OUKA
The University of Osaka
Institutional Knowledge Archive

Title	離散事象制御系のネットモデルによる解析に関する研究
Author(s)	辻, 孝吉
Citation	大阪大学, 1987, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/635
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

離散事象制御系の
ネットモデルによる解析に関する研究

昭和62年1月

辻　孝吉

内容梗概

本論文は、筆者が昭和59年春から大阪大学大学院工学研究科電子工学専攻博士後期課程に在学中、児玉研究室において行った”離散事象制御系のネットモデルによる解析に関する研究”の成果をまとめたものである。

複雑な動作をする大規模な並行離散事象システムを体系的に取り扱うためのモデルとしてペトリネットの有用性が次第に認識されつつある。特に、ネット理論のような理論的基盤のもとで具体的な並行システムをペトリネットでモデル化することは、仕様記述からシステム設計、動作検証、故障診断、システム制御に至るまでの全過程を統一的に取り扱うことができ、このようなアプローチに基づく離散事象システムの設計理念を確立することは工学上極めて有意義なことである。大規模な並行離散事象制御系の解析における基本問題には、初期状態から目標の状態への到達可能性の問題および設計された制御フローや資源配置などにおけるデッドロックの検証問題がある。ペトリネットモデルにおいてこれらの基本問題はそれぞれ与えられた初期マーキングに関する可到達性問題およびライブ性問題として定式化されている。これらの二問題はそれぞれ等価な決定可能問題であることがわかっているが、実用上有効な検証アルゴリズムは極く限られたクラスのペトリネットについてしか得られていない。本論文は、特に生産システムで現れる複合シーケンス制御系をモデル化するペトリネットのクラスについてそのライブ性に関する理論的考察を行い、上記の目的に対する一つの基礎を与えようとするものである。本論文は５章からなり、その主要部分である第２、３、４章の内容の大略を述べると、応用上有用であるペトリネットのサブクラスの構造的性質とその有界性・ライブ性との関係の導出、ペトリネットを用いて階層的設計を行う場合のライブ性などの諸性質を保存する詳細化規則の導出、およびシーケンス制御系のペトリネットによるモデル化と動作検証である。以下、各章の内容をやや詳しく説明する。

第２章では、ペトリネットのサブクラスであり、並列処理などを表現するモデルとして応用上有用であるマークグラフに制御枝（ペトリネットの往復枝に相当）と呼ばれる枝を付加した拡張マークグラフのあるサブクラスの解析を行う。そして、ライブであるための必要十分条件を導出する。特に、ネット構造

と初期マーキングに関する適用しやすい条件として導出する。更に、本章で得られた結果が、従来の条件ではライブ性を調べることが困難であったネットに対しても有効であることを示す。

第３章では、従来の設計方法が、いくつかのサブシステムを組合わせて一つの統合されたシステムを構成する方法や、システムの大枠から設計を行い各部分を順次具体化してゆくという階層的設計方法であることから、ペトリネットモデルのトランジションをそのトランジションが表す動作を詳細表現した他のペトリネットで逐次置き換えるというトップダウン形式のシステム表現を考える。そして、詳細化する前のペトリネットの動作が詳細化後のペトリネットにおいても完全に保存される変換規則の導出を考える。特に、複数のトランジションを詳細化したとき、各々のサブネットの動作に相互関係が生ずる場合が多々あり、それらのサブネットを接続する必要が生ずる。そこで、従来の単一トランジションの詳細化に対して得られている性質の保存性に関する結果を、複数のトランジションの詳細化に対して一般化する。この規則に従って詳細化すれば、元のネットの有界性、安全性及びライブ性等の諸性質が、詳細化後のネットにおいても保存されることを示す。そして、このように階層表現することによって、大規模離散事象システムの動作検証の高効率化が図れることを示す。

第４章では、並行離散事象システム、特に、シーケンス制御系のモデル化と動作検証を行う。シーケンス制御系は、種々の制御対象と制御部を持つワークステーション、およびそれら相互間の通信網からなる分散システムと考えることができ、制御対象のモデル化は比較的容易に行え、その動作も比較的単純である。分散システムの階層的モデルを構築するための第一段階として、複雑な動作をすることが多い制御部のモデル化を考える。特に、従来慣用されている記述法との親和性という応用上の観点から、ＰＣ（Programmable Controller）のプログラムの際によく用いられる展開接続図（ラダー回路）や論理図のペトリネットへの自動変換を可能とする変換規則を提案する。そして、本章で提案するモデルは、これまで提案されたモデルに比べて、多くの利点を持っていることを示す。更に、不必要な動作の停止などが起きないように正しく設計された展開接続図のペトリネットモデルが有界でありかつライブであることを示す。次に、システムの一部の検証だけが必要な場合やネットモデルをより小さな規

模で実現する場合に有用であるラダー回路のペトリネットモデルの縮約規則を示す。また、ペトリネットがライブであるための必要十分条件を用いれば、設計者が見落しがちな意味のない素子も容易に検出できることを示す。

第５章は結論であり、本論文で得られた結果を述べるとともにその有効性について述べる。

関連発表論文

＜論文誌等＞

(1) K.Tsuji, S.Kumagai, S.Kodama and S.Takeda:" On the liveness of extended marked graphs," Proceedings of the IEEE International Symposium on Circuits & Systems, pp.471-474 (1985).

(2) K.Tsuji, S.Kumagai, S.Kodama and T.Yamada:" Modelling and verification of sequential control systems by Petri nets," Proceedings of the IEEE International Symposium on Circuits & Systems, pp.988-991 (1986).

(3) 辻，熊谷，児玉，山田，黄：" シーケンス制御系のペトリネットによるモデル化と動作検証" 電子通信学会論文誌，J69-A，7，pp.840-849 (1986).

(4) K.Tsuji, S.Kumagai and S.Kodama:" On the liveness of extended marked graphs," 電子通信学会英文論文誌，E69，12，(1986).

(5) 山田，辻，熊谷，児玉：" ペトリネットモデルによる分散システムのモデル化と動作検証" 電子通信学会論文誌，J70-A，2，(1987).

＜研究会等＞

(1) 山本，山田，内藤，辻，熊谷，児玉：" ペトリネットに基づいた並行システムシミュレータPCSSについて" 第24回計測自動制御学会学術講演会予稿集，pp.917-918 (1985).

(2) 辻，熊谷，児玉，黄：" シーケンス制御系のペトリネットによるモデル化と動作検証" 電子通信学会技術報告，CAS85-98，pp.29-36 (1985).

(3) 山田，辻，森田，熊谷，児玉：" 簡約ネットモデルによる並行システムの階層的シミュレーションについて" 昭和61年電子通信学会総合全国大会予稿集，pp.302-303 (1986).

(4) 山田，辻，熊谷，児玉：" ペトリネットモデルによる分散システムのモデル化と動作検証" 情報処理学会マルチメディア通信と分散処理研究会，29-3，(1986).

(5) 黄，辻，熊谷，児玉：" ペトリネットによるスイッチング回路の解析" 電子通信学会ネット理論研究会予稿集，pp.15-26 (1986).

離散事象制御系の
ネットモデルによる解析に関する研究

目　次

第１章　緒論 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 1

第２章　拡張マークグラフのライブ性
2.1　緒言 ・・・・・・・・・・・・・・・・・・・・・・・・・・ 7
2.2　記号と準備 ・・・・・・・・・・・・・・・・・・・・・・・ 7
2.3　拡張マークグラフのライブ性 ・・・・・・・・・・・・・・・12
2.4　ライブであることを保証できるクラスの拡張 ・・・・・・・・18
2.5　例題 ・・・・・・・・・・・・・・・・・・・・・・・・・・32
2.6　定理の証明 ・・・・・・・・・・・・・・・・・・・・・・・32
2.7　結言 ・・・・・・・・・・・・・・・・・・・・・・・・・・36

第３章　ネットモデルによるシステムの階層的設計
3.1　緒言 ・・・・・・・・・・・・・・・・・・・・・・・・・・37
3.2　記号と準備 ・・・・・・・・・・・・・・・・・・・・・・・38
3.2.1　ペトリネット ・・・・・・・・・・・・・・・・・・・・38
3.2.2　well-behavedness ・・・・・・・・・・・・・・・・・・40
3.3　トランジションの詳細化 ・・・・・・・・・・・・・・・・・43
3.4　ライブ性・有界性の保存 ・・・・・・・・・・・・・・・・・50
3.5　大規模システムの動作検証 ・・・・・・・・・・・・・・・・58
3.6　結言 ・・・・・・・・・・・・・・・・・・・・・・・・・・60

第４章　シーケンス制御系のモデル化と動作検証
4.1　緒言 ・・・・・・・・・・・・・・・・・・・・・・・・・・61
4.2　記号と準備 ・・・・・・・・・・・・・・・・・・・・・・・62
4.3　ペトリネットによるモデル化 ・・・・・・・・・・・・・・・64

4.3.1　展開接続図のモデル化 ・・・・・・・・・・・・・・・・・64
4.3.2　他のモデルとの比較 ・・・・・・・・・・・・・・・・・72
4.4　ペトリネットモデルの機能検証 ・・・・・・・・・・・・・72
4.5　ペトリネットモデルの縮約 ・・・・・・・・・・・・・・・81
4.6　例題 ・・・・・・・・・・・・・・・・・・・・・・・・・84
4.7　結言 ・・・・・・・・・・・・・・・・・・・・・・・・・86

第５章　結論 ・・・・・・・・・・・・・・・・・・・・・・・・87

謝辞 ・・・・・・・・・・・・・・・・・・・・・・・・・・・90

参考文献 ・・・・・・・・・・・・・・・・・・・・・・・・・91

第１章　緒　論

システム理論や制御理論で扱われるシステムのクラスで馴染みがあるのは、物理法則を反映した微分方程式や差分方程式により記述されるシステムである。これに対し、どのようなレベルのふるまいを問題にするのかによって、離散事象システムと呼ばれるシステムのクラスがある。離散事象システムは、離散的な事象がある条件に従って生起するかどうかを問題にし、それ以下のレベルは上述のシステムのクラスで扱う問題として考慮しないシステムである。前者のシステムの理論は、微分方程式や差分方程式などのモデルを用いて形成され、伝達関数や状態方程式のようにシステムの入出力関係に注目することにより表現されている。離散事象システムを入出力の関係から見ると、ある入力条件が成立すると事象が発生し、その結果出力条件が成立するという事象駆動システムと考えることができる。このような最小単位のシステムが複数個（たとえば二つ）接続している場合を考えてみる。すると、それらの間には、独立に発生しうるという並列性、一方が先に必ず発生するという先行関係、一方が発生すると他方は発生しないという選択性（競合性）などの関係が起こり、全体として非常に複雑な動作をする。このため、離散事象システムの取扱が困難となってくる。このような複数の離散的な事象が同時に並列進行する並列離散事象システムには、FA・OAシステム、通信プロトコル、並列計算機、並列分散システム、シーケンス制御系など多くのシステムが含まれる。

離散事象システムのモデルとしては、オートマトンなどが知られているが、オートマトンは選択性や先行関係を表現することはできるが、並列性を表現することができないという短所を持っている。これに対し、並列離散事象システムをモデル化するのに適したモデルとしてペトリネット[1]がある。ペトリネットは、プレース，トランジションと呼ばれる二種類の節点集合とそれらの間を結ぶ有向枝からなる二部グラフである。通常、プレースは事象生起に関するシステムの条件を表し、トランジションは事象の生起と完了を表す。そして、有向枝はそれらの関係を表しており、選択性や先行関係および並列性を表現することができ、前述の事象駆動システムをモデル化するのに適している。更に、プレースで表される条件が成立している場合にはプレースにトークンを置き、

事象の発生とともにそれらを変化させる。これにより、ペトリネットは静的な関係を表現するだけでなく、動的な挙動をも表現している。

ペトリネットは、最近さらに一般化されてネット理論というシステム理論の一分野が形成されつつあり、このような理論的基盤のもとで具体的な並行システムをペトリネットでモデル化することにより、仕様記述からシステム設計、動作検証、故障診断、システム制御に至るまでの全過程を統一的に取り扱うことができ、このようなアプローチに基づく離散事象システムの設計理念を確立することは工学上極めて有意義なことであると考えられる。更に、時間付ペトリネット(Timed Petri Net)[2]を用いれば、実システムにおける時間遅れをも含めて考えることができる。大規模な並行離散事象制御系の解析における基本問題には、初期状態から目標の状態への到達可能性の問題および設計された制御フローや資源配置などにおけるデッドロックの検証問題がある。ペトリネットモデルにおいてこれらの基本問題はそれぞれ与えられた初期マーキングに関する可到達性問題およびライブ性問題として定式化されている。これらの二問題はそれぞれ等価である決定可能な問題であることがわかっている[1]が、実用上有効な検証アルゴリズムは極く限られたクラスのペトリネットについてしか得られていない。本論文は、特に生産システムで現れる複合シーケンス制御系をモデル化するペトリネットのクラスについてそのライブ性に関する理論的考察を行い、上記の目的に対する一つの基礎を与えようとするものである。

ところで、ペトリネットの基本的性質である可到達性問題に関しては、Mayrによって可到達性問題を解くためのアルゴリズムが提案されている[3]。従って、可到達性問題と等価であるライブ性問題も解けたといえる。しかし、このアルゴリズムの手数は、ネットの規模に対して指数オーダとなり、実用的であるとはいえない。一方、ペトリネットに制限を加えることにより工学的に有用で、より簡明な解析的条件が得られる場合がある。しかし、ペトリネットの表現能力と解析能力を考えると、それらは相反する関係にあるのでどちらか一方の能力を上げようとすれば、もう一方の能力が下がってしまう。したがって、実システムをモデル化する場合には表現能力と解析能力とのバランスを考えながら、適当なサブクラスを定義することにより解析されねばならない。このため、ペトリネットの幾つかのサブクラスに対してライブであるための検証しや

すい必要十分条件の導出が試みられている[4]～[9]。例えば、ペトリネットのサブクラスであり、並列処理や並列演算などのモデルとして応用上有用であるマークグラフにおいて、可到達性やライブ性に関する必要十分条件がネット構造と初期マーキングに関する条件として得られている[6],[7],[8]。しかし、選択性（競合性）をマークグラフでモデル化することはできない。そこで、更にモデル化能力を高めるために、マークグラフに制御枝（ペトリネットの往復枝に相当）と呼ばれる枝を付加した拡張マークグラフ[9]のあるサブクラスに対して、ライブであるための必要十分条件がネット構造と初期マーキングに関する条件として得られている[9]。しかし、従来の拡張マークグラフに対する制限は厳しすぎ、モデル化能力はあまり高いとは言えない。

第２章では、更にその制限を緩め、モデル化能力を高めた拡張マークグラフのあるサブクラスに対して、ライブ性の必要十分条件をネット構造と初期マーキングに関する条件として導く。また、例題を通して、その有効性を示す。

