

TM-0423

T-invariantによる
通信システム用サービスの解析

上田佳寛, 柴田健次, 田中 亘, 長谷川晴朗

December, 1987

©1987, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456 3191-5
Telex ICOT J32964

Institute for New Generation Computer Technology

T-invariantによる 通信システム用サービスの解析

Analysis of Services in a Communication System
by means of T_invariant

上田佳寛

柴田健次

田中 亘

長谷川晴朗

Yoshihiro UEDA Kenji SHIBATA Wataru TANAKA Haruo HASEGAWA

沖電気工業株式会社

Oki Electric Industry Co., Ltd.

1. はじめに

我々は、通信システムにおいて曖昧なユーザ要求から全体として整合の取れた設計仕様を自動的に作成する仕様設計エキスパートシステムEXPRESS (EXPeRt system for ESS)を構築中である。本システムでは、拡張ペトリネットにより設計仕様を記述し、同時にペトリネットを利用した代数的解析手法を利用して設計仕様の論理検証を行っている。ここでは、システムの概要を簡単に述べた後でペトリネットの関数行列を利用した解析について報告する。

2. 仕様設計エキスパートシステムの概要(EXPRESS)

2.1 システム構成

図1に全体のシステム構成概念図を示す。本システムは、仕様獲得サブシステム、知識ベースおよび仕様検証サブシステムの3つから構成される。

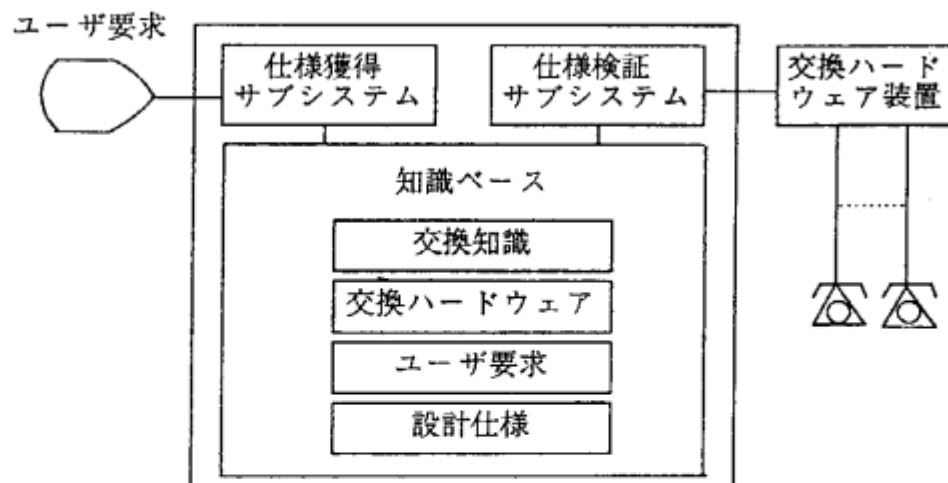


図1 システム構成概念図

(1)仕様獲得サブシステム

サービス要求定義者が自然言語により交換サービスに関する要求仕様を入力すると、知識ベースに基づいて要求理解を行う。また、知識ベースと矛盾するサービスが入力された時は、ユーザとインタラクションを取ることで矛盾を解消する。

(2) 知識ベース

設計仕様および設計仕様を作成するために必要な知識が蓄積されている。即ち、ここには交換処理、交換ハードウェア、ユーザ要求、設計仕様等に関する情報が、知識ベース化されている。

