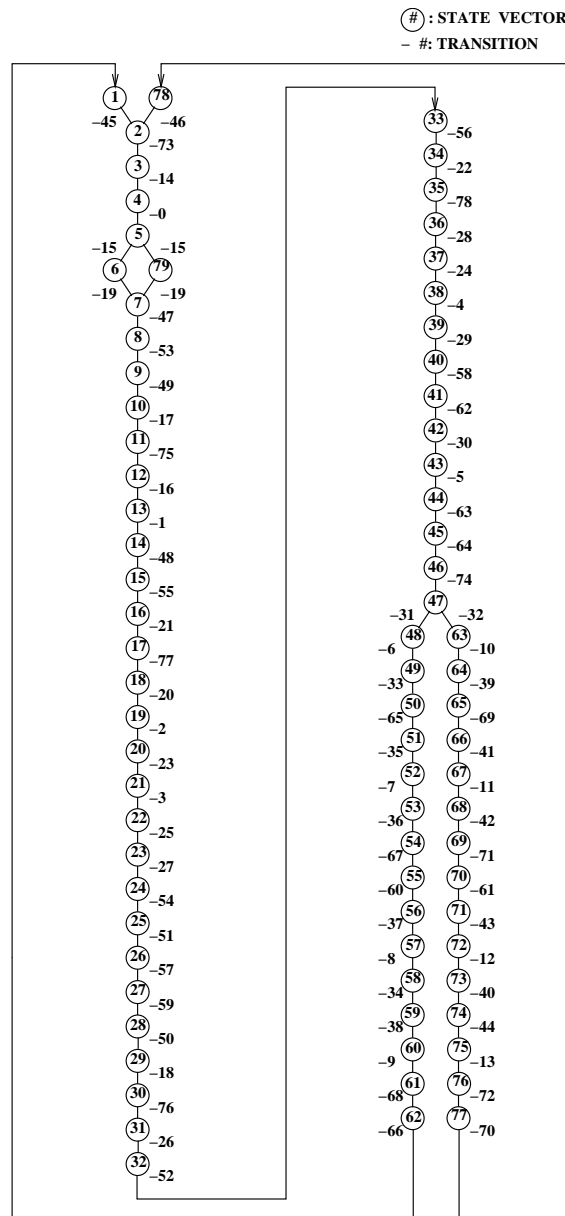


図 59 サイロの可達木

図 59において、システムの検証を行う。検証項目は表 15に示した通りである。

表 15に挙げている項目について解析を行ったところ、表 17に示す結果が得られた。

異常検証 No.	発火 tr-	異常状態ベクトル	発火可能 tr-
-	-	-	-

表 17 自動サイロ運転システムの検証結果

今回修正された SFC を含む制御系には、チェック項目すべてにパスして問題がないことがわかった。これで、SFC を含む制御系の設計を終了する。

## 6. 手法の評価

今回、適用した2つのシステムに関する限り、提案した手法がある程度の有効性があることが確認された。

しかしその反面、下記に挙げた項目について不備が指摘された。

- 可達木のみからの設計ミス箇所特定が困難
- 制御対象、機器を正常時のみ設定
- 制御対象、機器の動作が不明確(簡略化)
- 特別なトランジション条件に対するペトリネットへの変換手法の不備

現段階では解析時の手法が充実しておらず、可達木のみから設計ミス箇所を特定することはできない。また、制御対象・機器の振舞いをペトリネットに変換する際に、正常時のみ設定しており、制御対象・機器の故障や設計ミスに伴う振舞いがペトリネット表現されていないことが挙げられる。異常に対する動作は予測し難く、どのように表現するかが今後の大きな課題といえる。また、制御対象・機器をモデル化する際、どこまで簡略化するかという問題もある。

また、SFCのトランジション条件に関するペトリネット表現に対して、SFCの動作を忠実に反映してるとは言い難いものがある。表18にタイマおよび押しボタンに対するSFC, PNの動作を示す。

	表 現	タイムチャート
S F C		
P N		

表 18 SFC および PN の動作 (1)

表 18のタイムチャートを見ると分かるように、SFC では、条件の成立の後に検出信号  $x$  は元の状態に戻される。しかし、PN では条件の成立と同時に検出信号  $x$  は元の状態に戻されるている。

この解決策として競合したトランジションに優先順位を与えてやるという方法が考えられる (表 19)。



	表 現	タイムチャート
S F C		
P N		

表 19 SFC および PN の動作 (2)

