知識工学と人口知能　68－8
（１９９０．１．１９）

ペトリネット可到達問題に対する線形整数計画法と時制論理の適用

伊藤　潔　（上智大学　理工学部　一般科学研究室　情報科学部門）
本位田真一　（(株)東芝　システム・ソフトウェア技術研究所）
内平直志　（(株)東芝　システム・ソフトウェア技術研究所）
志田圭介　（上智大学大学院　理工学研究科　機械工学専攻　博士前期課程）

ペトリネットの可到達問題に対して発火系列に焦点を絞って線形整数計画法および時制論理を適用する．ペトリネットに対応するベクトル加算系に，可到達木内のベクトルの非負性，およびベクトル間の遷移の繰返し性を導入する．これをさらに線形整数計画法に変換して，発火系列の部分集合を求める．デッドロックなどの状態に陥る発火系列が求まると，それを回避するための矛盾のない回避策の記述に時制論理を用いる．発火系列の検出と回避をこのように順序立てて行うことは有効である．

A Combined Approach of Linear Integer Programming and Temporal Logic to the Reachability Problem in Petri Nets

Kiyoshi Itoh*, Shinichi Honiden**, Naoshi Uchihira**, Keisuke Shida***

*: Laboratory of Information Science, Natural Sciences Center,
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan
**: Systems and Software Engineering Laboratory, Toshiba Corporation
Yanagi-cho 70, Saiwai-ku, Kawasaki 210, Japan
***: Department of Mechanical Engineering,
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan

This paper describes a combined approach of linear integer programming and temporal logic to the reachability problem in Petri Net. Application example is the detection and avoidance of deadlock state. The deadlock avoidance is not easy to plan before detecting of firing sequences from the initial state to the deadlock state. Linear integer programming is constructed in the way the non-negative property of state vectors in the reachability tree and the repeatable property of state vector transition are incorporated into the vector addition system of Petri Net. Linear integer programming is used to obtain the feasible and optimum solutions which indicate the firing sequences from the initial state to the deadlock state. It does not cover the whole set of firing sequences, but it can produce appropriate subset of sequences. By the subset, Petri Net analyst can obtain the useful property of the objective system. Temporal logic description is used to control the firing sequence in order to avoid the deadlock state.

1. はじめに

ペトリネットの「可到達問題」は，ある初期状態をもつペトリネットが別の指定された状態に遷移可能かどうかを解く問題である．従来，この可到達問題の解法として，ペトリネットに対応するベクトル加算系による方法と可到達木による方法がある．

ペトリネットに対応するベクトル加算系による方法では，ペトリネットが有界でない時でも有限アルゴリズムで可到達かどうかの判定ができることが証明されている([Kos82][Lam88][May84])．その計算量は指数時間オーダである．しかし，その際の発火系列を検出する手順は明確にはアルゴリズム化されていない．従来のベクトル加算系による実用上の解析範囲は，発火回数の検出[Kar69]にとどまっていると考えられる．この理由は，ベクトル加算系では系列に関する情報が極めて少ないことである．

可到達木は，ペトリネットが有界の時は発火系列を求めるための有益な情報を提供する．しかし，有界でない時は，ωが導入されるため，発火系列を求めることのみならず可到達かどうかの判定もできない．

ペトリネットの解析者に有効な情報は，ある特定状態(たとえばデッドロック状態)に到達するかどうかの判定情報と，その状態に到達するための発火系列の情報である．この情報によりペトリネットの解析者は，その状態に到達しないための回避策を作り出すことができる．この情報の計算機援用による収集は有効である．なぜなら，ペトリネットを静的に見ながら適切な回避策を人手で得る作業は，かなり困難な場合が多い．

本稿では，可到達木を構成する際に使っている情報を顕在化し，可到達木内のベクトルの非負性，およびベクトル間の遷移の繰返し性を，ペトリネットに対応するベクトル加算系に導入して構成される線形整数計画法によって，簡便に発火系列集合の部分集合を検出する手法を提案する．さらに，必要に応じて，例えば，デッドロックへの発火系列に対する回避策を時制論理で記述する手法を提案する．

発火系列の事前の検出なしに，ペトリネットを静的に見ながら回避策を記述することは，一般に必ずしも容易ではない．本稿で示す通り，発火系列の検出と回避を順序立てて行うと，この作業が効率的になる．