次に、一般的なシステムの設計ということを考えてみる。従来、大規模システムの設計においては、いくつかのサブシステムを組合わせて一つの統合されたシステムを構成する方法や、システムの大枠から設計を行い各部分を順次具体化してゆく方法がとられている。このような手法において、柔軟かつ迅速な設計の自動化を図るためには、各設計段階における仕様記述・解析手法の形式化・体系化が重要である。このため、ペトリネットをシステムの仕様記述から動作解析・検証に至るまで一貫して用いることにより、上記の目的を達成する設計手法が構築し得るものと期待される。また、ペトリネットモデルの利点の一つは、トランジションをそのトランジションが表す動作を詳細表現した他のペトリネットで逐次置き換えることにより、トップダウン形式のシステム表現が可能となることである。つまり、従来からある階層的設計方法に適したモデルである。このような階層表現において問題となるのは、詳細化する前のペトリネットの動作が詳細化後のペトリネットにおいても完全に保存されているかどうかということである。逆に言うと、このような性質を保証する代置ネットの構造と初期トークンの配置を明らかにしておく必要がある。従来、単一トランジションの詳細化に関して上記の問題はValette[10]及び鈴木、村田[11]らによって考察され、逐次詳細化されるシステムのライブ性、有界性などの性質

が、元のシステムの性質と代置ネットの性質を調べることにより検証し得ることが判っている。ところが、複数のトランジションを詳細化したとき、各々のサブネットの動作に相互関係が生ずる場合が多々あり、それらのサブネットを接続する必要が生ずる。換言すれば、階層化のレベルに従いシステム各部の相互干渉が陽に現れてくるのである。従って、上位レベルで互いに関連のありそうな複数のトランジションをまとめて一つのネットで詳細化できれば、より効率的な階層表現が可能となる。

第３章では、階層表現における層間での性質の保存性に関する文献(10)や文献(11)の結果を、システムの任意の部分（トランジションの集合）の詳細化に対して一般化する。また、この階層表現によって大規模分散システムの動作検証が容易になることを示す。

更に、実システムのモデル化と動作検証について、シーケンス制御系を取上げて考えてみる。ＦＡ，ＦＭＳで代表されるようにシーケンス制御の対象が大規模かつ複雑化・階層化してくるに伴い、仕様記述から具体的回路設計に至るまでの設計過程に一貫した階層的システムモデルと動作記述言語の必要性が生じてくる。シーケンス制御系設計の体系化の必要性が叫ばれて久しいが(12)、上記のような大規模系の計算機援用設計方式を確立するまでには至っていない。ところで、シーケンス制御系は、離散的な事象がある条件に従って並列・非同期で互いに干渉し合いながら動作する並列離散事象システムと考えることができ、ペトリネットを用いることができる。シーケンス制御系へペトリネットを応用した例としては、文献(13)～(16)などがあり、シーケンス制御系設計方式の体系化の手段として、その有効性が次第に認識されつつある。ＦＡ，ＦＭＳで代表される複合シーケンス制御系は、種々の制御対象と制御部を持つワークステーション、およびそれら相互間の通信網からなる分散システムと考えることができる。論理シミュレータなどでも各制御部の解析を行うことはできるが、分散システム全体の並行動作や階層的動作記述を行うことは困難である。一方、このような分散システムの各部およびその上位レベルとの通信機能をまとめてペトリネットでモデル化することにより、シーケンス制御系全体の構造的あるいは階層的な動作解析を行うことが可能となる。このためには、まず、シーケンス制御系での最下位レベルの構成単位である制御部および制御対象のペトリ

ネットによる適当なモデル化が必要となる。この際、制御対象のモデル化は比較的容易に行え、その動作も比較的単純である。

第４章では、上記のような分散システムの階層的モデルを構築するための第一段階として、複雑な動作をすることが多い制御部のモデル化を考える。従来慣用されている記述法との親和性という応用上の観点から、特に、ＰＣ（Programmable Controller）のプログラムの際によく用いられる展開接続図（ラダー回路）や論理図のペトリネットへの自動変換を可能とする変換規則を提案する。そして、本章で提案するモデルは、これまで提案されたモデル[16],[17]に比べて、もとの制御システムでは存在しない競合がモデルにおいて生起するという不都合がない、モデルの規模が小さい、変更・保守およびモジュール化・階層化が容易である、などの特徴を持っていることを示す。更に、不必要な動作の停止などが起きないように正しく設計された展開接続図のペトリネットモデルが有界かつライブであることを示す。また、システムの一部の検証だけが必要な場合やネットモデルをより小さな規模で実現する場合に有用であるネットモデルの縮約規則を示す。最後に、本章で提案したモデルの有効性をシミュレータPCSS(Petri net-based Concurrent System Simulator)[18]によるシミュレーションにより確かめる。

第５章は結論であり、本論文で得られた結果を述べるとともにその有効性について述べる。

第２章　拡張マークグラフのライブ性

２．１　緒言

並列計算機・並列分散システム・シーケンス制御系などのような並列離散事象システムのモデルとしてペトリネット[1]-[7]がある。このようなシステムの制御問題の多くは、基本的にペトリネットの可到達性やライブ性を解くことに帰着する。最近、Mayr[2]が可到達性問題を解くためのアルゴリズムを提案しており、可到達性問題と等価であるライブ性問題もこの意味で解決したと言える。しかし、このアルゴリズムの手数はネットの規模に対して指数オーダとなり実用的でない[2],[3]。一方、ペトリネットに制限を加えることにより工学的に有用で、より簡明な解析的条件が得られる場合がある。例えば、ペトリネットのサブクラスであり、並列処理や並列演算などのモデルとして応用上有用であるマークグラフにおいて、可到達性やライブ性に関する必要十分条件がネット構造と初期マーキングに関する条件として得られている[4],[5],[6]。しかし、競合性（一方が発生すると他方は発生しない）をマークグラフでモデル化することはできない。そこで、更にモデル化能力を高めるために、マークグラフに制御枝（ペトリネットの往復枝に相当）と呼ばれる枝を付加した拡張マークグラフ[1]のあるサブクラスに対しても、ライブであるための必要十分条件がネット構造と初期マーキングに関する条件として得られている[1]。しかし、従来の拡張マークグラフに対する制限は厳しすぎ、モデル化能力はあまり高いとは言えない。

まず、2.3節において従来得られている結果を述べる。次に、2.4節では更にその制限を緩め、モデル化能力を高めた拡張マークグラフのあるサブクラスに対して、ライブ性の必要十分条件をネット構造と初期マーキングに関する条件として導く。最後に、2.5節で例題を通してその有効性を示す。

２．２　記号と準備

本章で用いる記号・用語の説明を行う。ペトリネットは、プレース（○で表す）とトランジション（｜で表す）と呼ばれる二種類の節点集合とその間の有向枝集合からなる二部グラフである。ｐからｔ（ｔからｐ）の向きに枝が存在

する場合に、ｐをｔの入力（出力）プレースと呼び、その集合を ･t（t･）で表し、ｔをｐの出力（入力）トランジションと呼び、その集合をp･（･p）で表す。サブネットＬに含まれるプレース（トランジション）の集合をＳ(Ｌ)（Ｔ(Ｌ)）で表す。Ｓの次数をｎとする。通常、プレースは事象生起に関するシステムの条件を表し、トランジションは離散事象の生起と完了を表現する。そして、プレースに対応する条件が現時点で成立している場合には、そのプレースにトークン（●で表す）を置き、トランジションの発火によるトークンの遷移によりシステムの状態遷移を表す。各プレースのトークン数を要素にもつ非負整数ベクトルをマーキングM∈N^nと呼び、M(p)でプレースｐのトークン数を表す。任意のｐ∈･tについて、M(p)≧1 が成り立つとき、ｔはマーキングMで発火可能であるといい、Mで発火可能なトランジションｔの発火により、Mは次の発火規則に従ってM’に変化する。

$$\begin{cases} M'(p)=M(p)+1 & : p \in t^{\bullet} \wedge p \notin {}^{\bullet}t \quad (2.1) \\ M'(p)=M(p)-1 & : p \in {}^{\bullet}t \wedge p \notin t^{\bullet} \quad (2.2) \\ M'(p)=M(p) & : \text{otherwise} \quad (2.3) \end{cases}$$

マーキングMから到達可能なマーキングの集合をR(M)で表す。任意のM∈R(Mo)に対して、M’∈R(M)が存在して、M’においてｔが発火可能となるとき、Mo においてｔはライブ(live)であるという。ライブでないｔはデッド(dead)であるという。また、すべてのｔがライブであるとき、Mo においてペトリネットはライブであるという。デッドなｔを含むペトリネットはデッドであるという。任意のマーキングにおいて、プレースｐに対してM(p)≦K_pを満足する正整数K_pが存在するとき、ｐは有界(bounded)であるという。また、すべてのｐが有界であるとき、ペトリネットは有界であるという。

Ｄをプレースの集合とするとき、･D⊆D･（D･⊆･D）を満足するプレースの集合をデッドロック（トラップ）と呼ぶ。デッドロックＤ（トラップＥ）の真部分集合が存在しないとき、Ｄ（Ｅ）は最小（minimal）であると呼ぶ。あるマーキングMにおいてデッドロックにトークンが存在しないとき、Mから到達可能なすべてのマーキングにおいてデッドロックにトークンは存在しない。

また、あるマーキングMにおいてトラップにトークンが存在するとき、Mから到達可能なすべてのマーキングにおいてトラップにトークンは少なくとも一つ存在する。

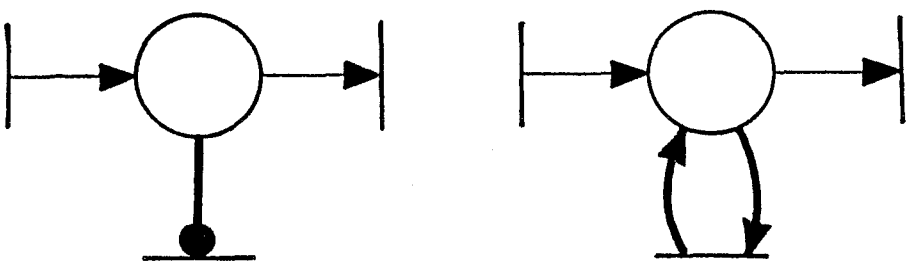
また、初期マーキングMo に対し、デッドなトランジションのR(Mo)に関する極大集合Deadmを

$$Dead_M := \sup\{t \in T \mid t \text{はM}' \in R(Mo) \text{でデッド}\} \qquad (2.4)$$

と定義する。ただし、Mは右辺を与えるマーキングを表す。

本章では、往復有向枝対 "⇄" を "⊸" で表し、この枝を制御枝と呼ぶ。制御枝を持つプレースｐはトランジションｔの発火に対する許可条件を表し、ｔの発火によるトークンの変化を伴わない。このようなプレースｐをｔの制御プレース(controlling place)と呼び、ｔ を被制御トランジション(controlled transition)と呼ぶ。

マークグラフ（marked graph）はペトリネットのサブクラスであり、各プレースの入出力トランジションの数がそれぞれ１のネットである。拡張マークグラフ（extended marked graph）は、マークグラフに制御枝の存在を許したより広いクラスのネットである（図2.1参照）。また、拡張マークグラフは無競合（conflict-free）ペトリネットやＦＣネット、ＥＳネット[7]のクラスに包含されないクラスに属している。



(a) A controlling arc　　(b) Its Petri net model

図2.1　拡張マークグラフ

Fig. 2.1　An extended marked graph

ここで、最小デッドロックD（図2.2(a)）を考える。Dの出力トランジションであるがDの入力トランジションでないトランジションtを排出トランジション（outlet transition）と呼び、その集合をToで表す。また、被制御トランジションtc∈ p∩p の集合をTcで表し、特に、Dに存在する被制御トランジションの集合をTbで表す。被制御トランジション(tb∈Tb, tb∉To)から排出トランジションto∈Toへの有向路（directed path）に存在するプレースの集合がDに存在するプレースの集合に含まれないとき、その有向路をtoの自己制御路（self-controlling path）と呼び、Pcで表す。Dに存在するtb∈TbからTo∈Toへの有向路をtoの誘導路（leading path）と呼び、PLで表す。簡単のために、これらの有向路に存在するプレース（トランジション）の集合や部分ネットにも同じ記号を用いる。有向路Pの集合を{P}で表す。tb∈Tbやto∈Toを節点で表し、Pcの方向を逆にすることによって得られる図2.2(b)のような有向グラフをDのデンドロイド（dendroid）Rと呼ぶ。デンドロイドは図2.2のように閉路（circuit）とそこから伸びる触手枝（feeler）から構成される。

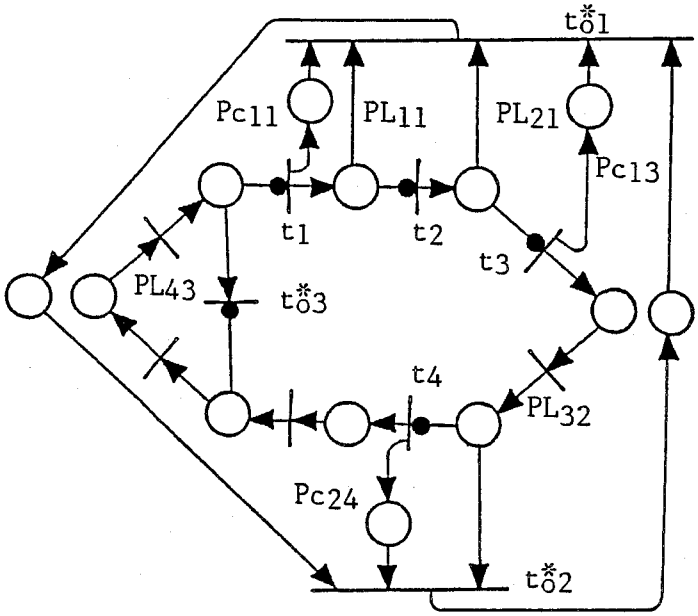
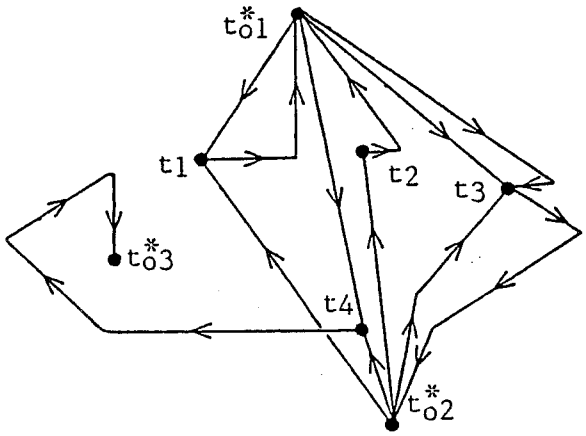