つまり、トランジション (tr-)a の条件成立後、ステップ B に移行したら他にいくつかの発火可能トランジションが存在しても、最初に tr-x を発火させ元の状態に戻してやるという手法である。しかしこれでは、ペトリネット特有の視覚的に理解しやすいという特徴が損なわれてしまう。

今回、できるだけ SFC から正規の (時間) ペトリネットで表現するという方針から、敢えてこの視覚的な特徴が損なわれてしまう解決策を取ることを避けた。

## 7. 結論

本研究では、SFC 設計時のミスを発見するための手法を提案した。本論文の中で、提案した項目を下記に挙げた。

- SFC からペトリネットへの変換手法
- 従来の時間ペトリネットに新たな規則を追加
- 人間をシステムの一部とし、制御系を構成
- 機器と信号を分離

シミュレーションを行なうためのモデルの一つである時間ペトリネットに着目し、SFC からペトリネットに変換するための手法を提案し、実際に2つのシステムに提案した手法を適用した。また、実際の物理現象を反映するために従来の時間ペトリネットに新たな規則を追加した。

制御系の設計に関しては、人間をシステムの一部として制御系を構成した。設計ミスの箇所を特定しやすいように、機器と信号を分離した。

以下に今後の課題について挙げる。

- グラフィック系の SFC より自動変換
- 初期状態のトークンの配分法
- 設計ミスの箇所の特定法
- 制御対象、機器のあらゆる状況下を設定
- 制御対象、機器の簡略化の度合
- 故障時の SFC の設計法

今後の課題として、グラフィック系のSFCより自動変換する手法の開発が挙げられる。また、本研究では問題にとりあげてないが、初期状態のトークンの配分により状態の遷移が全く異なってくることから、トークンの配分に関する課題をここでは挙げている。解析時においても、手法が充実しておらず、可達木のみから設計ミスの箇所を特定することはできない。また、制御対象・機器の振舞いをペトリネットに変換する際に、正常時のみ設定しており、制御対象・機器の故障や設計ミスに伴う振舞いがペトリネット表現されていないことが挙げられる。異常に対する動作は予測し難く、どのように表現するかが今後の大きな課題といえる。また、制御対象・機器をどこまで簡略化するかという問題もある。最後に、設計ミスの箇所の特定や作成された、正常時のSFCから故障時のSFCを設計することなどを今後の課題とされる。

## 謝辞

研究室全般にわたり、常に的確な御指導、御助言を賜りました西谷紘一教授に深く感謝致します。本研究に関し、終始有益な御助言を下さいました藤原健史助教授、黒岡武俊助手、北島禎二助手に心より感謝致します。本論文をまとめるにあたり、適切な御指導、御助言を承りました福嶋 雅夫教授に深く感謝致します。最後に、あらゆる面で大変お世話になりました西谷研究室の皆様に厚く御礼申し上げます。

## 参考文献

- [1] 関口隆 「シーケンス制御工学」．電気学会,pp.124～207,1992
- [2] 青木正夫 「プログラマブルコントローラの新しいプログラミングテクニック  
シーケンシャルファンクションチャート (SFC) 入門」．近代図書,pp.15～88,1992.
- [3] 離散事象システム研究専門委員会編 「創立30周年記念 ペトリネットとその  
応用」．計測自動制御学会,pp.1～209,1994.
- [4] 熊谷貞俊, 薦田憲久 「ペトリネットによる離散事象システム論」．コロナ  
社,pp.103～124,1995.
- [5] 関口隆 「第2章 シーケンス制御」．電学誌,pp.653～656,1993.
- [6] プラント事業部 「パイプレスプラントのシミュレーション」．三井造船技  
法,pp.34～44,1991.
- [7] 関口隆, 「シーケンス制御系におけるペトリネット自動生成 ES」．電学論  
D,pp.349～356,1991.
- [8] 奥川峻史, 「ペトリネットの基礎」．共立出版,pp.2～70,1995.
- [9] 村田忠夫, 「ペトリネットの解析と応用」．近代科学社,pp.1～11,pp.28～  
141,1995.
- [10] 椎塚久雄, 「ペトリネットの応用とそのツールとしての有用性」．電学論  
C,pp.855～861,1994.
- [11] 青山幹雄, 「ペトリネットによるシステムの開発の方法」．電学論 C,pp.862  
～868,1994.