(3) 知識検証サブシステム

知識ベースに蓄積された交換サービス仕様にしたがって、外部交換ハードウェア装置を制御し、ユーザ要求が正確に理解されているか否かの検証を行う。

2.2 設計仕様作成過程

本システムにおいて、ユーザ要求から相互に矛盾のない仕様を作成する過程を、図2に示す。

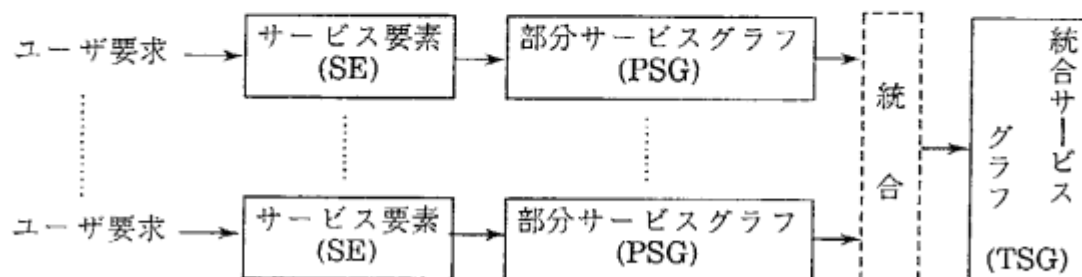


図2 設計仕様の作成過程

ユーザ要求から設計仕様を決定する過程は、ユーザの要求を表すサービス表現から設計仕様を表すサービス表現への変換として捉えることができる。以下にそれぞれのサービス表現について述べる。

(1) ユーザ要求

ユーザ要求は、1サービスの開始から終了までを、時間経過に従い動作とその結果変化するリソースの状態とについて、自然言語を用いて記述したものである。

(2) サービス要素

サービス要素(Service Element,以下SEと略す)は、ユーザ要求中の動作及び状態を表す文を格構造に変換し、システムとしての表現を統一するものである。SEの一節は、ある一つの動作とその動作の結果変化する状態を表すいくつかの文を変換したもので、一サービスについてその動作の数だけの節が作成される。

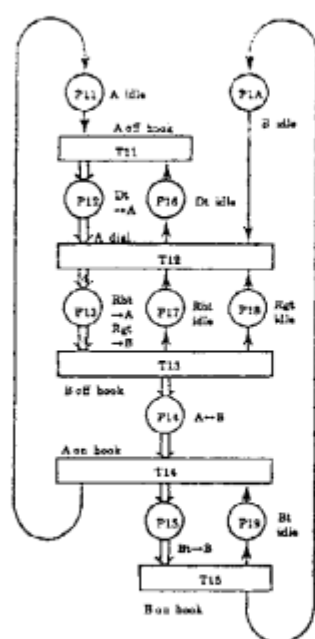
(3) 部分サービスグラフ

部分サービスグラフ(Partial Service Graph,以下PSGと略す)は、SEをベトリネットに対応させたものである。SEの動作をトランジションとし、対象毎に変化前の状態と変化後の状態をそれぞれ入力プレースの集合、出力プレースの集合として表す。ここで、ベトリネットを使用したことにより、単に状態を表現するだけでなく、トランジションを発火後トークンを移動させて状態の推移を明確にできる利点がある。ここでは、強保存のベトリネットを使用している。

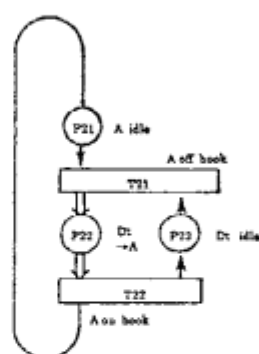
(4) 統合サービスグラフ

統合サービスグラフ(Total Service Graph,以下TSGと略す)は、複数の断片的なPSGを統合して作成した求める最終の設計仕様である。表現形式は、ベトリネットとの対応を明確にしたものとしており、単一の状態に一つのプレースを対応させ、かつ、一つのプレースに接続し得るトランジションを明確にしている。ここでも、強保存のベトリネットを使用している。

図3に内線相互接続サービスの発呼者先掛、ダイヤル音聴取中切断及びホットライン接続呼出中切断の3つのサービスを統合したTSGを示す。統合方法は、一つのPSGをTSGとパターンマッチングさせ、TSGの中に存在しないものがあれば、これをTSGに付加することにより行う。