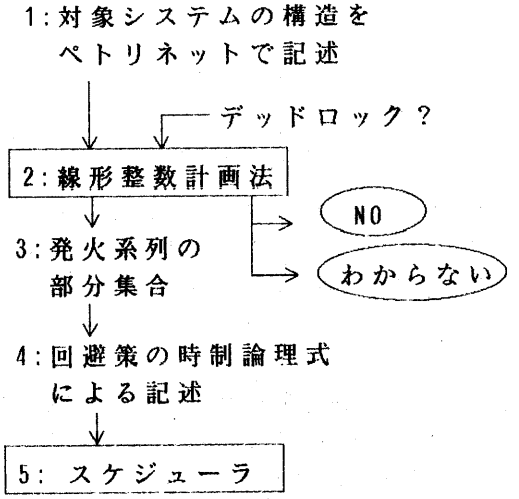


図1　デッドロックの検出と回避

2. 発火系列の検出・回避手順

発火系列の検出とその回避のための手順を述べる．

① 図1のblock1に示す通り，ペトリネットでの記述が必要以上に複雑になることを防ぐため，ペトリネットで記述すべきことを対象システムの構造に関することに限定して，ペトリネット自身の繁雑さを抑制する．あるいは対象システムのプログラムから構造に関する部分を自動抽出し，ペトリネットに変換する．

② ペトリネットに対して発火系列を求める．このために，可到達木を変換して，初期状態から可到達な状態集合を求める問題を作る．この際に，可到達木内のベクトルの非負性，およびベクトル間の遷移の繰返し性を使って，初期状態から可到達な状態系列を，状態遷移式と呼ぶ式の集合に集約し，これらの式の解を求める．本稿では，この式の集合を線形整数計画法(block2)の範囲で求めて，発火系列集合の部分集合(block3)を求める．

③ 必要に応じて，その発火系列を回避するための回避策を検討する．この回避策は，発火可能な複数のトランジションに対して決定的選択を与えること，すなわち，本来ならば非決定的に選択されるトランジョンの発火に対して順序関係を導入することである．この回避策として，人手による新たなプレースやトラン

ジションの挿入が通常行われる．しかし，ペトリネットが複雑になったり誤りをもつようになる場合がある．前もって①のような発火系列の検出を行わずに，ペトリネットのみで回避策を人手で与えることは容易ではない．

④ 発火系列の回避策を時制論理で記述する．ペトリネットの発火系列を制御するスケジューラ(block5)にこの回避策を渡す．

3. ペトリネットの発火系列

3.1 諸定義

以下の定義は[Kar69]，[Hac76]，[Pet84]による．

[定義1] r次元ベクトル加算系(Vector Addition System:VAS)wは，r次元の非負整数の状態ベクトルdと，r次元の整数ベクトルの有限集合Vの対w＝(d,V)である．VAS wの可到達集合R(d,w)は，次の形式を有する．

R(d,w)＝ {d+v1 +v2 +…+vs ｜vi∈V
&d+v1 +v2 +…+vi ≧0 for i =1,2,…s}

[定義2] ペトリネットNは次の4つ組で定義される有向グラフである．N=(Π,Σ,E,M0)．ここで，

(1)Π＝ {r1 ,r2 ,…rp } ：有限個のプレースの集合．p=｜Π｜

(2)Σ＝ {t1 ,t2 ,…tq } ：有限個のトランジションの集合．q=｜Σ｜

(3)E：有限個の有向なアークの集合．各アークはトランジションからプレースへ，またはプレースからトランジションに向かう．

(4)M0 ：初期状態ベクトル，開始時にプレース中にあるトークン数のベクトル．

もし，プレースrからトランジションt(あるいはtからr)へのアークがあれば，rはtの入力プレース(あるいは出力プレース)と呼ばれる．tの発火とは，各入力プレースから，所定のトークンを取り去り，各出力プレースへ所定のトークンを付加することである．

[定義3] VAS とペトリネットの対応づけ．

(1)Σの要素数｜Σ｜＝Vの要素数．

(2)Πの要素数｜Π｜＝ベクトルvi の次元数．

(3)tの発火条件とは，入力プレース中に所定のトークンがある時に限り，そのプレースからトークンを取り去り，出力プレースに所定のトークンを付加すること

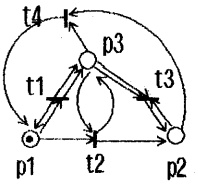


図2 ペトリネット
([Pet84])

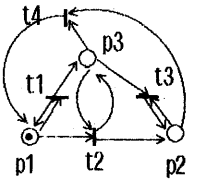


図3 ペトリネット
([Pet84])

である．すなわち，VAS W の可到達集合R(W)に含まれる可到達な状態ベクトルと同じベクトルは，トランジションの発火によって得られる．

[定義4] 可到達問題：非負整数の状態ベクトルd(初期状態)とd'(目標状態)が与えられたとき，d'∈R(d,w)であるかどうかを解く問題．

以下は，本稿で使用する用語の定義である．

[定義5]可到達問題の解が求まる時，dからd'へ遷移するためのトランジションの1つ以上の発火系列を集めたものを発火系列集合と呼ぶ．

[定義6]どのトランジションも発火できない状態をデッドロックと呼ぶ．

3.2 発火回数検出の限界

図2のペトリネットで(1,0,0)が初期状態で，(0,1,0)がデッドロックである．このデッドロックに陥る発火回数の解析は，t1,t2,t3,t4の発火回数をx1,x2,x3,x4とすると，