図2.2(a) A deadlock containing traps



(b) A dendroid of (a)

図2.2　最小デッドロック

Fig. 2.2　A minimal deadlock

＜注2.1＞

簡単のために、t∈Tbからt'∈Toへの有向路P（|p•|=|•p|=1, p∈P）が複数存在し、それらに共通に存在するトランジションがtとt'だけであるとき、t∈Tbからt'∈Toへの有向路を一本の自己制御路Pcで表す。ただし、Pcはt∈Tbからt'∈Toへの有向路のうちでトークンの和が最小の有向路P'∈{P}である。また、tを発火することなしにt'を発火できる回数はトークン和が最小の有向路によって制限される。　□

分散処理システムの拡張マークグラフによるモデル化の例を図2.3に示す。

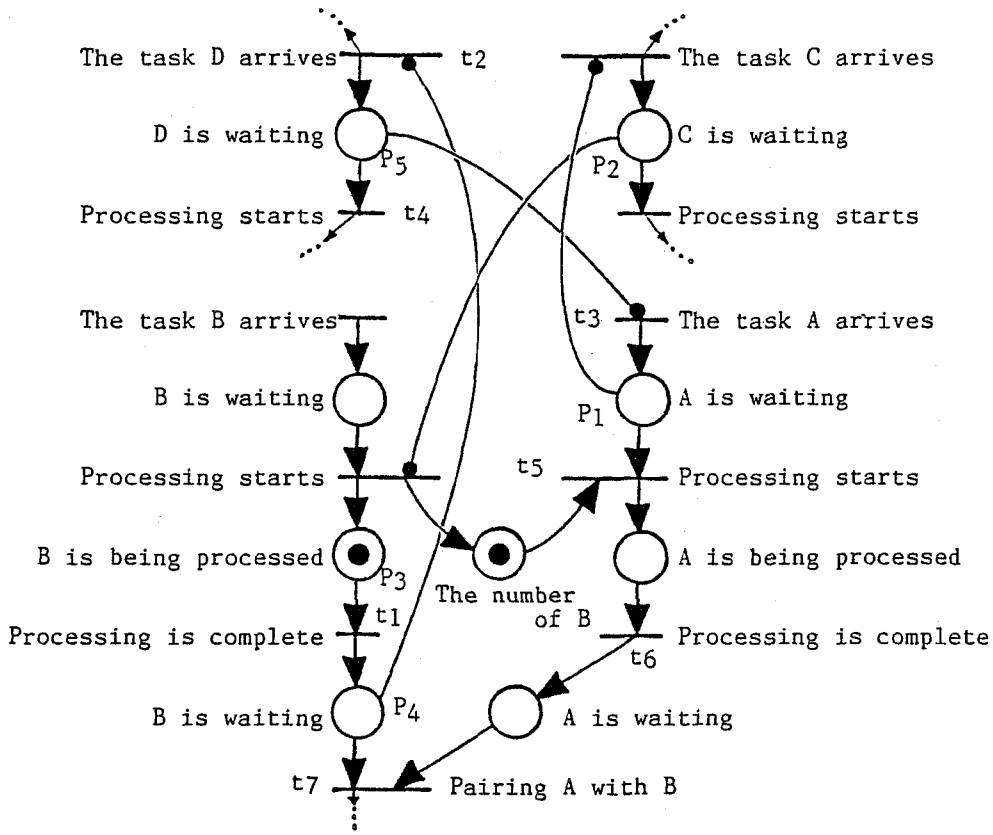


図2.3　分散処理システムの拡張マークグラフによるモデル化

Fig. 2.3　An extended marked graph model of malti-task systems

２．３　拡張マークグラフのライブ性

前節で定義されたデッドロックやトラップをもとに、ライブ性のような動的なふるまいとネット構造との関係を調べる。

拡張マークグラフに存在する最小デッドロックや最小トラップを考える。図2.4のように、拡張マークグラフに存在する有向閉路（directed circuit）は

最小デッドロックかつ最小トラップである。拡張マークグラフに存在するそれ以外の最小デッドロックや最小トラップの構造は図2.5のようになる。

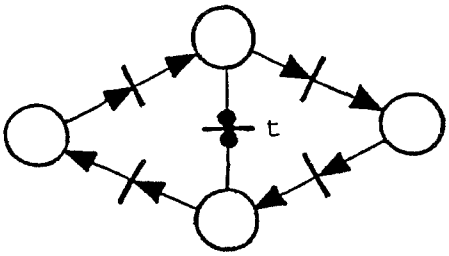
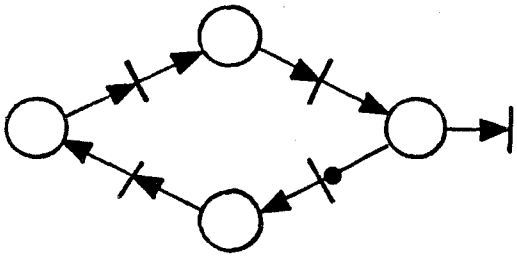
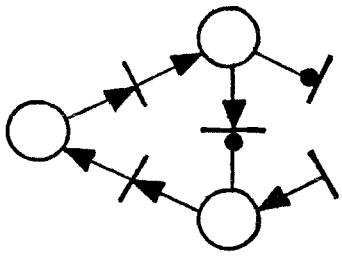


図2.4　拡張マークグラフに存在する有向閉路

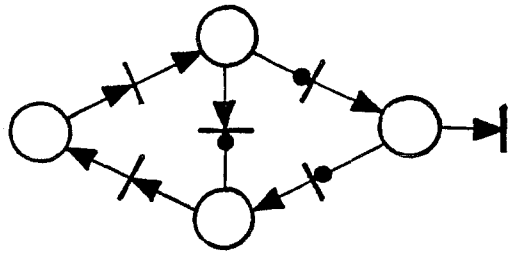
Fig. 2.4　A directed circuit in extended marked graphs



(a) A deadlock



(b) A trap



(c) A deadlock containing a trap

図2.5　拡張マークグラフに存在する最小デッドロックと最小トラップ

Fig. 2.5　A minimal deadlock and trap in extended marked graphs

図2.4のような拡張マークグラフの簡単な例を考えてみよう。トランジションtが発火可能であるためには、有向閉路に存在するトークンの和が2以上である必要がある。そこで、本章を通して次の仮定が成立しているネットを扱う。

<仮定2.1>

各トランジションtに対して、tの制御プレースをn個含む有向閉路が存在する場合、その有向閉路に存在するトークンの和はn以上である。 □

ペトリネットに存在する任意のデッドロックDに対して、M_0から到達可能なあるマーキング$M \in R(M_0)$において $M(D)=0$ を満足しているならば、Mから到達可能なすべてのマーキング$M' \in R(M)$において$M'(D)=0$を満足する。また、任意のトラップTrに対して、あるマーキング$M \in R(M_0)$において $M(Tr) \neq 0$を満足しているならば、すべてのマーキング$M' \in R(M)$において$M'(Tr) \neq 0$を満足する。したがって、あるデッドロックDに対して、$M(D)=0$を満足するマーキングMが存在するとき、D に含まれるすべてのトランジションtはマーキングMでデッドあることがわかる。このことから、ペトリネットがライブであるための必要条件は、トークンが存在しない空のデッドロックが存在しないことであることがわかる。しかし、図2.4の例のようにマーキングによって、各デッドロックが空にならなくてもネットがデッドとなる場合があるので、この条件は一般的には十分条件とはならない。拡張マークグラフについては、文献(1)で導出されている次の性質によって、仮定2.1を満足しているとき、空のデッドロックの存在しないことがライブであるための十分条件であることがわかる。

[性質2.1] (1)

すべてのトランジション$t \in Dead_n$に対して、任意のマーキング$M' \in R(M)$について$M'(p)=0$となるプレース$p \in {}^{\bullet}t$が存在する。 □

このとき、拡張マークグラフがライブであるための必要十分条件が得られる。

[性質2.2]〔1〕

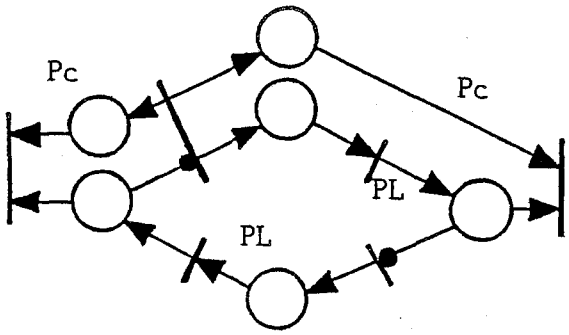
仮定2.1が満足されているものとする。このとき、拡張マークグラフがMoにおいてライブであるための必要十分条件は、初期マーキングMoから到達可能な任意のマーキングMにおいて空のデッドロックが存在しないことである。□

マークグラフにおいては、有向閉路(デッドロックかつトラップ)内のトークンの和は任意のマーキングM∈R(Mo)において不変である。しかし、拡張マークグラフにおいては、デッドロック（トラップ）内のトークンの和は任意のマーキングM∈R(Mo)において不変ではない。したがって、検証が容易である条件を導出するためには、任意のマーキングM∈R(Mo)においてデッドロック内のトークンの和が正となる条件を見出さなければならない。具体的には、図2.5に示したような最小デッドロックについての条件を考えればよい。たとえば、図2.5(c)のようにデッドロックDがトラップD'を含む場合を考えよう。このとき、トラップの性質から、Mo(D')≠0を満足しているならば任意のマーキングM∈R(Mo)においてD内のトークンは空にならない。このようなデッドロックDは、Moにおいて<u>トラップ条件（trap-condition）</u>を満足しているという。

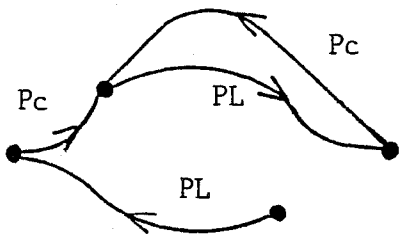
拡張マークグラフがライブであり、トラップ条件を満足しない最小デッドロックを含む場合を考えよう。たとえば、図2.6のようなトラップを含まない最小デッドロックDを考える。もし、すべての排出トランジションに自己制御路が存在しないならば、排出トランジションを発火することによってデッドロック内のトークンを空にすることができる。したがって、自己制御路を考えることが重要となってくる。文献(1)では、次の仮定を設けることにより、扱う拡張マークグラフの構造を簡単にしている。

＜仮定2.2＞

各最小デッドロックに対して、各排出トランジションは高々一つの自己制御路を持つ。□



(a) A deadlock



(b) A dendroid of (a)

図2.6　自己制御路をもつデッドロック

Fig. 2.6　A self-controlled deadlock

本章では、仮定2.2が満足されないような、より一般的な場合を考える。

さて、デッドロックDのデンドロイドRに存在する有向閉路C_Rを考えよう。一般に、初期マーキングMoにおいて、

$$\sum_{PL \in C_R} \sum_{p \in PL} Mo(p) \leqq \sum_{Pc \in C_R} \sum_{p \in Pc} Mo(p) \qquad (2.5)$$

が満足されないとき、任意のマーキング$M \in R(Mo)$において式(2.5)は満足さ

れない。逆に、Moにおいて式(2.5)が満足されるならば、任意のマーキングM∈R(Mo)において式(2.5)は満足される。この性質を利用して考えるが、文献(1)では、更に次の仮定が設けられている。

<仮定2.3>

(i) 各プレースp∈Pc＼PLに対して、次式を満足する。

$$|{}^{\bullet}p| = |p^{\bullet}| = 1 \qquad (2.6)$$

(ii) 異なる最小デッドロックに属するプレースは存在しない。

(iii) 各最小デッドロックに対して、任意の排出トランジション間に有向路は存在しない。 □

仮定2.3が満足されているとき、不等式(2.5)を満足すれば式(2.5)の左辺を零とするようなマーキングM∈R(Mo)が存在することを示すことができる。また、不等式(2.5)が満足されているとき、デンドロイドの触手枝に存在するPLのトークンをすべてデッドロックDから排出することができることを示すことができる。したがって、次の性質が得られる。

[性質2.3](1)

仮定2.1，仮定2.2および仮定2.3が満足されているものとする。このとき、拡張マークグラフがMoにおいてライブであるための必要十分条件は、次の条件を満足することである。

(H－1)

初期マーキングMoにおいて空のデッドロックは存在しない。

(H－2)

各最小デッドロックDに対して、次の条件のうち少なくともどちらか一方が満足される。

(i) DはMoにおいてトラップ条件を満足する。言換えると、MoにおいてDにはトークンが空のトラップは存在しない。

(ii) M_0において不等式(2.5)を満足しないDのデンドロイドRの有向閉路C_Rが少なくとも一つ存在する。 □

文献(1)では、デッドロックがトラップ条件を満足しない場合、デッドロックはトラップを含まないことを仮定している。

2．4 ライブであることを保証できるクラスの拡張

本節では、拡張マークグラフの構造的性質をもっと詳しく調べることにより、文献(1)で扱われたものより広いクラスの拡張マークグラフについてライブであるための必要十分条件を導出する。特に、仮定2.2を取除くことができること、および仮定2.3の制限を緩めることができることを示す。

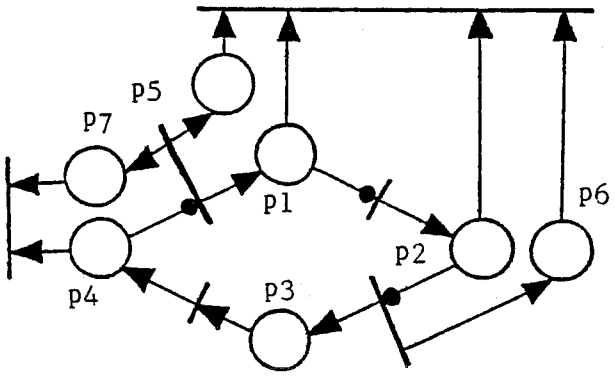
マークグラフにおいて有向閉路はデッドロック（かつトラップ）と等価であり、デッドロック（有向閉路）内に存在するトークンの和は不変であることがよく知られている。ところが、拡張マークグラフにおいてこの不変性は満足されない。しかし、デンドロイドRの有向閉路C_RにおいてPcに存在するトークンの和とPLに存在するトークンの和との差は不変である。つまり、任意のマーキング$M \in R(M_0)$に対して、次式を満足する。

$$\sum_{p \in Pc} M_0(p) - \sum_{p \in PL} M_0(p) = 一定 \qquad (2.7)$$

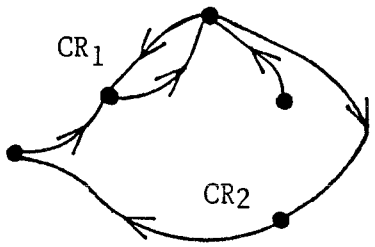
この拡張マークグラフのトークンの不変性を利用することにより、仮定2.2のような構造的制限を取除くことができる。例えば、図2.7のような仮定2.2を満足しない拡張マークグラフを考えよう。