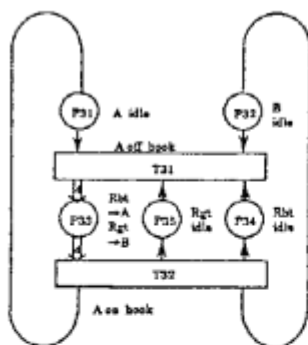


(a) PSG1 : 内線相互発着先提

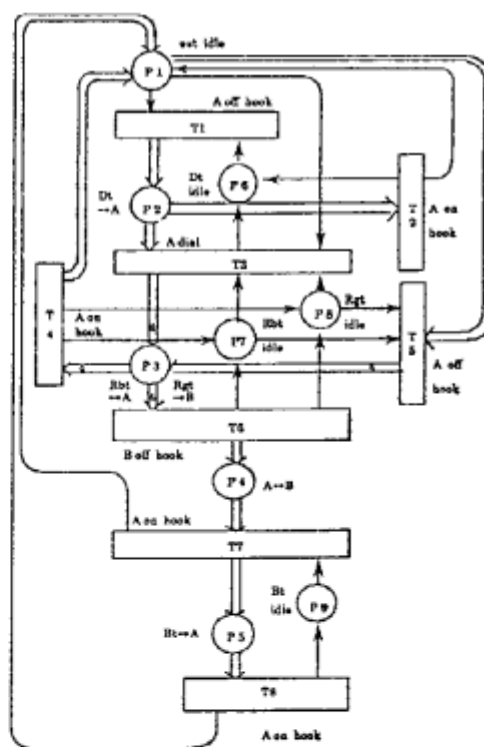


(b) PSG2 : 内線相互ダイヤル音
聴取中切断

INTEGRATION



(c) PSG3 : ホットライン呼出中切断



(d) TSG

図3 PSG及びTSGの表現例

3. ペトリネットの関数行列

ここで簡単に、ペトリネットの行列について述べる。トランジションの入力関数及び出力関数を表す行列 D^-, D^+ を定義する。各行列は、トランジションに対応する m 個の行とプレースに対応する n 個の列から成る。 D^- の j 行 i 列の要素は P_i から T_j に入るアークの多重度を、また D^+ の j 行 i 列の要素は T_j から P_i に出ていくアークの多重度を示す。各プレースに存在するトークンの数を、 n 次元ベクトル μ (マーキングと呼ぶ) で表すと、(a) 式が成立する時 T_j が発火する。

$$\mu \geq e[j] \cdot D \quad \dots\dots(a)$$

但し、 $e[j]$ は第 j 成分が 1 でそれ以外の成分が 0 の m 次元ベクトルである。

また、マーキング μ_0 において、 T_j が発火すると、新しいマーキングは次のようになる。

$$\begin{aligned} \mu &= \mu_0 + e[j] \cdot D \quad \dots\dots(b) \\ D &= D^+ - D^- \end{aligned}$$

従って、マーキング μ_0 において発火列

$$\sigma = T_{j_1}, T_{j_2}, \dots, T_{j_k}$$

が発火すると、新しいマーキングは次のようになる。

$$\begin{aligned} \mu &= \mu_0 + (e[j_1] + \dots + e[j_k]) \cdot D \\ &= \mu_0 + f(\sigma) \cdot D \quad \dots\dots(c) \end{aligned}$$

以上のことを利用すると、あるベクトル $f(\sigma)$ の発火順序を帰納的に、求めることができる。

4. 設計仕様におけるサービスの定義

設計仕様におけるサービスは発火可能なトランジションの列で表現されている。以下に、その発火列に対応するベクトル(発火可能ベクトル)の定義を示し、次にサービスの階層的定義を示す。

4.1 発火可能ベクトルの定義

$f(\sigma)$ が次の 3 項目を充たす時、これを発火可能ベクトルと定義する。

- (1) $f(\sigma)$ のすべての成分は、非負整数である。
- (2) $f(\sigma)$ は、ペトリネット行列のカーネルベクトルである。

$$f(\sigma) \cdot D = 0$$

- (3) $f(\sigma)$ の示すトランジションの発火列において少なくとも一つは、すべて (a) 式を充たす発火列を持つ。

項目 (1), (2) は、 $f(\sigma)$ が T-invariant であることを表しており、項目 (3) では、 $f(\sigma)$ が擬似解(トランジションの可能な発火列に対応しない解)ではないことを意味している。

このとき、発火可能ベクトルとなる $f(\sigma)$ は、マーキング μ が μ 自身から可達となるときの解になる。

4.2 サービスの階層的定義

設計仕様におけるサービスには、サービス(service)、素サービス(primary service)、隠れたサービス(hidden service)がある。ここでは、これらを以下のように定義する。