(0,1,0)=(1,0,0)+(0,0,2)x1+(-1,1,0)x2
+(0,0,-2)x3 +(1,-1,-1)x4，すなわち

(0,1,0)=(1-x2+x4,x2-x4,2x1-2x3-x4)であるから

x2-x4=1，2x1-2x3=x4

を解く．解は，(x1,x2,x3,x4)={(n1,1,n1,0),(n2,3,n2,2),(n3,4,n3,4),…}となる．

この解を満たすt1t2t3は，途中でベクトルの非負性を保つ正しい発火系列である．しかし，解を満たすt1t2t2t2t3は，t1t2t2まで進むと非負性を保たない不正な発火系列である．

以上のように発火回数の検出は，可到達問題の必要条件であり，十分条件ではない．

3．3　可到達木によるペトリネット解析の問題点

図2と図3のペトリネットに対する可到達木を図4に示す．(可到達木の構成手順は例えば文献[Pet84]．)

可到達木による解析の問題点を列挙する．

(A)状態ベクトル内のωは無限の発散を表すが，このωの導入により可到達木の成長を有限に抑制する．しかし，具体的数値で指定された状態ベクトルが可到達木に属するか否かを調べることが困難．

(B)ωがベクトル中に複数個ある時，それらのωの初期値・増分が等しいとは限らない．

(C)構造あるいは初期状態の異なる複数のペトリネットに対して可到達木が等しい場合がある(図2と図3に対して図4)．

以上の問題点により，可到達木による解析は，安全性，有界性，保存性，被覆性の問題への適用にとどまり，可到達問題には適用されていない([Pet84])．

3．4　状態遷移式の構成[Ito81]

線形整数計画問題により可到達問題の解である発火系列集合の部分集合を求める．

ωを有する可到達木は，トランジションの無限な発火を表現している．この可到達木を次の原則で可能な限り有限な線形方程式系に置き換える．

次の(1)～(3)は，可到達木を構成する際に用いた可到達木の性質を顕在化したものである．

(1)可到達木のノードとなる状態ベクトルは非負．

(2)可到達木のルートからのパス上に，上位の状態ベクトルvjと等しい下位のviが存在するとき，vjからviへのトランジションストリングを0回以上繰返してよい．

(3)ルートからのパス上に，上位のvjに対してvi＞vjとなりωを導入したviが存在するとき，vjからviへの発火系列を1回以上繰返してよい．

上の(2)と(3)の繰り返しは本来は非線形の繰り返しである．しかし，次の(4)の原則でこれを線形の範囲にとどめる．

(4)(2)や(3)の繰返しは線形に連接させる．繰返しの中に別の繰返しがある非線形な連接は行わない．

この(4)については3.6節で詳述する．

以上の原則で，次の形式の状態遷移式を構成する．

(state transition exp.)
　:=(left hand of STE) ←(right hand of STE)
(left hand of STE):=(state vector id.)
(right hand of STE)
　:=(term of STE) {｜(term of STE)}$_1$
(term of STE):=(non-negative integer vector)‖
　(state vector id.){(transition string)}$_0$
(transition string)
　:={[(transition id.)]
　　‖(recurrence of transition id.)}$_1$
(recurrence of transition id.)
　:=[{(transition id.)}$_1$]{* ‖ +}$_1$
　　* は0回以上，＋は1回以上の繰返しを示す．

可到達木から状態遷移式集合を求める手順を述べる．

<可到達木から状態遷移式への変換>

[1]可到達木の構成

[2]状態識別名付与

[3]状態遷移式構成

<状態識別名付与>

[1]可到達木のルートからリーフへのパス上において状態ベクトルに対して等しいものには同名，異なるものには異名となるように識別名を付ける．

[2]各々の状態ベクトルVi に対して，Vi ＞Vj であるVj が，Vi からルートノードに至るパス上に存在する時，Vj はVi に被覆されているという関係付けを行う．

<状態遷移式構成>

[1]可到達木の全ノードVi に対して，

「Vi からルートへのパス上で直上のノードからのトランジションストリング を構成する．さらにこのパス上にVi と同名のノードVj があれば(直上のノードも含む)，Vi はVj からの0回以上の繰返しを含む遷移式とする．Vj がVi に被覆されていれば，Vi はVj からの1回以上の繰返しを含む遷移式とする．」(状態遷移基本式)

[2]ルートからリーフへの全てのパスに対して上位から，

「下位レベルの遷移式の右辺の状態名が，上位レベルの左辺の状態名に等しい時，下位の右辺の状態名に上位レベルの右辺を代入して状態遷移式を構成する．」

[3]上で求めた遷移式の内で左辺に同じ状態名があれば，それらの右辺の和をとる．

3．5　発火系列検出手順[Ito81]