Rの各有向閉路C_Rに対して、拡張マークグラフのトークンの不変性が成り立つ。つまり、次式を満足する。

$$\begin{cases} M(p_5) - M(p_1) = 一定 \\ M(p_6) + M(p_7) - \{M(p_1) + M(p_3) + M(p_4)\} = 一定 \\ \forall M \in R(M_0) \end{cases} \qquad (2.8)$$



(a) A deadlock D



(b) A dendroid R of D

図2.7　仮定2.2を満足しない最小デッドロック

Fig. 2.7　A minimal deadlock which does not satisfy Assumption 2.2

すべての有向閉路C_Rに対して、不等式(2.5)が満足されている場合を考える。つまり、次式を満足している。

$$\begin{cases} Mo(p_5) \geqq Mo(p_1) \\ Mo(p_6)+Mo(p_7) \geqq Mo(p_1)+Mo(p_3)+Mo(p_4) \end{cases} \qquad (2.9)$$

ここで、ネットがライブであると仮定しよう。このとき、不等式(2.5)を満足

すれば、各排出トランジションt₀∈Toに対して次式を満足するマーキングM∈R(Mo)が存在することを意味する。

$$\sum_{p\in PL} Mo(p) \leqq \sum_{p\in Pc} Mo(p) \qquad (2.10)$$

ただし、PLとPcはt₀へ入力する有向路である。不等式(2.10)を満足すれば、最小デッドロックが空となるマーキングM'∈R(Mo)が存在する。性質2.2から、ネットがライブであることに矛盾する。つまり、ライブであるためには仮定2.3の(H-2)(ii)が必要であることがわかる。

不等式(2.10)を満足するマーキングが存在することを示す。まず、排出トランジションの入力プレースの中で最小デッドロックDに存在するプレースの集合をSoで表す。"●"枝を"↑"枝とみなしたときにDに存在する有向閉路の集合をDcで表す。必要十分条件を導出するために、まず、補題2.1と補題2.2を導く。補題2.1は任意のプレースp∈Soに対して、M(p)≠0となるマーキングM∈R(Mo)が存在することを保証するものである。

仮定2.3の(ii)と性質2.3の(H-1)が満足される場合を考える。任意のマーキングM∈R(Mo)に対して発火可能とならないトランジション$\tilde{t}$∈Dc存在するならば、任意のM∈R(Mo)に対してトークンが空であるプレースp∈$^{\bullet}\tilde{t}$が存在する。このとき、任意のM∈R(Mo)に対してt₁∈$^{\bullet}$p＼p$^{\bullet}$は発火可能でない。性質2.1から、トークンが空であるプレースp₁∈$^{\bullet}$t₁が存在する。同様にして、任意のM∈R(Mo)に対して、発火可能でないトランジション　tₖ∈$^{\bullet}$pₖ₋₁＼p$\dot{}$ₖ₋₁, k=2,3,…, を得る。トランジションの数は有限であることから、tₖ=tとなるkが存在する。このとき、D*={p,p₁,p₂,…,pₖ₋₁}は最小デッドロックであり、トークンは空である。これは(H-1)に矛盾する。したがって、排出トランジションを発火することなく、Dcに存在する各トランジションtを発火可能とすることができる。つまり、各プレースp∈Soに対して、トランジションt∈$^{\bullet}$p∩Dcを発火可能とすることができる。仮定2.3の(ii)より、任意の排出トランジションからDcに存在するトランジションへの有向

路は存在しない。このことから、排出トランジションを発火することなく、各トランジションt∈•p∩Dcを同時に発火可能とすることができる（図2.8参照）。

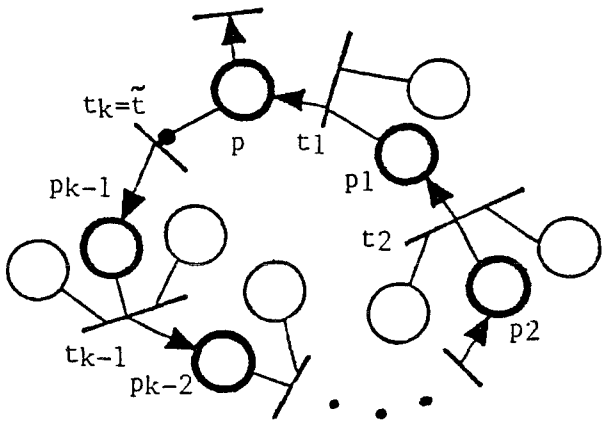


図2.8　証明の概略（補題2.1）

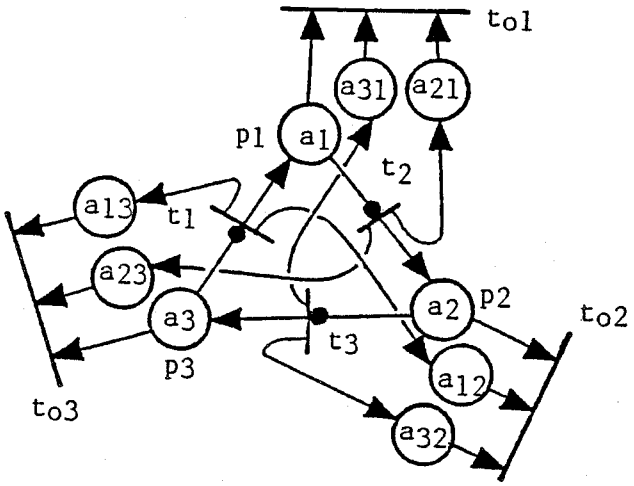
Fig. 2.8　An outline of the proof (Lemma 2.1)

したがって、次の性質が得られる。

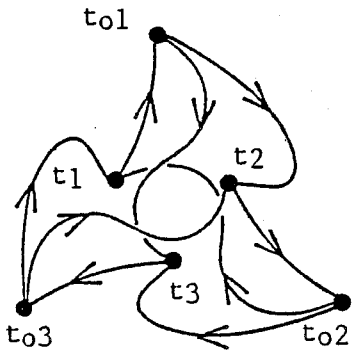
[補題2.1]

仮定2.1，仮定2.3の(ii)および(H-1)が満足されているものとする。このとき、各プレースに対して、M(p)≠0を満足する発火系列とマーキングM ∈ R(Mo) が存在する。 □

さて、すべてのプレースp∈Soにトークンが存在するようなマーキングMが得られたとしよう。このとき、各被制御トランジションt∈Tbはマーキング Mにおいて任意に発火することができる。マーキングMが得られたとして、不等式(2.5)が成立しているならば、不等式(2.10)を満足するマーキングが存在することを示す。図2.9のネットを例にして考える。



(a) A minimal deadlock



(b) A dendroid of (a)

図2.9　不等式(2.5)を満足する最小デッドロック

Fig. 2.9　A minimal deadlock satisfying inequalities (2.5)

図2.9において、So={p_1,p_2,p_3}, Tb={t_1,t_2,t_3}, To={t_{o1},t_{o2},t_{o3}}であり、a_{ij}はマーキングMにおけるトークンの数を表す。不等式(2.5)は次式で表される。

$$\begin{cases} a_{12}+a_{21} \leqq a_1+a_2 \\ a_{13}+a_{31} \leqq a_1+a_3 \\ a_{23}+a_{32} \leqq a_2+a_3 \end{cases} \tag{2.11}$$

$$\begin{cases} a_{12}+a_{23}+a_{31} \leqq a_1+a_2+a_3 \\ a_{13}+a_{32}+a_{21} \leqq a_1+a_3+a_2 \end{cases} \tag{2.12}$$

更に、不等式(2.11)と(2.12)は次のように書換えることができる。

$$\begin{cases} a_1-a_{21} \leqq -(a_2-a_{12}) \\ a_1-a_{31} \leqq -(a_3-a_{13}) \\ a_2-a_{32} \leqq -(a_3-a_{23}) \end{cases} \tag{2.13}$$

$$\begin{cases} (a_1-a_{21}) \leqq -(a_2-a_{32})-(a_3-a_{13}) \\ (a_1-a_{31})+(a_3-a_{23}) \leqq -(a_2-a_{12}) \\ (a_1-a_{31}) \leqq -(a_2-a_{12})-(a_3-a_{23}) \\ (a_1-a_{21})+(a_2-a_{32}) \leqq -(a_3-a_{13}) \\ (a_2-a_{32}) \leqq -(a_1-a_{21})-(a_3-a_{13}) \\ (a_1-a_{31})+(a_2-a_{12}) \leqq -(a_3-a_{23}) \end{cases} \tag{2.14}$$

また、不等式(2.11)と(2.12)から次式を得る。

$$\begin{cases} -(a_2-a_{32})-(a_3-a_{13})-(a_1-a_{31})-(a_3-a_{23}) \\ =(a_{32}+a_{23}-a_3-a_2)+(a_{13}+a_{31}-a_1-a_3) \geqq 0 \\ -(a_2-a_{12})-(a_3-a_{23})-(a_1-a_{21})-(a_2-a_{32}) \\ =(a_{12}+a_{21}-a_1-a_2)+(a_{32}+a_{23}-a_3-a_2) \geqq 0 \\ -(a_1-a_{21})-(a_3-a_{13})-(a_1-a_{31})-(a_2-a_{12}) \\ =(a_{21}+a_{12}-a_2-a_1)+(a_{13}+a_{31}-a_1-a_3) \geqq 0 \end{cases} \tag{2.15}$$

上の式から、次式を満足する非負整数k_1, k_2, k_3が存在することがわかる。

$$
\begin{cases}
a_1 - a_{21} \leqq k_2 - k_1 \leqq -(a_2 - a_{12}) \\
a_1 - a_{31} \leqq k_3 - k_1 \leqq -(a_3 - a_{13}) \\
a_2 - a_{32} \leqq k_3 - k_2 \leqq -(a_3 - a_{23}) \\
(a_1 - a_{21}) \leqq k_2 - k_1 \leqq -(a_2 - a_{32}) - (a_3 - a_{13}) \\
(a_1 - a_{31}) + (a_3 - a_{23}) \leqq k_2 - k_1 \leqq -(a_2 - a_{12}) \\
(a_1 - a_{31}) \leqq k_3 - k_1 \leqq -(a_2 - a_{12}) - (a_3 - a_{23}) \\
(a_1 - a_{21}) + (a_2 - a_{32}) \leqq k_3 - k_1 \leqq -(a_3 - a_{13}) \\
(a_2 - a_{32}) \leqq k_3 - k_2 \leqq -(a_1 - a_{21}) - (a_3 - a_{13}) \\
(a_1 - a_{31}) + (a_2 - a_{12}) \leqq k_3 - k_2 \leqq -(a_3 - a_{23})
\end{cases}
\qquad (2.16)
$$

不等式(2.16)から、被制御トランジションt_iをk_i(i=1,2,3)回発火すれば、各排出トランジションに対して不等式(2.10)を満足することがわかる。つまり、図2.9の最小デッドロックのトークンを空にすることができる。

上の議論を繰返せば、補題2.2のように一般的な場合にも同じことがいえる。証明は後節に示す。

[補題2.2]

(H-3)が満足されれば(H-4)が満足される。

（Ｈ－３）

任意の$m \in \{1,2,\cdots,n\}$に対して、$\{1,\cdots,m\}$の各組合わせ$\{i1,\cdots,im\}$において次式を満足する。

$$
a_{i1i2} + a_{i2i3} + \cdots + a_{imi1} \geqq a_{i1} + a_{i2} + \cdots + a_{im} \qquad (2.17)
$$

ただし、ａは非負整数である。

（Ｈ－４）

すべてのｉ，ｊ$\in \{1,\cdots,n\}$に対して、次式を満足する非負整数k_i, k_jが存在する。

$$
a_j - a_{ij} \leqq k_i - k_j \leqq -(a_i - a_{ji}),\ i \geqq j \qquad (2.18)\ \square
$$

＜注2.2＞

不等式(2.17)の左辺は、デンドロイドRの有向閉路C_Rに存在する自己制御路のトークンの総和と見ることができる。同様に、不等式(2.17)の右辺は、有向閉路C_Rに存在する誘導路のトークンの総和と見ることができる。一方、不等式(2.18)は、C_R内に存在するPcのトークンの総和からのPLのトークンの総和を引いたものが非負であることを意味している。もし、a_{ij}に相当する自己制御路が存在しない場合は、a_{ij}を∞と考えれば補題2.2が成立つ。　□

上の議論から、仮定2.2が成立たない場合、ネットがライブであるためには条件(H-2)の(ii)が必要であることがわかる。

次に、図2.10に示すように最小デッドロックにトラップが存在する場合を考える。特に、初期マーキングにおいてトラップにトークンが存在しない場合を考える（文献(1)では扱われていない）。

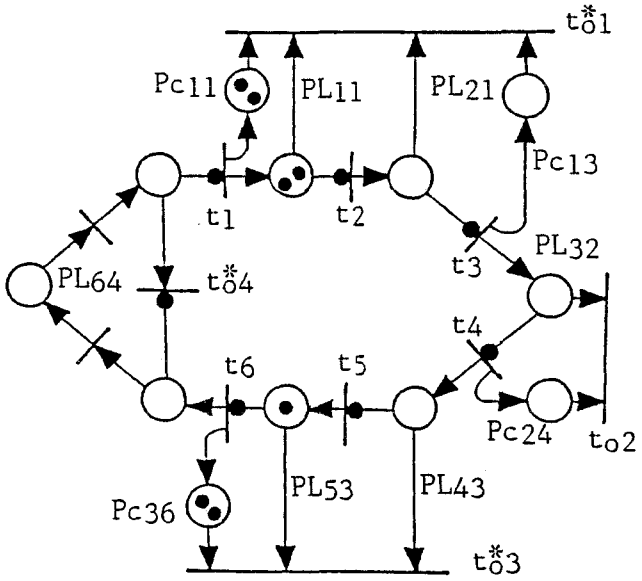
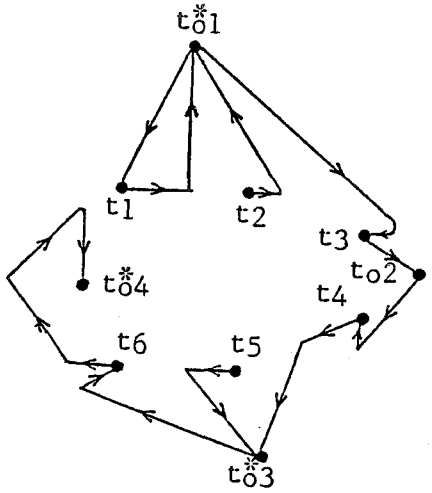