① サービス(service)

サービスとは、各リソースがアイドル状態である初期マーキングから最終的に初期マーキングに戻る発火可能ベクトルで示される発火列である。

② 素サービス(primary service)

ペトリネット行列における発火可能ベクトルの数は、一次従属なものまで考えると、無限である。そのためサービスの数も無限になると考えられる。このことは、多くのサービスが、他のどのようなサービスの和でも表せない独立したサービスの和で構成されていることを意味する。素サービスとは、他のどのようなサービスの和でも表せない独立したサービスのことである。ただし、サービスは発火列、発火可能ベクトルは T-invariant で表されるために一次従属な

発火可能ベクトルの中にも素サービスに対応するものがある。その例を6章に示す。このような場合のために、トランジションの発火回数の条件を考慮することでサービス数の制限を行った。付録1に一次独立な発火可能ベクトルを求めた例を示し、付録2にトランジションの発火回数の条件の考察を示す。

③隠れたサービス(hidden service)

素サービスには、次の3つがある。

- I) ユーザ要求が素サービスとなる。
 - II) ユーザ要求の一部が素サービスとなる。
 - III) ユーザ要求にないサービスが素サービスとなる。
- III)のサービスは統合過程で作成されるサービスである。このようなサービスを隠れたサービスという。

5. 素サービスの発見

次の2条件は、発火可能ベクトルから素サービスを求めるためのものである。

(i)マーキング μ にトークンが存在しているプレース以外のプレースにおいて、そのプレースに含まれるトークンが、常にそのプレースに対するトランジションの入力アーク数をこえてはならない。

(ii)移動する任意の2つのトークンが、ある時点において直接的あるいは間接的に関係しなければならない。

◎直接的とは、複数のトークンが、あるトランジションにより同時に発火し、同じプレースに同時に含まれる時のトークン間関係のことをいう。

◎間接的とは、直接的なトークンを辿ることにより、2つのトークンが関係づけられ、その2つのトークンが直接的な関係になっていない場合のことをいう。例えば、aとbが直接的関係、bとcが直接的関係であり、aとcが直接的関係でなければ、aとcを間接的関係という。

次に、(i),(ii)の2つの条件について簡単に説明する。

5.1 プレースに存在するトークン数の条件

(i)の条件は、プレースに存在するトークン数を制限することにより、発火可能ベクトルの一次従属なベクトルを素サービスの発火可能ベクトルとするときのものである。ペトリネットグラフのモデルで例を示す。(図4)

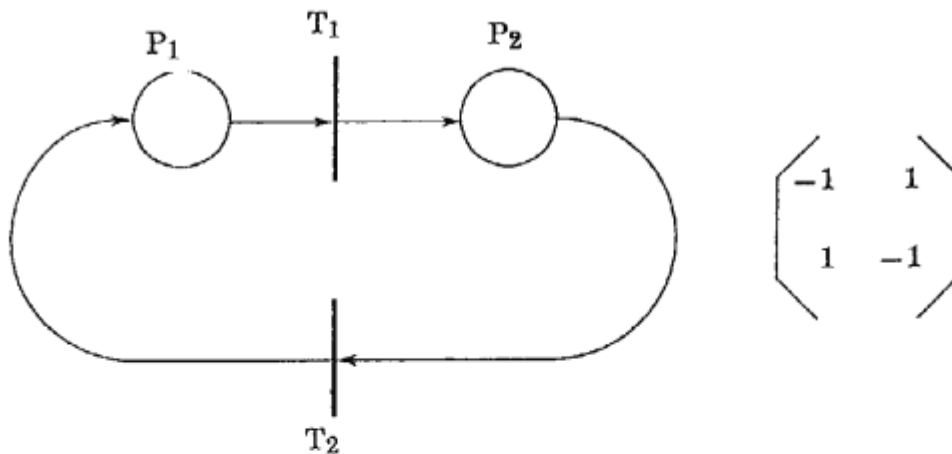


図4 条件(i)により制約を受ける例

この行列のカーネルベクトルは (n,n) (但し n は非負整数)となる。初期マーキングを $\mu=(2,0)$ とおくと、 T_1 発火 $\rightarrow T_2$ 発火よりマーキングは元に戻り $(1,1)$ は発火可能ベクトルとなる。次に、ベクトル $(2,2)$ を考えた場合において、 $T_1 \rightarrow T_1 \rightarrow T_2 \rightarrow T_2$ と発火することは可能である。このことは、 T_1 を発火し、その後 T_2 を発火するということ二度繰り返しているに過ぎない。このような場合は、 P_2 に存在するトークンの最大数を入力アーク数とすることにより、素サービスによる発火可能ベクトルを一意に決めることができる。