状態遷移式はωを含む状態ベクトルも生成できるように繰返し記号を使って構成されている．目標の状態ベクトルが遷移式により導出できるか否かを判定できるように式を変形する．

各々の状態遷移式の一般形は，

(Ct1,…,Cti,…,Ctn)

←(Cs1,…,Csi,…,Csn){(transition string)}ω

である．Csi，Ctiは，トランジションストリング の連接による遷移前後の状態ベクトルの第i番目の要素である．これをさらに吟味すると次の式集合(A)(B)が求まる．

1つのトランジションストリングは，1つ以上のトランジションから構成される．状態ベクトルの各要素の増分は，トランジション毎に固定であるため，トランジションストリング毎にも固定となる．従って次の式集合(A)が求まる．

<式集合(A)>

$$\text{全てのiに対して，} Cti = Csi + \sum_{j=1}^{m} Aij \cdot Xj \ ,$$

mはトランジションストリング の個数，Aijはj番目のトランジションストリング の1回の発火によるi番目の要素の増分(整数)，Xj はj番目のトランジションストリング の発火回数(非負あるいは正)．

上記で，CsiにA1j･XjからAmj･Xjを１つずつ加えた各々は，状態ベクトルの各要素である．これらは，負になってはならない．従って次の式集合(B)が求まる．

<式集合(B)>

ベクトルの各要素が常に非負なる条件として

$$\text{全てのiに対して，} Csi + \sum_{j=1}^{k} Aij \cdot Xj \geq 0$$

(Aik<0，k≦mである全てのkに対して)．

遷移の繰返しを含む場合，初期状態(Cs1，…，Csi，…，Csn)から，ある状態ベクトル(Ct1，…，Cti，…，Ctn)がどの遷移式により生成されるかは，遷移式毎に次の形式の線形整数計画問題を解いて判定される．

$$\text{minimize} \quad Z = \sum_{j=1}^{m} Xj$$

subject to 式集合(A)および(B)．

式集合(A)および(B)を連立したものは，計画問題の制約式と呼ばれ，制約式の解は許容解と呼ばれる．この許容解は複数存在できる．それらのいずれの解も，初期状態から目的状態への発火系列を構成するトランジションストリングの繰返し回数である．この計画問題の最適解は，許容解のうちでこの繰り返し回数が最小な解を表す．

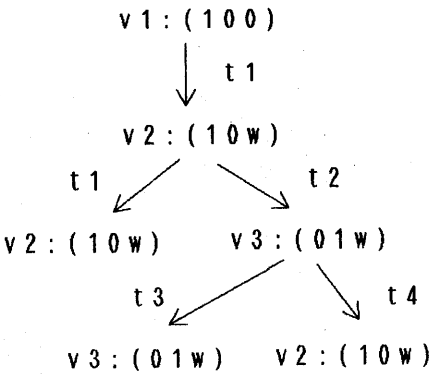
各遷移式に対応する計画問題のいずれかに解が存在すれば，その解は初期状態から目標状態への発火系列集合の部分集合の要素を構成する．どの線形整数計画問題にも解がなければ，線形の範囲では可到達か否かは判定できない．

図2のペトリネットに対して以上の手順で状態名付き可到達木(図4)，状態遷移基本式(図5)，逐次代入(図6)，状態遷移式(図7)を作成する．図7の初期状態からの他のベクトルへの遷移，および図4より初期状態以外のベクトル間の遷移を合わせて，図8を作成する．

3.6　トランジションストリングの連接の範囲

本稿の範囲は，トランジションストリング の連接が有限でかつ線形の場合である．連接が有限であるとは，ある状態遷移式を有限個数のトランジションストリング を用いて表現できる場合である．連接が線形であるとは，あるトランジションの繰り返しの中でさらに繰り返しのない場合である．以上を計画問題に定式化した時，線形整数計画問題になる．

たとえば，図7のv2←v1[t1]⁺ [t2t4]* というトランジションストリング の連接は有限でかつ線形である．この連接の後に，図8のv2に着目して[t1]*や[t2t4]*をさらに付加してできる連接も線形である．しかし，こ



v2 > v1

図4　状態名付き可到達木

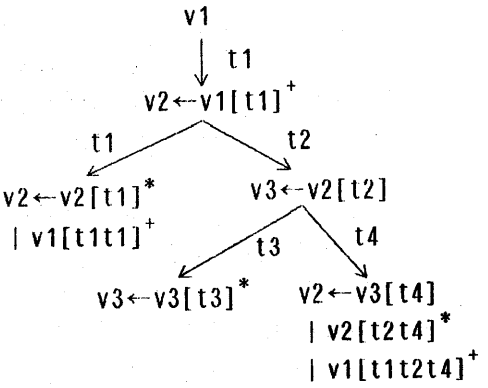


図5　状態遷移基本式

れらは次のように任意個付加できるため，連接は有限ではない．

v2←
v1[t1]+[t2t4]*…[t1]+ [t2t4]*…[t1]+ [t2t4]*

本稿では，図7の範囲にとどめて連接を有限とする．

さらに，v2からv3へも考慮して[t2[t3]*t4]* という付加も考えられる．しかし，これは，内側のt3をx回，外側をy回繰り返す非線形な連接である．本稿では，これも対象外とする．