図2.10(a) A minimal deadlock



(b) A dendroid of (a)

図2.10　トラップを持つ最小デッドロック

Fig. 2.10　A minimal deadlock containing a trap

Dcに存在するトランジションtb∈Tbを制御枝の方向（"↓"を"↑"と同一視する）に添って順にt1, t2,…, tnと呼ぶことにする。排出トランジションも同様に、制御枝の方向に添って順にto1, to2,…, tomと呼ぶことにする。ただし、PL11をt1からto1への有向路に存在する誘導路とする。同様にして、すべての誘導路と自己制御路をそれぞれtk, tokの番号に合せてPLki, Pcjkと呼ぶ。Dに存在するトラップから排出トランジションへ伸びる有向路のうち、制御枝の方向に添って順に見ていったときに最初にトークンが存在する有向路をPeで表す。図2.10において、Pe={PL11}である。誘導路のうちそのプレースの集合がPeのプレースの集合に含まれる有向路をPL'で表す。図2.10において、PL'={PL11}である。Dに存在するトラップに接続するトランジションの集合をTrで表す。特に、ti∈Tr∪{Pe}からtor∈Toへの有向路が存在する場合torをtor*と表す。

図2.10のようにデッドロック内にトラップが存在する場合には、性質2.3の

(H-2)が満足されない場合でもネットがライブであるための十分条件を得ることができる。性質2.3の条件(H-1)と(H-2)が満足されない場合を考える。まず、次式 (2.19)が満足されない場合を考える。

$$\sum_{p\in PL_{qi}} Mo(p) = \max\{\sum_{p\in PL_{ki}} Mo(p)\} \qquad (2.19)$$
$$k$$

ただし、$q,k\in\{1,2,\cdots,n\}$である。このとき、同じ排出トランジションに接続するすべての誘導路のトークンの総和が互いに等しくなるような発火系列が存在するならば、トラップに接続するトランジションに発火しなければならないトランジションが存在する。たとえば、図2.10を考えよう。PL_{11}のトークンの総和がMoにおいてPL_{i1}のトークンの総和の最大値と等しくないとき、任意のマーキング$M\in R(Mo)$においてトラップTrに接続する被制御トランジションt_6を発火しないでPL_{11}のトークンの総和をPL_{i1}のトークンの総和の最大値と等しくすることはできない。t_6を発火すれば、トラップ条件が満足される。したがって、トランジションt_6を発火しないで、同じ排出トランジションに接続する誘導路に存在するトークンの総和を互いに等しくするような発火系列は存在しない。つまり、Dのトークンは空にならない。

次に、(2.19)式が成立しネットDがライブである場合を考える。同じ排出トランジションに接続する誘導路に存在するトークンの総和を互いに等しくするような被制御トランジションの発火回数k_iが存在し、(2.19)式がすべての排出トランジションついて成立っているならば、Dは空のデッドロックとなりDがライブであることに矛盾する。つまり、(2.19)式が成立していないこともまたネットがライブであるための必要条件である。したがって、次の定理を導くことができる。定理の中にあるアルゴリズム "PROCEDURE" を使用すれば、同じ排出トランジションに接続する誘導路に存在するトークンの総和を互いに等しくするような被制御トランジションの発火回数k_iを見つけることができる。

[定理2.1]

仮定2.1および仮定2.3が満足されているものとする。このとき、拡張マーク

グラフが初期マーキングMoにおいてライブであるための必要十分条件は、条件(H-1)および次の条件(H-5)が満足されることである。

（Ｈ－５）

各最小デッドロックＤに対して、次の条件の中で少なくとも一つを満足する。

(i) 性質2.3の(H-2)

(ii) 初期マーキングMoにおいて、次の条件の中で少なくとも一つを満足する。

(a) すべての$i \in \{1,\cdots,n\}$に対して、(2.19)式を満足しない$PL_{qi} \in \{PL'\}$が存在する。

(b) すべての$i,j,k,r \in \{1,\cdots,n\}$に対して、次のアルゴリズム "PROCEDURE" を用いて非負整数k_i, k_j, k_rを見つけることができない。

PROCEDURE

各 $i=1,\cdots,n$ に対して、次のステップを繰返す。ただし、nは排出トランジションの数である。

Step 1： $MAX_i := \max\{\Sigma Mo(PL_{ki})\}$

Step 2： $d_{ki} := MAX_i - \Sigma Mo(PL_{ki})$

Step 3： $d_{ki} > 0$かつ $MAX_m = 0$, $m \in \{1,2,\cdots\}$ならば、$MAX_m = 1$として Step 2 へ戻る。ただし、mは$PL_{k-1,m}$が存在する場合の添字である。

Step 4： $a_i := MAX_i$

$a_{ij} := \min\{d_{ki} + f_{jk}\}$, $f_{jk} = \Sigma Mo(Pc_{jk})$

Step 5： 次の条件を満足する非負整数k_i, k_jを見つける。

(i) $a_j - a_{ij} \leqq k_i - k_j \leqq -(a_i - a_{ji})$, $i \geqq j$

(ii) t_{or}^*が存在するすべてのrに対して、$k_r = 0$である。

(iii) すべてのqに対して、$k_i > 0$ならば$k_q + MAX_q > 0$を満足する。ただし、PL_{pq}と$PL_{p+1,i}$がそれぞれ排出トランジションt_{op}と$t_{o,p+1}$への誘導路である。

Step 6： ・ Step 5 でk_i, k_jを見つけることができれば "ネットはデッドである"

・ Step 5 でk_i, k_jを見つけることができなければ "ネットはライブである" □

<注2.3>

"PROCEDURE" は有限ステップで "ネットはデッドである" か "ネットはライブである" を出力する。 □

定理2.1の証明は後節において示す。定理2.1において、$t \in Tb$ から t_o への自己制御路が複数存在する場合には、自己制御路としてトークンの和が最小のものを選ぶ。

"PROCEDURE" が、定理2.1の必要性の証明においてどのように用いられるかを示す。拡張マークグラフがライブでかつ条件(H-5)を満足しないと仮定する。たとえば、図2.10のような拡張マークグラフである。条件(H-5)の(i)と(ii)(a)を満足しないことは明らかである。"PROCEDURE" を実行すると、次の関係式を得る。

Step 1 と Step 2 から；

$$\left\{\begin{array}{l} MAX_1 = 2 \\ \underline{MAX_2 = 0} \\ MAX_3 = 1 \\ MAX_4 = 0 \\ d_{11} = 2 - 2 = 0 \\ d_{21} = 2 - 0 = 2 \\ d_{32} = 0 - 0 = 0 \\ \underline{d_{43} = 1 - 0 = 1} \\ d_{53} = 1 - 1 = 0 \\ d_{64} = 0 - 0 = 0 \end{array}\right. \quad \begin{array}{c} \text{Step 3} \\ \Longrightarrow \end{array} \quad \left\{\begin{array}{l} MAX_1 = 2 \\ MAX_2 = \mathbf{1} \\ MAX_3 = 1 \\ MAX_4 = 0 \\ d_{11} = 2 - 2 = 0 \\ d_{21} = 2 - 0 = 2 \\ d_{32} = \mathbf{1} - 0 = \mathbf{1} \\ d_{43} = 1 - 0 = 1 \\ d_{53} = 1 - 1 = 0 \\ d_{64} = 0 - 0 = 0 \end{array}\right. \qquad (2.20)$$

Step 4 から；

$$\begin{cases} a_1 = MAX_1 = 2 \\ a_2 = MAX_2 = 1 \\ a_3 = MAX_3 = 1 \\ a_4 = MAX_4 = 0, \end{cases} \qquad \begin{cases} a_{21} = d_{32} + f_{13} = 1 + 0 = 1 \\ a_{32} = d_{43} + f_{34} = 1 + 0 = 1 \\ a_{43} = d_{64} + f_{36} = 0 + 2 = 2 \end{cases} \tag{2.21}$$

Step 5 から；

$$\begin{cases} k_1 = 0 \\ k_2 = 1 \\ k_3 = 1 \\ k_4 = 0 \end{cases} \text{ ただし、} \quad \begin{cases} a_{21} + k_2 \geqq a_1 + k_1 \\ a_{32} + k_3 \geqq a_2 + k_2 \\ a_{43} + k_4 \geqq a_3 + k_3 \end{cases} \tag{2.22}$$

このとき、被制御トランジションt_iを$k_j + d_{ij}$回発火すると、すべての排出トランジションに対して不等式(2.10)を満足するようにできる。そして、Dのトークンを空にする発火系列が存在する。この例では、$t_1 : k_1 + d_{11} = 0$，$t_2 : k_1 + d_{21} = 2$，$t_3 : k_2 + d_{32} = 2$，$t_4 : k_3 + d_{43} = 2$，$t_5 : k_3 + d_{53} = 1$，$t_6 : k_4 + d_{64} = 0$，であり、ネットはライブでないことがわかる。

各最小デッドロックがトラップを含まないならば、次の系が得られる。

[系2.1]
仮定2.1および仮定2.3が満足されているものとする。このとき、拡張マークグラフが初期マーキングM_0においてライブであるための必要十分条件は、条件(H-1)および条件(H-2)の(ii)が満足されることである。 □

次に、仮定2.3の必要性を考える。図2.2のネットのように、仮定2.3の(iii)が満足されない場合を考える。排出トランジション$t_{oi} \in T_0$から$t_{oj} \in T_0$への有向路は、$t_{oj} \in T_0$への自己制御路の一部と考えることができる。条件(H-

1)および条件(H-5)が満足されるならば、定理2.1の証明と同様にして、ライブであることが示される。

逆に、このような拡張マークグラフがライブでかつ条件(H-1)および条件(H-5)を満足しない場合を考える。このとき、トラップに接続する任意のトランジションを発火することなく、不等式(2.10)を満足するマーキングへ到達可能な発火系列が存在することを示すことができる。

仮定2.3の(iii)が満足されない場合、次の三つの場合が考えられる。

(i) Case 1: 排出トランジション間に有向路が存在しない。

(ii) Case 2: 排出トランジション間に有向路が存在する(たとえば、図2.2のt_{o1}とt_{o2})。

(iii) Case 3: 少なくとも二つの排出トランジションを含む有向閉路が存在する(図2.2参照)。

すべての場合において、不等式(2.10)から、各誘導路のトークンの和を零にする発火系列を見つけることができる。つまり、Dのトークンを空にすることができる。仮定2.3の(i)と(ii)が満足されるという仮定を仮定2.3'と呼ぶことにする。

上の議論から、次の定理が得られる。

[定理2.2]

仮定2.1および仮定2.3'が満足されているものとする。このとき、拡張マークグラフが初期マーキングMoにおいてライブであるための必要十分条件は、条件(H-1)および条件(H-5)が満足されることである。 □

各最小デッドロックがトラップを含まないならば、次の系が得られる。

[系2.2]

仮定2.1および仮定2.3'が満足されているものとする。このとき、拡張マークグラフが初期マーキングMoにおいてライブであるための必要十分条件は、条件(H-1)および条件(H-2)の(ii)が満足されることである。 □

２．５　例題

図2.3に示したような大規模システムの設計においては、実行不可能なタスクを含むというような、設計者の意図に反する欠陥を含んでしまうことがしばしば起こる。このような場合にペトリネット（拡張マークグラフ）を用いれば、シミュレーションすることなしにこのような欠陥を発見することができる。図2.3の例では、集合{p1,p2,…,p5}は条件(H-2)を満足しないデッドロックである。したがって、系2.2から、このネットはライブでないことがわかる。事実、次のようにタスクを実行するとシステムは実行不可能（デッドロック）となる。これは、発火系列 $t_1t_2t_3t_4t_5t_6t_7$ にしたがって発火することに相当する。

(1) タスクＢの処理終了、
(2) タスクＤ到着、
(3) タスクＡ到着、
(4) タスクＤの処理開始、
(5) タスクＡの処理開始、
(6) タスクＡの処理終了、
(7) タスクＡとタスクＢを組合わせる。

２．６　定理の証明

本節では、前節において示された定理の証明をする。まず、補題2.2の証明をする。

[補題2.2の証明]

不等式(2.17)から、任意の $m \in \{1,\cdots,n\}$ に対して、次式を満足する。

$$\begin{aligned} & a_y - a_{xy} \\ & \leqq a_{i_1 i_2} + \cdots + a_{i_{q-1},x} + a_{y,i_{q+2}} + \cdots + a_{i_r i_1} \\ & \qquad -(a_{i_1} + \cdots + a_{i_{q-1}} + a_x + a_{i_{q+2}} + \cdots + a_{i_r}) \end{aligned} \quad (2.23a)$$

$$
\begin{aligned}
&-(a_x - a_{yx}) \\
&\quad \geqq -(a_{\tilde{i}_1\tilde{i}_2} + \cdots + a_{\tilde{i}_{u-1},y} + a_{x,\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_r\tilde{i}_1}) \\
&\qquad + a_{\tilde{i}_1} + \cdots + a_{\tilde{i}_{u-1}} + a_y + a_{\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_v} \qquad (2.23b)
\end{aligned}
$$

$$
a_y - a_{xy} \leqq -(a_x - a_{yx}) \qquad (2.24)
$$

ただし、$x,y \in \{1,\cdots,n\}$であり、$\{i1,\cdots,ir\}$と$\{\tilde{i}1,\cdots,\tilde{i}v\}$は$\{i1,\cdots,im\}$の部分集合である。更に、不等式(2.17)から、次式を満足する。

$$
\begin{aligned}
&a_{i_1i_2} + \cdots + a_{i_{q-1},x} + a_{y,i_{q+2}} + \cdots + a_{i_ri_1} \\
&\quad -(a_{i_1} + \cdots + a_{i_{q-1}} + a_x + a_{i_{q+2}} + \cdots + a_{i_r}) \\
&\qquad + a_{\tilde{i}_1\tilde{i}_2} + \cdots + a_{\tilde{i}_{u-1},y} + a_{x,\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_r\tilde{i}_1} \\
&\qquad\quad -(a_{\tilde{i}_1} + \cdots + a_{\tilde{i}_{u-1}} + a_y + a_{\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_v}) \\
&= a_{\tilde{i}_1\tilde{i}_2} + \cdots + a_{\tilde{i}_{q-1},x} + a_{y,i_{q+2}} + \cdots + a_{i_ri_1} \\
&\quad + a_{i_1i_2} + \cdots + a_{i_{u-1},y} + a_{x,\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_r\tilde{i}_1} \\
&\qquad -(a_{\tilde{i}_1} + \cdots + a_{\tilde{i}_{q-1}} + a_x + a_{i_{q+2}} + \cdots + a_{i_r} \\
&\qquad\quad + a_{i_1} + \cdots + a_{i_{u-1}} + a_y + a_{\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_v}) \qquad (2.25)
\end{aligned}
$$