5.2 ペトリネットにおけるトークン間関係

(ii)の条件は(i)と同様に、素サービスの発火可能ベクトルを求めるために制約を行うものである。一例を図5に示す。

この行列におけるカーネルベクトルは (n,n,n) (但し n は非負整数)となる。初期マーキングを $\mu=(2,0,0)$ とおくと、 $T_1 \rightarrow T_2 \rightarrow T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_3$ 発火よりマーキングは元に戻りベクトル $(2,2,2)$ は発火可能ベクトルとなる。また、(i)の条件も充たすが、この場合も $T_1 \rightarrow T_2 \rightarrow T_3$ と発火することを二度繰り返すことに過ぎない。このような場合については、移動する2つのトークンの関係を検証することにより、一意に決めることができる。

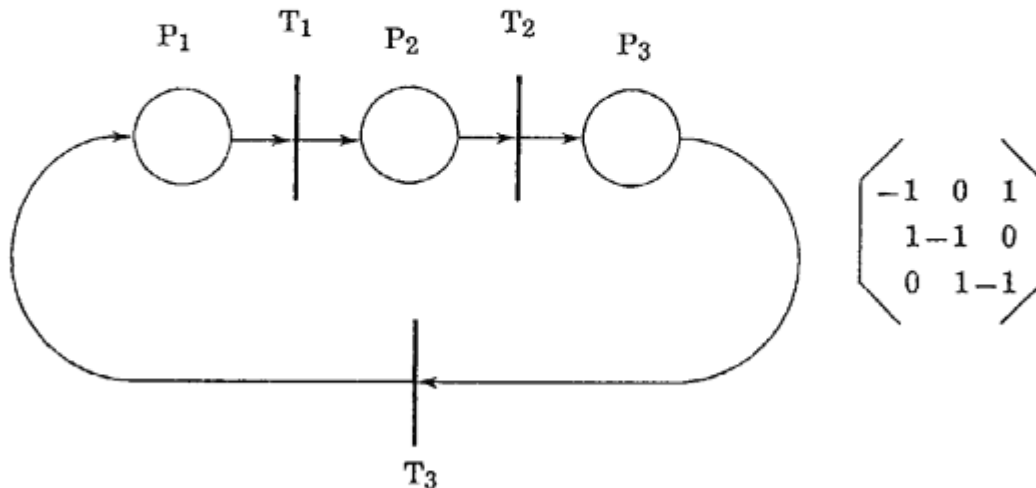


図5 条件(ii)により制約を受ける例

6. 統合後のTSGから隠れたサービスが求まる例

実際に、図3のTSGにより隠れたサービスを求めてみる。まず、図3のTSGから一次独立なカーネルベクトルの数を調べる。

$$m - \text{rank}(D) = 4 \quad m \text{は行列} D \text{の行数}$$

よって、一次独立なカーネルベクトルの数は4である。また、図3で統合する前の3つのサービスのベクトルは、以下の3ベクトルである。

$$\begin{aligned} p &= (1, 0, 1, 0, 0, 1, 1, 1) \\ q &= (0, 0, 0, 1, 1, 0, 0, 0) \\ r &= (1, 1, 0, 0, 0, 0, 0, 0) \end{aligned}$$

この3つのベクトルは一次独立なベクトルである。その他の一次独立なカーネルベクトルは、以下のものである。

$$s = (1, 0, 1, 1, 0, 0, 0, 0)$$

それについて発火順序を求めると、 $T_1 \rightarrow T_3 \rightarrow T_4$ のルートが成り立つことがわかる。またベクトル s は、条件(i)、(ii)も充たす。よってベクトル s により示される発火列は素サービスとなり、その発火列はユーザ要求にないサービスであるので、隠れたサービスといえる。次にベクトル p, q, r, s による一次従属なベクトルの中に、隠れたサービスがあるかどうかを調べる。任意の発火可能ベクトルは次のようになる。

$$\begin{aligned} f(o) &= \alpha p + \beta q + \gamma r + \delta s \\ \alpha, \beta, \gamma, \delta &\text{は整数、ただし} f(o) \text{の要素は非負} \end{aligned}$$