4. 発火系列集合の部分集合を求める例題

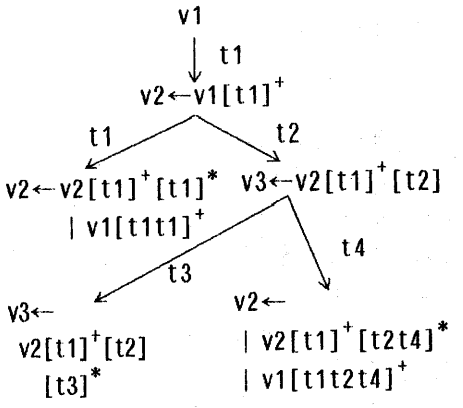


図6　逐次代入

v2←v1[t1]+[t2t4]* | v1[t1t2t4]+
v3←v1[t1]+[t2][t3]*

図7　状態遷移式

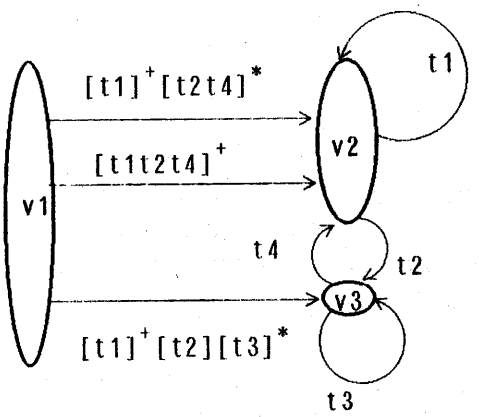


図8　ベクトル間遷移

4.1 デッドロックを起こすペトリネット

図2のペトリネットの状態遷移式(図7)から3章の手順により線形整数計画問題を構成する．状態(0,1,0)は，デッドロックである．

(7)　v2←v1[t1]+ [t2t4]* から，

minimize (x1+x2) subject to
(c1,c2,c3)=(1,0,0)+(0,0,2)x1+{(-1,1,0)
+(1,-1,-1)}x2
=(1,0,0)+(0,0,2)x1+(0,0,-1)x2,
(1,0,0)+(0,0,2)x1+(0,0,-1)x2≧0,
x1≧1,x2≧0

すなわち, minimize (x1+x2) subject to
c1=1, c2=0, c3=2x1-x2, 2x1-x2≧0, x1 ≧1,
x2≧0 ……(IP-1)

(イ) v2←v1[t1t2t4]+ から,
minimize x1 subject to
(c1,c2,c3)=(1,0,0)+{(0,0,2)+(-1,1,0)
+(1,-1,-1)}x1
=(1,0,0)+(0,0,1)x1, x1≧1

すなわち minimize x1 subject to
c1=1, c2=0, c3=x1, x1 ≧1 ……(IP-2)

(ウ) v3←v1[t1]+[t2][t3]* から,
minimize (x1+x2+x3) subject to
(c1,c2,c3)=(1,0,0)+(0,0,2)x1+(-1,1,0)x2
+(0,0,-2)x3.
(1,0,0)+(0,0,2)x1+(-1,1,0)x2≧0
(1,0,0)+(0,0,2)x1+(-1,1,0)x2+(0,0,-2)x3≧0
x1≧1, x2=1, x3 ≧0

すなわち minimize (x1+x3) subject to
c1=0, c2=1, c3=2x1-2x3,
2x1-2x3 ≧0, x1 ≧1, x3 ≧0 ……(IP-3)

次に3種類の目標状態への発火系列を求める.

(A) 目標状態(5,3,4) への発火系列は?

c1やc2は0か1なので適合せず, 解はないと断定.

(B) 目標状態(1,0,10)への発火系列は?

(IP-3)には確実に適合しない.

(IP-1)からminimize(X1+x2) subject to
c1=1, c2=0, 10=2x1-x2, 2x1-x2≧0, x1 ≧1,
x2 ≧0 であり, 許容解は, (x1,x2)={(5,0),(6,2),(7,4), …} である. たとえば, 許容解(6,2)より, t1を6回, t2t4を2回, この順に発火すればよい. 最適解(5,0)より, t1を5回発火すればよい.