このとき、次式を満足する非負整数k_x, k_yが存在する。

$$
\begin{aligned}
&a_y - a_{xy} \\
&\leqq \max[a_y - a_{xy}, \{a_{i_1i_2} + \cdots + a_{i_{q-1},x} + a_{y,i_{q+2}} + \cdots + a_{i_ri_1} \\
&\qquad\qquad -(a_{i_1} + \cdots + a_{i_{q-1}} + a_x + a_{i_{q+2}} + \cdots + a_{i_r})\}] \\
&\leqq k_x - k_y \\
&\leqq \min[-(a_x - a_{yx}), \{-(a_{\tilde{i}_1\tilde{i}_2} + \cdots + a_{\tilde{i}_{u-1},y} + a_{x,\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_r\tilde{i}_1}) \\
&\qquad\qquad + a_{\tilde{i}_1} + \cdots + a_{\tilde{i}_{u-1}} + a_y + a_{\tilde{i}_{u+2}} + \cdots + a_{\tilde{i}_v}\}] \\
&\leqq -(a_x - a_{yx}) \qquad (2.26)
\end{aligned}
$$

＜証明終＞

次に、定理2.1の十分性の証明をする。

[定理2.1の十分性の証明]

まず、条件(H-1)と条件(H-2)の(i)を満足する最小デッドロックDを考えよう。トラップ条件の定義から、Dは任意のマーキングM∈R(Mo)においてトークンが空にならない。

次に、条件(H-1)と条件(H-2)の(ii)を満足するが条件(H-2)の(i)を満足しない最小デッドロックDを考えよう。このとき、DのデンドロイドRの有向閉路C_Rに存在する誘導路PLと自己制御路Pcのトークン和は、任意の$k \in \{\pm 1, \pm 2, \cdots\}$に対して次式を満足する。

$$\sum_{PL \in C_R} \sum_{p \in PL} Mo(p) + k > \sum_{Pc \in C_R} \sum_{p \in Pc} Mo(p) + k \geq 0 \qquad (2.27)$$

ただし、kは排出トランジションを発火したときは負の値、被制御トランジションを発火したときは正の値をとる。したがって、任意のマーキングM∈R(Mo)においてトークンが空になるデッドロックは存在しないことがわかる。

最後に、条件(H-1)と条件(H-5)の(ii)を満足するが条件(H-5)の(i)を満足しない最小デッドロックDを考えよう。条件(ii)の(a)を満足するとき、トラップに接続するトランジションt_qを発火することなしに、同じ排出トランジションt_{oi}に至るすべての誘導路PL_{ki}のトークン和が等しくなるマーキングM∈R(Mo)は存在しない。つまり、Dのトークンを空にしようとすると、Dはトラップ条件を満足してしまう。したがって、Dは任意のマーキングM∈R(Mo)においてトークンが空にならない。

更に、条件(ii)の(a)を満足しないが条件(ii)の(b)を満足を満足する場合を考える。アルゴリズム"PROCEDURE" において、MAX_iは排出トランジションt_{oi}から排出することができるトークンの数と考えることができる。d_{ki}は、すべての誘導路PL_{ki}のトークン和をMAX_iと等しくするために被制御トランジションt_kを発火する回数である。Step 2, Step 3 において、各誘導路PL_{ki}のトークン和をMAX_iと等しくするために被制御トランジションt_kを発火可能にす

る。Step 4 において、すべての排出トランジションt$_{oi}$について、各誘導路PL$_{ki}$のトークン和をa$_i$（=MAX$_i$）と等しくする。a$_{ij}$はt$_k$においてPL$_{ki}$と接続する自己制御路Pc$_{jk}$のトークンの和の値を表している。Step 5 において、すべての自己制御路のトークンの和を同じ排出トランジションに接続する誘導路のトークンの和の値以上にする被制御トランジションの発火回数k$_i$, k$_j$ を見つける。(ii)のk$_r$が正のとき、Dにおいて不等式(2.10)を満足させるためにはトラップに接続するトランジションを発火しなければならない。(iii)のk$_i$が正でかつk$_q$+MAX$_q$=0を満足するとき、誘導路PL$_{pi}$に存在する被制御トランジションt$_p$は発火可能でない。(i)にあるk$_i$とk$_j$を見つけることができないとき、任意のマーキングM∈R(Mo)において不等式(2.10)を満足しない排出トランジションが存在する。　＜証明終＞

最後に、定理2.1の必要性の証明をする。

［定理2.1の必要性の証明］

拡張マークグラフが初期マーキングMoにおいてライブであり、条件(H-1)と条件(H-5)のうち少なくともどちらか一方を満足しないと仮定する。

まず、条件(H-1)を満足しないとき、Moにおいてトークンが空のデッドロックが存在する。このとき、性質2.2から、拡張マークグラフはライブでない。

次に、条件(H-1)を満足するが条件(H-5)を満足しない場合を考える。条件(H-5)の(i)を満足しないとき、アルゴリズム "PROCEDURE" を用いてk$_i$, k$_j$を見つけることができる。各誘導路PL$_{ij}$に存在する被制御トランジションt$_i$をk$_j$+d$_{ij}$回発火することにより、トラップに接続するトランジションを発火することなく、すべての排出トランジションに関して不等式(2.10)を満足させることができる。更に、各排出トランジションをa$_i$回発火することができるので、Dのトークンを空にすることができる。このとき、性質2.2から、拡張マークグラフはライブでない。

以上のことは、最初の仮定に矛盾する。　＜証明終＞

２．７　結言

従来の拡張マークグラフに対する制限は厳しすぎ、モデル化能力はあまり高いとは言えなかった。そこで、本章では、文献(1)で論議されている拡張マークグラフよりも広いクラスの拡張マークグラフについて、ライブであるための必要十分条件を導出した。特に、ネット構造と初期マーキングに関する条件として導出した。たとえば、従来得られていた条件では図2.2や図2.3に示すネットのライブ性を調べることは困難であった。また、本章で導いた定理を用いれば、図2.3の例のようにシミュレーションすることなしに実行不可能なタスクを発見することができる。

第３章　ネットモデルによる
システムの階層的設計

３．１　緒言

通信プロトコル、ＦＡ、ＯＡシステム、シーケンス制御系などのように、複数の離散的事象が非同期・並列に進行する大規模な離散事象システムに対して、その体系的な設計手法の確立が望まれている。また、大規模システムの設計においては、いくつかのサブシステムを組合わせて一つの統合されたシステムを構成する方法や、システムの大枠から設計を行い各部分を順次具体化してゆく方法がとられる。このような手法において、柔軟かつ迅速な設計の自動化を図るためには、各設計段階における仕様記述・解析手法の形式化・体系化が重要である。

ペトリネット(1)は離散事象システムの動作解析のためのモデルとしてその有効性が広く認識されつつある。ペトリネットをシステムの仕様記述から動作解析・検証に至るまで一貫して用いることにより、上記の目的を達成する設計手法が構築し得るものと期待される。また、ペトリネットモデルの利点の一つは、トランジションをそのトランジションが表す動作を詳細表現した他のペトリネットで逐次置き換えることにより、トップダウン形式のシステム表現が可能となることである。つまり、従来からある階層的設計方法に適したモデルである。このような階層表現において問題となるのは、詳細化する前のペトリネットの動作が詳細化後のペトリネットにおいても完全に保存されているかどうかということである。逆に言うと、このような性質を保証する代置ネットの構造と初期トークンの配置を明らかにしておく必要がある。従来、単一トランジションの詳細化に関して上記の問題はValette(2)及び鈴木、村田(3)らによって考察され、逐次詳細化されるシステムのライブ性、有界性などの性質が、元のシステムの性質と代置ネットの性質を調べることにより検証し得ることが判っている。ところが、複数のトランジションを詳細化したとき、各々のサブネットの動作に相互関係が生ずる場合が多々あり、それらのサブネットを接続する必要が生ずる。換言すれば、階層化のレベルに従いシステム各部の相互干渉が陽に現れてくるのである。従って、上位レベルで互いに関連のありそうな複

数のトランジションをまとめて一つのネットで詳細化できれば、より効率的な階層表現が可能となる。

本章では、階層表現における層間での性質の保存性に関する文献(2)や(3)の結果を、システムの任意の部分（トランジションの集合）の詳細化に対して一般化する。3.3節においてその詳細化規則を示し、3.4節でネットの性質(有界性、ライブ性等)が保存されることを示す。また、3.5節においてこの階層表現によって大規模離散事象システムの動作検証が容易になることを示す。

３．２　記号と準備

本章で用いる諸記号及び諸定義を行い、定義から直ちに導くことができる性質を示す。特に、文献(3)との対比をしやすくするため他の章と少し異なる記号等を用いている箇所がある。

3.2.1　ペトリネット

ペトリネットは、プレース（○で表す）、トランジション（□で表す）と呼ばれる２種類の節点集合とそれらを結合する有向枝で構成される２部グラフである（図3.1）。通常、プレースは事象生起に関するシステムの条件を表し、トランジションは離散事象の生起と完了を表現する。この２部グラフはシステムの静的構造を表し、また、システムの動的挙動はトランジションを発火させ、プレース中に置かれるトークン（●で表す）を移動することによって表現される。

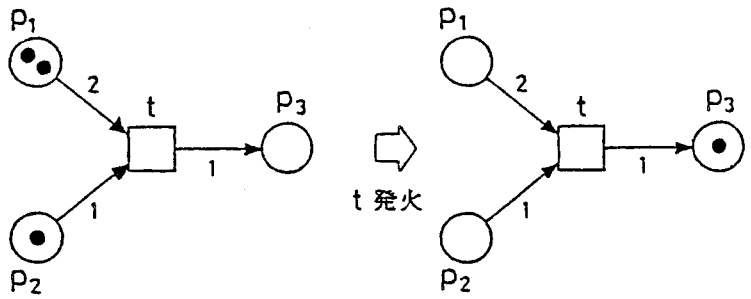


図3.1　ペトリネット

Fig. 3.1　Petri nets

具体的な定義をする前に、記号の定義をしておく。Ｚ⁺を非負整数，Ｓ*を集

合Sの要素から構成されるすべての系列の集合，$S^+ = S^* \setminus \{\lambda\}$とする。また、任意に大きな数$\omega$を$\omega > n$，$\omega \pm n = \omega$，$\omega \geqq \omega$，$\forall n \in Z$を満足する数とする。$\sigma \in S^*$，$s \in S$，$S' \subseteq S$とするとき、系列$\sigma$の長さを$|\sigma|$，系列$\sigma$上でsが現れる回数を$\#(\sigma, s)$，系列$\sigma$からS'の要素のみを取り出して構成される部分系列を$\delta(\sigma, S')$で表す。また、プレースの集合をP，トランジションの集合をT（$P \cap T = \phi$），IN：$T \times P \rightarrow Z^+$（トランジションの入力枝重み関数），OUT：$T \times P \rightarrow Z^+$（トランジションの出力枝重み関数），初期マーキングをM_0で表す。このとき、ペトリネットに関して次のように定義される。

[定義3.1]（ペトリネット）

ペトリネットNは、N＝（P，T，IN，OUT，M_0）の５項組で構成される。 □

[定義3.2]（発火条件）

ペトリネット$N = (P, T, IN, OUT, M_0)$において、$IN(t,p) \leqq M(p)$, $\forall p \in P$ を満足するとき、トランジションtはマーキングMで発火可能であるという。 □

[定義3.3]（マーキング遷移則）

マーキングM_1で発火可能なトランジションtが発火するとマーキングM_1は、

$$\forall p \in P ; M_2(p) = M_1(p) - IN(t,p) + OUT(t,p) \qquad (3.1)$$

で定まるマーキングM_2に遷移する。これを、$M_1[t>M_2$と表す。 □

[定義3.4]（可到達性）

マーキング$M_1, M_2, \cdots, M_{n+1}$に対し、$M_i[t_i>M_{i+1}$ $(i=1,2,\cdots,n)$となるトランジション$t_1, t_2, \cdots, t_n$が存在するとき、

$$\sigma = t_1 t_2 \cdots t_n \qquad (3.2)$$

を発火系列という。このとき、マーキングM_{n+1}はマーキングM_1から可到達であるといい、$M_1[\sigma>M_{n+1}$で表す。 □

ペトリネットの有界性、安全性、ライブ性などの諸性質は、システムを解析する上で重要な性質である。

ここで、集合R(N)をペトリネットNにおいて初期マーキングM_0から可到達なマーキングの集合，L(N)をペトリネットNにおいて初期マーキングM_0から始るすべての発火系列の集合として、次の定義をする。

[定義3.5]（有界性）

すべてのマーキング$M \in R(N)$に対して、$M(p) \leqq k$，$\forall p \in P$,を満足する $k \in Z^+$が存在するとき、ペトリネット$N=(P,T,IN,OUT,M_0)$は有界であるという。 □

[定義3.6]（安全性）

すべてのマーキング$M \in R(N)$に対して，$M(p) \leqq 1$，$\forall p \in P$，を満足するとき、ペトリネット$N=(P,T,IN,OUT,M_0)$は安全であるという。 □

[定義3.7]（ライブ性）

(1)任意の$\forall \sigma \in L(N)$に対して、トランジション$t \in T$が発火可能となる$\sigma' \in T^*$が存在するとき、tはライブであるという。

(2)ペトリネット$N=(P,T,IN,OUT,M_0)$に存在するすべてのトランジション$t \in T$がライブであるとき、ペトリネットNはライブであるという。 □

3.2.2 Well-behavedness

ペトリネット$N=(P,T,IN,OUT,M_0)$に対し、トランジション$t \in T$(プレース$p \in P$)から$p \in P$($t \in T$)への有向枝が存在するとき、tをpの入力(出力)トランジションと呼ぶ。更に、入力トランジションの集合を$T_{in}=\{t_{i\text{-}in} \mid 0 \leqq i \leqq n-1,\ i \in Z^+\}$ ($T_{in} \subset T$)及び出力トランジションの集合を$T_{out}=\{t_{i\text{-}out} \mid 0 \leqq i \leqq n-1,\ i \in Z^+\}$, ($T_{out} \subset T$, $T_{in} \cap T_{out} = \phi$)で表す(図3.2(a))。これらの入出力トランジションの間に、次に示す規則に従いある初期トークンを持ったテ

ストプレースの集合S＝{s_i | 0≦i≦n-1, i∈Z^+}を接続して得られるペトリネットをN_B＝(P_B,T_B,IN_B,OUT_B,M_{B0})とする(図3.2(b))。