この $f(o)$ の中で、4章の項目(3)を充たし、かつ条件(i)、(ii)を充たすベクトルは、 p, q, r, s 及び

$$t = (0, 0, 0, 0, 1, 1, 1, 1)$$

の5つである。よってTSGのペトリネットによる行列から、ベクトル p, q, r 以外の素サービスによる発火可能ベクトル s, t を求めることができる。このことは、 p, q, r を統合することにより、隠れたサービス s, t が作成されることを意味している。

7. おわりに

ペトリネットを用いて表現した設計仕様に、関数行列を使用した解析を行って隠れたサービスの発見に適用した。今後は、色つきトークンの導入されたシステムにも関数行列を使用した解析の可能性を、調べてみるつもりである。

なお、本研究は第5世代コンピュータプロジェクトの一環として行っているものである。

付録1

図6のペトリネットより発火可能ベクトルを求める一例を示す。

図に対応する行列を D , 求めるベクトルを (a, b, c, d, e, f) とおく。発火可能ベクトルの定義により

$$(a, b, c, d, e, f)D = 0$$

a, b, c, d, e, f は非負整数

を充たすことが必要である。

$$(a, b, c, d, e, f)D = 0 \text{ より}$$

$$-a-d+e+f = 0 \quad \text{---①}$$

$$a-b = 0 \quad \text{---②}$$

$$b-c+d-e = 0 \quad \text{---③}$$

$$c-f = 0 \quad \text{---④}$$

②, ④より

$$a=b, c=f$$

①, ③に代入すると

$$a-c+d-e=0$$

I) $a=0$ と置くくと $b=0$

i) $c=0$ と置く、 $f=0$

$$d=e, \text{ よってベクトル } (0, 0, 0, d, d, 0)$$

ただし $d \neq 0$ ---⑤

ii) $c \neq 0$ と置く、 $f=c$

イ) $d=0$ と置く

$$e=-c, \text{ よってベクトル } (0, 0, c, 0, -c, c)$$

ただし $c \neq 0$ ---⑥

ロ) $d \neq 0$ と置く

$$e=-c+d, \text{ よってベクトル } (0, 0, c, d, d-c, c)$$

これは⑤, ⑥の一次結合

II) $a \neq 0$ と置くくと、 $b=a$

i) $c=0$ と置く、 $f=0$

イ) $d=0$ と置く

$$e=a, \text{ よってベクトル } (a, a, 0, 0, a, 0)$$

ただし $a \neq 0$ ---⑦

ロ) $d \neq 0$ と置く

$$e=a+d, \text{ ベクトル } (a, a, 0, d, a+d, 0)$$

これは⑤, ⑦の一次結合

ii) $c \neq 0$ と置く $f=c$

$$\text{ベクトル } (a, a, c, d, a-c+d, c)$$

これは⑤, ⑥, ⑦の一次結合

よって一次独立なカーネルベクトル

$$(0, 0, 0, d, d, 0)$$

$$(0, 0, c, 0, -c, c)$$

$$(a, a, 0, 0, a, 0)$$

の3ベクトルが得られる。ただし a, c, d は0以外

次にすべてのベクトルが正の整数となることより、

$$(0, 0, 0, 1, 1, 0)$$

$$(1, 1, 0, 0, 1, 0)$$