(IP-2)からminimize x1 subject to
c1=1, c2=0, 10=x1, x1 ≧1 であり, 解は x1=10であるから, t1t2t4を10回発火すればよい.

(C) 目標状態(0,1,0) , これはデッドロック状態である. この状態への発火系列は?

(IP-1)〜(IP-2)には適合しない.

(IP-3)からminimize(x1+x3) subject to
c1=0, c2=1, 0=2x1-2x3,2x1-2x3 ≧0, x1 ≧1,
x3 ≧0 であり, 許容解はx1=x3の任意の自然数であるから, t1をn回, t2を1回, t3をn回この順序で発火すればデッドロック状態に到達する. 最適解はx1=x3=1 であるから, t1を1回, t2を1回, t3を1回この順序で発火するとデッドロック状態へ到達する.

3種類の目標状態に対するいずれの解も, 図10のループを処理していないため他の解が存在する可能性がある. すなわち, 発火系列集合の部分集合である.

4.2 デッドロックを起こさないペトリネット

図3のペトリネットに対して, 発火系列を求める. 可到達木から状態遷移式を求める手順およびその結果は図2のペトリネットで求めたものと同様である. 違いはトランジションによる増減ベクトルと線形整数計画法により求まる発火系列集合である. 状態(0,1,0)はデッドロックではない.

(エ) v2←v1[t1]+ [t2t4]* から,
minimize (x1+x2) subject to
(c1,c2,c3)=(1,0,0)+(0,0,1)x1+{(-1,1,0)
+(1,-1,0)}x2
=(1,0,0)+(0,0,1)x1+(0,0,0)x2,
x1≧1,x2≧0

すなわち, minimize (x1+x2) subject to
c1=1, c2=0, c3=x1+0x2, x1 ≧1,
x2≧0 ……(IP-4)

(オ) v2←v1[t1t2t4]+ から,
minimize x1 subject to
(c1,c2,c3)=(1,0,0)+{(0,0,1)+(-1,1,0)
+(1,-1,0)}x1
=(1,0,0)+(0,0,0)x1, x1≧1

すなわち minimize x1 subject to
c1=1, c2=0, c3=0x1, x1 ≧1 ……(IP-5)

(カ) v3←v1[t1]+[t2][t3]* から,
minimize (x1+x2+x3) subject to
(c1,c2,c3)=(1,0,0)+(0,0,1)x1+(-1,1,0)x2

+(0,0,-1)x3.

(1,0,0)+(0,0,1)x1+(-1,1,0)x2≧0

(1,0,0)+(0,0,1)x1+(-1,1,0)x2+(0,0,-1)x3≧0

x1≧1, x2=1, x3 ≧0

すなわち minimize (x1+x3) subject to

c1=0, c2=1, c3=x1-x3, x1-x3 ≧0,

x1 ≧1, x3 ≧0 ……(IP-6)

次に2種類の目標状態への発火系列を求める.

(D) 目標状態(1,0,10)への発火系列は?

(IP-6)には確実に適合しない.

(IP-4)からminimize(X1+x2) subject to c1=1, c=0, 10=x1+0x2, x1 ≧1, x2 ≧0 であり, 許容解は, (x1,x2)={(10,n) | nは0以上の整数} である. t1を10回, t2t4を任意回, この順に発火すればよい. 最適解は, t1のみの10回発火である.

(IP-5)からminimize x1 subject to c1=1, c2=0, 10=0x1, x1 ≧1 であり, 解はない.

(E) 目標状態(0,1,0), これはデッドロック状態でない. この状態への発火系列は?

(IP-4)～(IP-5)には適合しない.

(IP-6)からminimize(x1+x3) subject to c1=0, c2=1, 0=x1-x3,x1-x3 ≧0, x1 ≧1, x3 ≧0 であり, 許容解はx1=x3の任意の自然数であるから, t1をn回, t2を1回, t3をn回この順序で発火すれば到達する. 最適解はx1=x3=1 であるから, t1を1回, t2を1回, t3を1回この順序で発火する.

5. 時制論理とペトリネットの組み合わせ

並行処理システムを対象として, 時制論理とペトリネットの組み合わせの方法について述べる[Hon87].

5.1 システムの構造と制約

システムの構造と無関係に, システムの満たすべき性質を時制論理式で記述可能である. 時制論理式で並行処理システムの動作の満たすべき制約条件を記述する. たとえば, プロセスPがデッドロックに陥ってはならないという制約条件は, □◇P(Pは無限回起動される)と書く. (時制オペレータの説明は図9参照)

並行処理システムの構造も時制論理式で記述すると, 全ての制約条件の成立を時制論理式だけで形式的に証明できる. しかし, システムの構造自体の記述には必ずしも適していない.