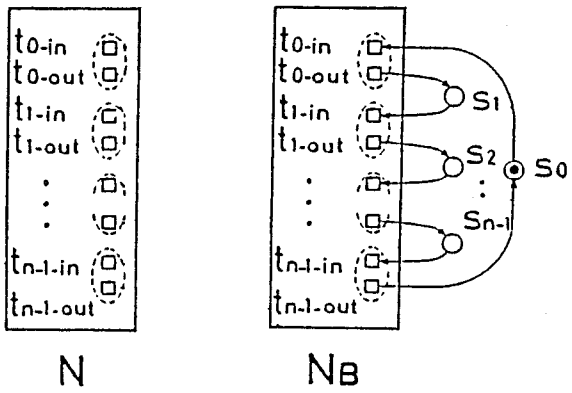


図3.2　N_Bの構成

Fig. 3.2　Compositions of N_B

[N_Bの構成規則]

N_B＝(P_B,T_B,IN_B,OUT_B,M_{B0})；

$$P_B = P \cup S, \tag{3.3}$$

$$T_B = T, \tag{3.4}$$

$$IN_B(t,p) = \begin{cases} 1 & ; t=t_{i\text{-}in} \text{ and } p=s_i \\ 0 & ; t=t_{i\text{-}in} \text{ and } p=s_j \ (i \neq j) \\ IN(t,p) & ; \text{otherwise}, \end{cases} \tag{3.5}$$

$$OUT_B(t,p) = \begin{cases} 1 & ; t=t_{i\text{-}out} \text{ and } p=s_{i+1} \ (s_n=s_0) \\ 0 & ; t=t_{i\text{-}out} \text{ and } p=s_j \ (j \neq i+1) \\ OUT(t,p) & ; \text{otherwise}, \end{cases} \tag{3.6}$$

$$M_{B0}(p) = \begin{cases} 1 & ; p=s_0 \\ 0 & ; p=s_i \ (i \neq 0) \\ M_0(p) & ; \text{otherwise}. \end{cases} \tag{3.7}$$

次に、入出力トランジションの集合T_{in},T_{out}が与えられたサブネットのT_{in}, T_{out}の発火順序に着目した性質であるwell-behavednessを以下のように定義する。

[定義3.8] (well-behavedness)

ペトリネット$N=(P,T,IN,OUT,M_0)$は、次の条件(1), (2)を満たすときトランジション$T_{in}\subset T$, $T_{out}\subset T$ ($T_{in}\cap T_{out}=\phi$)に関してwell-behavedであるという。

(1) $N_B=(P_B,T_B,IN_B,OUT_B,M_{B0})$において $t_{0\text{-}in}$ はライブである。

(2) すべての$\sigma\in L(N_B)$, i,j, に対して、次式を満足する。

$$\#(\sigma,t_{i\text{-}in})-\#(\sigma,t_{j\text{-}in})=1 \text{ or } 0,\ i<j \tag{3.8}$$

$$\#(\sigma,t_{i\text{-}out})-\#(\sigma,t_{j\text{-}out})=1 \text{ or } 0,\ i<j \tag{3.9}$$

$$\#(\sigma,t_{i\text{-}in})-\#(\sigma,t_{i\text{-}out})=1 \text{ or } 0 \tag{3.10}$$

このとき、直ちに次の性質が得られる。

[性質3.1]

ペトリネット$N=(P,T,IN,OUT,M_0)$がトランジション $T_{in}\subset T$, $T_{out}\subset T$ ($T_{in}\cap T_{out}=\phi$)に関してwell-behaved条件(2)を満たすことは、$T_{in}\cup T_{out}$の要素から構成される部分系列を$\delta(\sigma,T_{in}\cup T_{out})=a_0a_1a_2a_3\cdots a_{2k}a_{2k+1}$とするとき、すべての発火系列$\sigma\in L(N_B)$に対し、

$$a_{2k}=t_{k(\mathrm{mod}\ n)\text{-}in},\ a_{2k+1}=t_{k(\mathrm{mod}\ n)\text{-}out} \tag{3.11}$$

を満足することと等価である。 □

また、ライブの定義から、次の性質を得る。

[性質3.2]

ペトリネットN＝(P,T,IN,OUT,M_0)が、トランジションT$_{in}$⊂T,T$_{out}$⊂T (T$_{in}$∩T$_{out}$=φ)に関してwell-behavedであるならば、δ(σ,T$_{in}$∪T$_{out}$)が無限系列となる発火系列σ∈L(N_B)が存在する。 □

3．3　トランジションの詳細化

まず、トランジションを他のペトリネットで置き換えるときの規則を与える。

ペトリネットN＝(P,T,IN,OUT,M_0)中のトランジションの集合Ts＝{t_i | 0≦i≦n-1, i∈Z^+} (Ts⊂T)を、ペトリネットN'＝(P',T',IN',OUT',M_0')の入出力トランジション

$$T_{in}=\{t_{i\text{-}in} \mid 0\leqq i\leqq n-1,\ i\in Z^+\},\ T_{in}\subset T', \tag{3.12}$$

$$T_{out}=\{t_{i\text{-}out} \mid 0\leqq i\leqq n-1,\ i\in Z^+\},\ T_{out}\subset T',\ T_{in}\cap T_{out}=\phi \tag{3.13}$$

で置き換えることによって得られるネットをN"=(P",T",IN",OUT",M_0")とする。

[N"の構成規則]

N"＝(P",T",IN",OUT",M_0")；

$$P"=P\cup P', \tag{3.14}$$

$$T"=(T\cup T')\setminus Ts, \tag{3.15}$$

$$IN"(t,p)=\begin{cases} IN(t,p) ; & t\in T\setminus Ts \text{ and } p\in P,\\ IN'(t,p); & t\in T' \text{ and } p\in P',\\ IN(t_i,p); & t=t_{i\text{-}in} \text{ and } p\in P,\\ 0 ; & \text{otherwise,}\end{cases} \tag{3.16}$$

$$OUT"(t,p)=\begin{cases} OUT(t,p) ; & t\in T\setminus Ts \text{ and } p\in P,\\ OUT'(t,p); & t\in T' \text{and } p\in P',\\ OUT(t_i,p); & t=t_{i\text{-}out} \text{ and } p\in P,\\ 0 ; & \text{otherwise,}\end{cases} \tag{3.17}$$

$$M_0"(p)=\begin{cases} M_0(p) ; & p\in P,\\ M_0'(p) ; & p\in P'\end{cases} \tag{3.18}$$

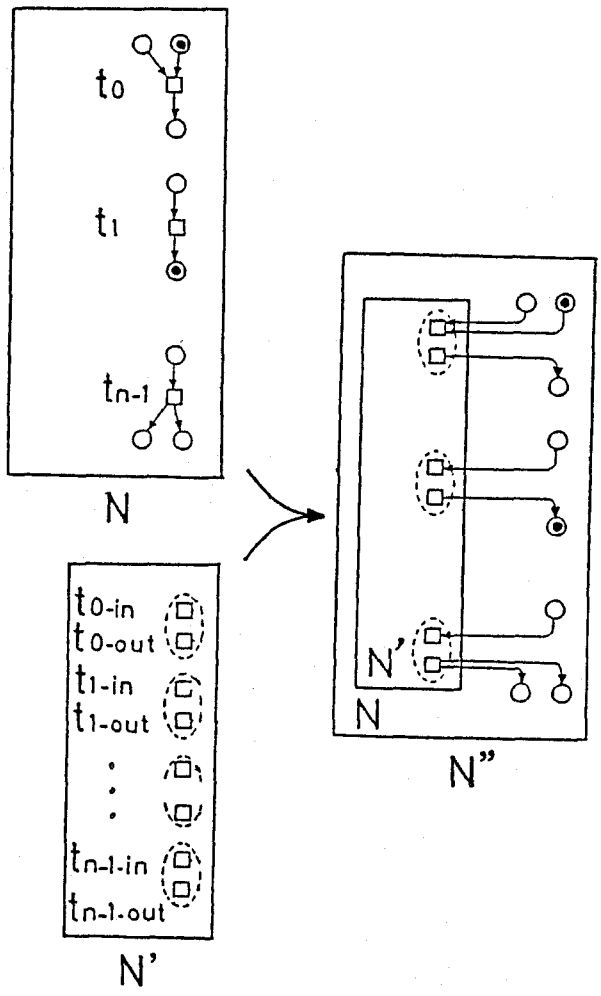


図3.3　ネットの詳細化

Fig. 3.3　Detail drawings of nets

以上の詳細化の規則に従ってシステムがどのように階層表現されるかを、ロボットアームの動作をモデル化することによって示す。図3.4は、同じ仕様を持つ２基のロボットアームが部品をＰ点からＳ点へ運ぶシステムである(4)。

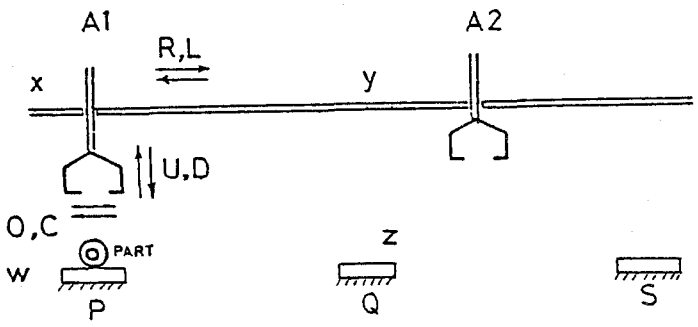


図3.4　ロボットアーム運搬システム

Fig. 3.4　Manipulator control systems

動作仕様は次のようなものである。

［動作仕様］

・Ｐ点に部品が置かれるとアームが作動し部品をＱ点へ運ぶ。

・Ｑ点に部品が置かれるとアームが作動し部品をＳ点へ運ぶ。

これをモデル化すると図3.5(a)になる。

次にトランジションA1、A2が表すアーム単体の動作を動作仕様に従って具体的にモデル化すると図3.5(b)になる。

［アーム１基の動作仕様(１サイクル)］

・初期状態…位置：ｘ，ハンド：開

・ｗ点に部品が置かれると下降(ｘ→ｗ)する。

・ハンドを閉じ部品をつかむ。

・ｗ→ｘ→ｙ→ｚの経路に従って移動する。

・ハンドを開き部品をｚ点に置く。

・ｚ→ｙ→ｘの経路に従って初期位置に戻る。

トランジションAlin, Aloutが、A1を詳細化するときの入力トランジション、出力トランジションとなる。A2についても同様である。詳細化したネット(図3.5(c))はアームの動作仕様を満足している。

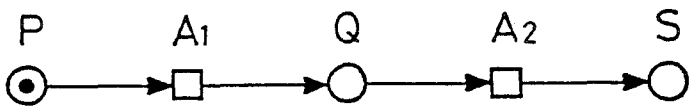


図3.5(a) A spesification of the manipulator control process

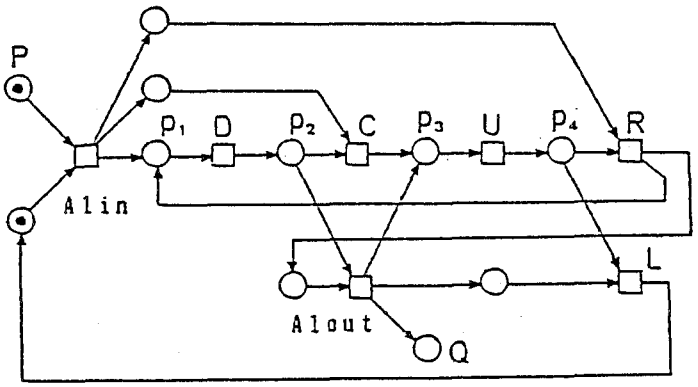
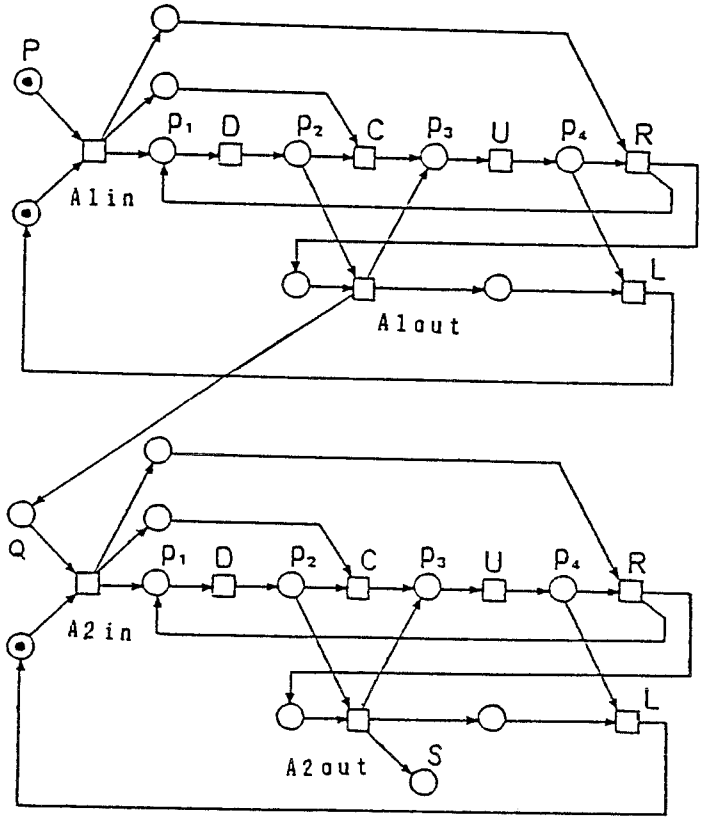


図3.5(b) N : One manipulator control process



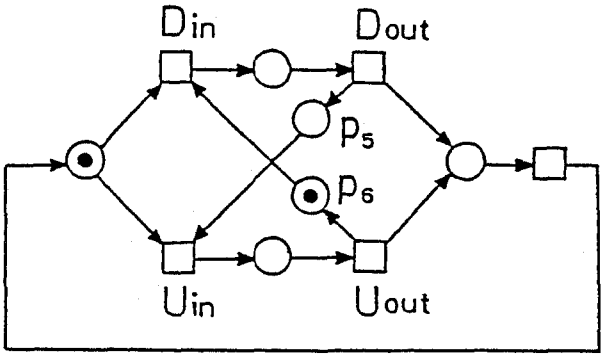
(c) Two manipulator control processes

図3.5　ロボットアームのモデル化

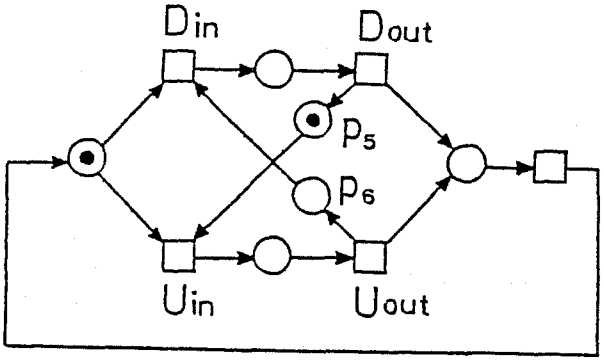
Fig. 3.5　Modelling of manipulator control systems