$$(0, 0, 1, 1, 0, 1)$$

の一次独立な発火可能ベクトルを得る。

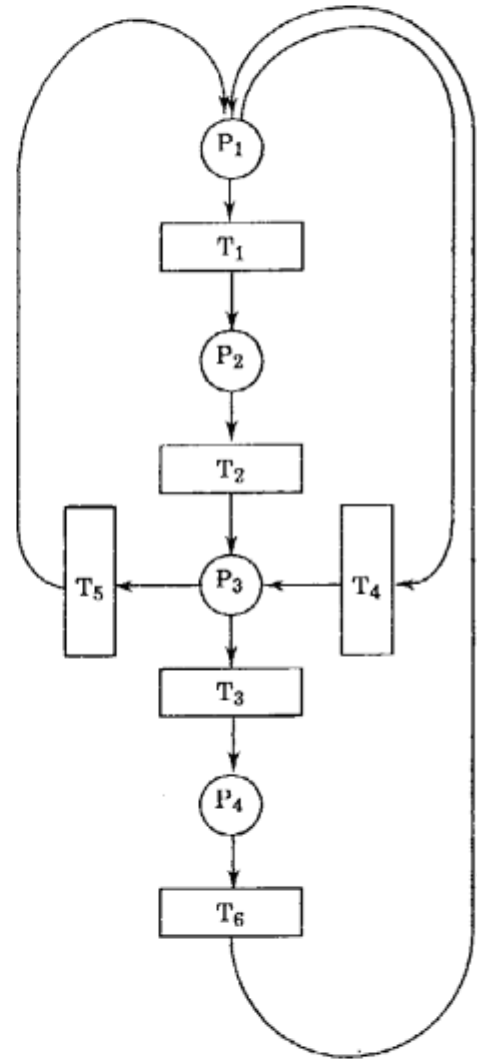


図6 付録

付録2

素サービスとなる発火可能ベクトルによるトランジションの発火回数について、以下の3つの補題を考えた。解については補題一のみしか解けていない。

補題一

サービスを形成しないループをもたないペトリネットグラフにおいて、トークンが一個のみで素サービスを形成するならば、素サービスにおいて、あるトランジションの発火回数は最大1回だけである。

サービスを形成しないループ……各リソースがアイドル状態を示すプレースのみに存在することはないが、あるマーキングから、そのマーキングに戻る発火列のこと。このループは、発火可能ベクトルに対応している。

補題二

サービスを形成しないループをもたないペトリネットグラフにおいて、トークンが n 個で素サービスを形成するとすれば、素サービスにおいて、任意のトランジションの発火回数は最大 n 回である。

補題三

サービスを形成しないループを持つペトリネットの場合、トークンが n 個で素サービスを形成するとすれば、素サービスにおいての、サービスを形成しないループ内のトランジションの発火回数は最大 $(\text{ループの数} + 1) \times n$ である。

解補題一

トークンが一個のみで素サービスを形成しているということは、トランジションに対する入力アーク数と出力アーク数はそれぞれ1である。また、そのトランジションに対する入力プレースと出力アークは一意である。(強保存性)……①

次に、素サービスにおいては、トークンがアイドル示すプレース以外のプレースを2度通ることはない。つまり、アイドル以外のマーキングから、そのマーキングに戻ることはない。そのことは、素サービスの定義と、サービスを形成しないループがないことから明らかである。……②

①より任意のトランジションに対する入力プレースは、ただ一つ決る。また②よりその入力プレースをトークンが通るのは、素サービスにおいては、ただ一回だけであるので、そのプレースを入力プレースとするトランジションの発火回数はただ一回だけである。

このことは、任意のトランジションについていえる。よって補題一が成り立つ。

参考文献

- [1]青柳廣,長谷川晴朗,田中亘,柴田健次:“通信システムにおける仕様設計エキスパートシステムの一検討”,電子通信学会,交換研究会,SE86-10,(1986)
- [2]J.L.Peterson:“Petri Net Theory and Modeling of Systems”,Prentice-Hall,(1981)
- [3]長谷川晴朗,田中亘,柴田健次,山口政己:“ペトリネットを利用した通信システム設計仕様の解析”,電子通信学会,第2回ネット理論研究会,pp.75-84(1986)
- [4]H.Hasegawa,W.Tanaka and K.Shibata:“Analysis of Design Specification in a Communication System by means of Petri Nets”,Proceeding of 11th Computer Software and Applications Conference,pp.701~706,(1987)
- [5]翁長健治,葛崎偉:“ペトリネットのT-invarianceの構造解析”,電子情報通信学会論文誌,Vol.J70-A,No.2,pp.185~194(1987)