(時制オペレータ)

○ f : 次の状態で f が真である

◇ f : 将来 f が真である

$f_1 U f_2$: f_2 が真になるまで 常に f_1 が真である

□ f : 常に f が真である

(分解ルール)

□ f = f ∧ ○ □ f

◇ f = f ∨ ○ ◇ f

$f_1 U f_2 = f_2 \vee (f_1 \wedge \bigcirc f_1 U f_2)$

図9 時制オペレータと分解ルール

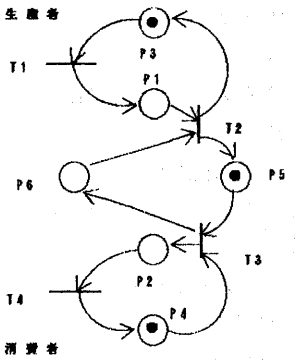


図10 生産者/消費者問題

たとえば, 複雑な並行処理システムでも, ある時点で変化する部分がごく一部であり, 他の部分は変化しない. ペトリネットでは, 発火するトランジションに無関係なプレースの状態は変わらない. このため, 変化しない部分はそのつど書く必要がない. 時制論理では, 各時間毎に真偽の解釈を行うために, 変化のない部分も記述しなければならない.

以上より, システムの構造はペトリネットで記述し, 制約条件は時制論理で記述することが望ましい.

5.2 ペトリネットと時制論理の対応

時制論理の原子論理式をペトリネットの何に対応させるかによって次の2つの方式がある.

① 原子論理式=プレースのトークンの存在: 1つの

プレースに複数のトークンが存在しない安全な(safe)ペトリネットを対象として，また一回に発火できるトランジションは1つであるとして，片井・岩井ら[Kat81, 82]は，時制命題論理に限定して，プレースPにトークンが存在することを命題の意味とした．図10のペトリネットに対する命題は，

P3，P4，P5は真，　P1，P2，P6は偽

である．このペトリネットで，P1にトークンが存在した次の時点でP2にトークンが存在しなければならないのであれば，その制約条件は，

□(P1⊃○P2)　と書ける．

② 原子論理式＝トランジションの発火：　片井らの手法と異なり，トランジションTが発火することを命題の意味とすることができる．複数の発火可能なトランジションが存在する場合，その中のどれを選択するかの制約条件を時制命題論理で表現する．図10で，トークンがP1にある間は，T4を発火しないのであれば，その制約条件は，

□(T1⊃¬T4UT2)　　と書ける．

時制論理式による制約記述は，Wolperの分解ルール(図9)によるタブロー法と呼ぶ定理証明系を使って，制約を満たす正しい制御手順に変換できる(この変換の詳細は文献[Wol82]参照)．この変換は，スケジューラにより行われ，発火の順序が制御される．

6. デッドロック回避の例題

図2のペトリネットに対して，デッドロック状態(目標状態(0,1,0))へ陥ることを回避するために，制約を与えて発火の順序を制御する．このために，線形整数計画法の出力である許容解および最適解を吟味する．

図7に対応する図8のベクトル間遷移から，さらに図11の通常の形式の状態遷移図を作っておく．

3章の例(C) は，最適解であるt1 ，t2 ，t3 の順で1回ずつ発火するとデッドロック状態になる．すなわち，t1 ，t2 の順で発火した時は，t3 を発火してはならない．時制論理式で記述する．

(P1∧¬P2∧¬P3∧t1∧○t2)⊃¬○○t3…(L-1)

許容解を吟味すると，t1 の発火回数がt3の発火回数が同等になるとデッドロック状態になることが判る．しかし，(L-1)の回避策は必要以上に制約を与えている．すなわち，事前にt1 をn 回発火していた場合にはt3をn-1回発火してもデッドロック状態には陥らないが，(L-1)の回避策はその発火系列も禁止している．

図11の状態遷移図と照合すると，(L-1)はノードgとhの自己ループを経ないで，ノードv1からv3へ進むことを示す．しかし，許容解は，gを何回かループすれば，t2t3と進んでよいことを示す．

許容解の吟味により，デッドロックに陥らないためには t3 を発火した後にはP3に1つ以上のトークンが存在しなければならない．すなわちこの例では

□((P2∧P3)⊃○(P2∧P3))　…(L-2)

と記述することにより，デッドロック状態を回避することが可能となる．図1のblock5のスケジューラに渡す．

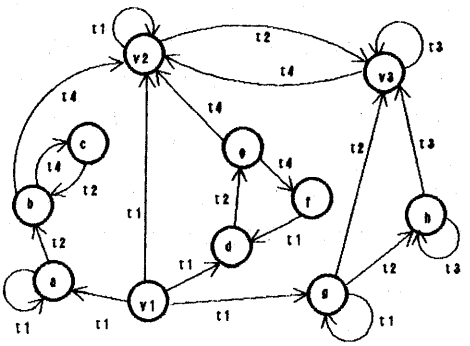


図11　図8のﾍﾞｸﾄﾙ間遷移を書き直した状態遷移図

t1 とt3 の発火回数が同数であり，かつその間にt2 が1回発火すると，デッドロックに陥る．これを，シミュレーションなどで発見することは，上記のような小さなモデルに対しても容易ではない．発火順序の検出により，「t3 を発火する時には，発火後にもP3に1つ以上のトークンが存在しなければならない」という本質的な回避策を見い出すことができる．

7. 他の手法との比較

ペトリネットと時制論理の融合について，片井らがペトリネットの可到達木と，タブロー法で生成されるモデルグラフを対応づけることにより，デッドロック状態に陥らないためのトランジションの発火禁止則を導く手法を提案した[Kat82]．片井の手法では，ペトリ

ネットと同時に対応する全ての時制論理式を与えなければならないこと，および可到達木においてωの存在を許していないことが本稿の手法と異なる．

Suzukiらは，ペトリネットと時制論理を融合したTemporal Petri Nets を提案した[Suz89]．この記述は決定不能である．Suzukiらは時制論理式の採用して，eventualityなどの性質がペトリネットにおいて満たされているかの検証を行った．

Leveson はhigh-risk 状態に陥るかどうかの検出およびその回避のために低コストで解が得られる簡略化した可到達グラフを作成している[Lev87]．系列の検出および回避を目的としている点と，簡便さを重視している点では類似性がある．しかし，可到達グラフはωを含まないこと，回避策の生成についてはサポートしていないこと，さらに要素技術(時制論理，および線形計画法)が大きく異なる．

8. おわりに

本稿において新たに提案したことは2点である．

①ペトリネットの可到達問題の解である発火系列集合の部分集合を，線形整数計画法で求めることが可能．

②デッドロックなどの特殊な状態に陥る発火系列が求まると，それを回避するための回避策の記述に対して時制論理が有効．

この①について，線形整数計画問題で解ける可到達性問題の解の部分集合の理論的な同定が課題である．

また，著者らは現在，並行処理システムを対象としたソフトウェア開発環境を開発中[Hon89ab]である．並行プロセス間の関係はペトリネットとして自動抽出され，それに対して線形計画法によって発火系列を求め，それを踏まえて回避策を時制論理で与える．この回避策は並行プロセスを管理するスケジューラに渡され，並行プロセス間の同期制御を行う．

参考文献

[Hac76] Hack,M. : The Equality Problem for Vector Addition System is Undecidable, Theoret. Comput. Sci.,vol.2, pp.77-95 (1976)

[Hon87]本位田真一他：時制論理とペトリネット，オペレーションズリサーチ，Vol.32, No.9 (1987)

[Hon89a] Honiden,S. et al. : An Application of Structural Modeling and Automated Reasoning to Concurrent Program Design, in Proc.of HICSS-22 (1989)

[Hon89] 本位田真一他：線形整数計画法と時制論理のペトリネット到達可能問題への適用，第2回回路とシステム軽井沢ワークショップ論文集，pp.1-8(1989)

[Ito81]伊藤　潔：ペトリネット解析における双方向状態遷移式の構成と状態類別の手順について，情報処理学会第23回全国大会論文集(1981)

[Kar69] Karp,R.M.,Miller,R.E. : Parallel Program Schema, Journal of Computer and System Science 3 (1969)

[Kat81]片井　修他：並列プロセスの動態に対する時制論理に基づく記述・検証体系，システムと制御，Vol.25,No8 (1981)

[Kat82]片井　修他：非同期同時進行システムに対する時制論理に基づくスケジューリング則の構成，計測自動制御学会論文集，Vol.18, No.12 (1982)

[Kos82] Kosarazu,S.R. : Decidability of Reachability in Vector Addition Systems,Proc. of 14th ACM STOC (1982)

[Lam88] Lambert,J.L. : Some Consequences of the Decidability of the Reachability Problem for Petri Nets, LNCS No.340 (1988)

[Lev87] Leveson,N.G., Stolzy,J.L. : Safety Analysis Using Petri Nets, IEEE Trans. on S.E. Vol.SE-13,pp.386-397, March (1987)

[May84] Mayr,E. : An Algorithm for the General Petri Net Reachability Problem, SIAM, J.Comput., Vol.13, No.3,pp.441-460 (1984)

[Pet84]ピータスン(市川，小林共訳)：ペトリネット入門，共立出版(1984)

[Suz89] Suzuki,I. et al : Temporal Petri Nets and Their Application to Modeling and Analysis of a Handshake Daisy Chain Arbiter, IEEE Trans. Comput., Vol.38, No.5,pp.696-704,(1989)

[Wol82] Wolper,P.L. : Specification and Synthesis of Communicating Processes Using an Extended Temporal Logic, Ph.D. Thesis,Department of Computer Science, Stanford Univ.(1982)