次に、モータの動作の部分を詳細化する場合を考える。トランジションU，Dが表すアームの昇降は1個のモータで駆動される。モータのモデルは図3.6(a),(b)が考えられる。共に単体では正常に動作するモータである。Uin,Dinをそれぞれ U,Dの入力トランジション、Uout,Doutを出力トランジションとして置き換えると図3.7(a),(b)になる。図3.7(a)はアーム1基の仕様を満足しているが、図3.7(b)はプレースp_6にトークンが置かれないためDinが発火で

きずデッドロックとなってしまう。即ち、元のネットのライブ性が詳細化後のネットでは保存されなくなる。従って、図3.7(b)のネットは代置するネットとして用いることが出来ないことになる。



(a) N'



(b) N'

図3.6 モータのモデル

Fig. 3.6 Modelling of motors

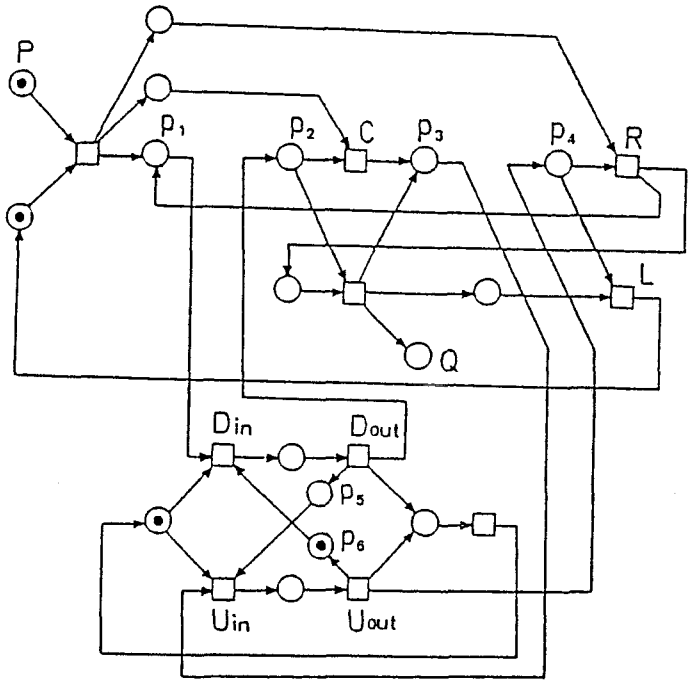
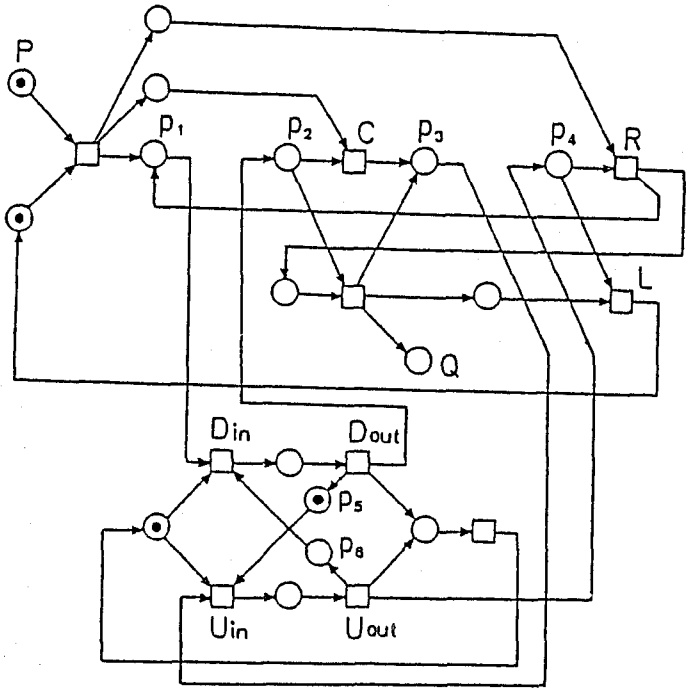


図3.7(a) N"：Live nets



(b) N" : A net having deadlocks

図3.7　詳細化したアームのモデル

Fig. 3.7　Detail drawings of manipulator control system models

これらの例からもわかるように、構造と初期マーキングの与え方によっては、元のネットでの諸性質が保存されないような詳細化が存在するのである。次節において、代置ネットがどのような条件を満足すれば元のネットでの諸性質が詳細化後も保存されるかという問題を考える。

３．４　ライブ性・有界性の保存

本節では、代置するネットをwell-behavedであるとし、詳細化を行うトランジションの集合が以下のような発火順序に関する条件を満足していれば、諸性質（ライブ性など）を保存したまま詳細化できることを示す。

［変換条件］

ペトリネットN＝(P,T,IN,OUT,M_0)の中の詳細化されるトランジションの集合Ts⊆Tを考える。このとき、任意の$\sigma \in L(N)$, i,j, に対して、次式を満足する。

$$\#(\sigma, t_i) - \#(\sigma, t_j) = 1 \text{ or } 0, \quad i<j \tag{3.19}$$

このとき、直ちに次の性質が得られる。

［性質3.3］

ペトリネットN＝(P,T,IN,OUT,M_0)の中のトランジションの集合$Ts=\{t_i \mid 0 \leq i \leq n-1, i \in Z^+\}$,(Ts⊆T)が変換条件を満たすことは、すべての発火系列$\sigma \in L(N)$に対しTsの要素から構成される部分系列を$\delta(\sigma, Ts) = a_0 a_1 \cdots a_k$とするとき$a_k = t_{k(\mathrm{mod}\ n)}$となることと等価である。 □

上述の規則に従って、ペトリネットNのトランジション集合Tsをサブネット N'で詳細化し新たにペトリネットN"を構成する。

まず、NとN"の間の性質の保存性に関する基本的な結果（定理3.1）を導くために、以下の補題3.1から補題3.6が成り立つことを示す。

N"のトランジション T"に関する発火系列をNのトランジションTに関する発火系列に変換する関数ϕ、およびN'のトランジションT'に関する発火系列に変換する関数ψを定義する。

［定義3.10］

(1)ϕ : (T")*→T*は次式を満足する。

(a) $\phi(\lambda) = \lambda$ (3.20)

(b)$\sigma" \in (T")^*$について

$$\phi(\sigma" t) = \begin{cases} \phi(\sigma") & ; t \in T' \setminus T_{in} \\ \phi(\sigma") t_i & ; t = t_{i\text{-}in} \in T_{in} \\ \phi(\sigma") t & ; t = T \setminus Ts \end{cases} \tag{3.21}$$

(2) ψ : $(T'')^* \rightarrow (T')^*$は次式を満足する。

(a) $\psi(\lambda) = \lambda$ (3.22)

(b) $\sigma'' \in (T'')^*$について

$$\psi(\sigma''t) = \begin{cases} \psi(\sigma'') & ; t \in T \setminus Ts \\ \psi(\sigma'')t & ; t \in T' \end{cases} \quad (3.23)$$

[補題3.1]

すべての$\sigma'' \in L(N'')$について、次式が成り立つ。

$$\delta(\sigma'', T_{in} \cup T_{out}) = a_0 a_1 a_2 a_3 \cdots a_{2k} a_{2k+1} \quad (3.24)$$

ただし、$a_{2k} = t_{k(\mathrm{mod}\ n)\text{-}in}$, $a_{2k+1} = t_{k(\mathrm{mod}\ n)\text{-}out}$である。 □

（証明）$\sigma'' = \lambda$のとき$\delta(\sigma'', T_{in} \cup T_{out}) = \lambda$, $|\sigma_1''| = d$となるすべての$\sigma_1'' \in L(N'')$について成立すると仮定する。

$\sigma_1''t \in L(N'')$とおくと $t \in (T \cup T') \setminus (Ts \cup T_{in} \cup T_{out})$のとき

$$\delta(\sigma_1''t, T_{in} \cup T_{out}) = \delta(\sigma_1'', T_{in} \cup T_{out}) \quad (3.25)$$

となり、成立する。

$t \in T_{in} \cup T_{out}$のとき、$|\delta(\sigma_1'', T_{in} \cup T_{out})| = 2k$、即ち$a_{2k} = t_{k(\mathrm{mod}\ n)\text{-}in}$のとき$t = t_{k(\mathrm{mod}\ n)\text{-}out}$であることを示す。

$t = t_{i\text{-}out}$ $(i \neq k(\mathrm{mod}\ n))$とすると、$N_B$において$t_{k(\mathrm{mod}\ n)\text{-}in}$の発火後、$t_{k(\mathrm{mod}\ n)\text{-}out}$を発火せずに他の出力トランジションが発火し得ることになる。これは、well-behaved条件に反する。

$t = t_{i\text{-}in}$ $(\exists i)$とすると、N''においてマーキングM_1から $t_{k(\mathrm{mod}\ n)}$の発火なしに、ある入力トランジションが発火可能となるならば、Nにおいて$t_{k(\mathrm{mod}\ n)}$を発火せずにトランジションt_iが発火可能となる。これは変換条件に反する。

$|\delta(\sigma_1'', T_{in} \cup T_{out})| = 2k+1$のときも同様。 <証明終>

[補題3.2]

$\sigma''\in L(N'')$は与えられているとする。

$\phi(\sigma'')\in L(N)$, $\psi(\sigma'')\in L(N_B)$, $M_0''[\sigma''>M_1''$, $M_0[\phi(\sigma'')>M_1$, $M_{B0}[\psi(\sigma'')>M_{B1}$とする。このとき、すべての $p\in P$, $p'\in P'$は

$$M_1(p)=M_1''(p)+\sum_i [OUT''(t_{i-out},p)\{\#(\sigma'',t_{i-in})-\#(\sigma'',t_{i-out})\}] \quad (3.26)$$

$$M_{B1}(p')=M_1''(p') \quad (3.27)$$

を満足する。 □

（証明）式(3.27)は明らか。式(3.26)について示す。

$\sigma''=\lambda$のとき式(3.26)は成立する。

$|\sigma_1''|=d$となるすべての$\sigma_1''\in L(N'')$について、任意の $p\in P$が次式を満足すると仮定する。

$$M(p)=M''(p)+\sum_i [OUT''(t_{i-out},p)\{\#(\sigma_1'',t_{i-in})-\#(\sigma_1'',t_{i-out})\}] \quad (3.28)$$

ただし、$M_0''[\sigma_1''>M''$, $M_0[\phi(\sigma_1'')>M$である。また、$\sigma''=\sigma_1''t\in L(N'')$, $M''[t>M_1''$, $M[\phi(t)>M_1$とおく。

$t\in T\setminus T_s$のとき、任意の $p\in P$は次式を満足する。

$$M_1''(p)=M''(p)-IN(t,p)+OUT(t,p) \quad (3.29)$$

$$\forall p\in P\ ;\ M_1(p)=M(p)-IN(t,p)+OUT(t,p) \quad (3.30)$$

上式を(3.28)式に代入し(3.26)式を得る。

$t\in T'\setminus(T_{in}\cup T_{out})$のとき、任意の $p\in P$は次式を満足する。

$$M_1''(p)=M''(p),\ M_1(p)=M(p) \quad (3.31)$$

上式を(3.28)式に代入し(3.26)式を得る。

$t=t_{j\text{-}in}\in T_{in}$のとき、任意の $p\in P$は次式を満足する。ただし、$\phi(t_{j\text{-}in}) = t_j$である。

$$
\begin{aligned}
M_1(p) &= M(p)-IN(t_j,p)+OUT(t_j,p)\\
&= M''(p)+\sum_i [OUT''(t_{i\text{-}out},p)\{\#(\sigma_1'',t_{i\text{-}in})-\#(\sigma_1'',t_{i\text{-}out})\}]\\
&\qquad -IN(t_j,p)+OUT(t_j,p)\\
&= M''(p)-IN''(t_{j\text{-}in},p)+OUT''(t_{j\text{-}out},p)\\
&\qquad +\sum_i [OUT''(t_{i\text{-}out},p)\{\#(\sigma_1'',t_{i\text{-}in})-\#(\sigma_1'',t_{i\text{-}out})\}]\\
&= M_1''(p)+\sum_i [OUT''(t_{i\text{-}out},p)\{\#(\sigma'',t_{i\text{-}in})\\
&\qquad -\#(\sigma'',t_{i\text{-}out})\}] \qquad (3.32)
\end{aligned}
$$

$t=t_{j\text{-}out}\in T_{out}$のとき、任意の $p\in P$は次式を満足する。ただし、$\phi(t_{j\text{-}out}) = \lambda$である。

$$
\begin{aligned}
M_1(p) &= M(p)\\
&= M''(p)+\sum_i [OUT''(t_{i\text{-}out},p)\{\#(\sigma_1'',t_{i\text{-}in})-\#(\sigma_1'',t_{i\text{-}out})\}]\\
&= M''(p)+OUT''(t_{j\text{-}out},p)\\
&\qquad +\sum_i [OUT''(t_{i\text{-}out},p)\{\#(\sigma'',t_{i\text{-}in})-\#(\sigma'',t_{i\text{-}out})\}]\\
&= M_1''(p) \qquad (3.33)
\end{aligned}
$$

したがって、結果を得る。　　<証明終>

[補題3.3]

すべての発火系列$\sigma\in L(N)$について、$\phi(\sigma'')=\sigma$となる発火系列$\sigma''\in L(N'')$が存在する。　□

（証明）性質3.1，性質3.2から、発火系列の集合$L(N_B)$には、次式を満足する無限発火系列$\sigma_0\sigma_1\sigma_2\sigma_3\cdots\ \sigma_{2k}\sigma_{2k+1}\cdots$が存在する。

$$\#(\sigma_{2k},t) = \begin{cases} 1 ; t=t_{k(\mathrm{mod}\ n)-in} \\ 0 ; t\in(T_{in}\cup T_{out})\setminus\{t_{k(\mathrm{mod}\ n)-in}\} \end{cases} \quad (3.34)$$

$$\#(\sigma_{2k+1},t) = \begin{cases} 1 ; t=t_{k(\mathrm{mod}\ n)-out} \\ 0 ; t\in(T_{in}\cup T_{out})\setminus\{t_{k(\mathrm{mod}\ n)-out}\} \end{cases} \quad (3.35)$$

性質3.3，N"の構成規則から、各々の発火系列$\sigma\in L(N)$においてk番目に現れるトランジション$t\in T_S$に部分系列$\sigma_k\sigma_{k+1}$を代入して得られる発火系列をσ"とすると、σ"$\in L(N")$である。また、明らかに$\phi(\sigma")=\sigma$である。　＜証明終＞

［補題3.4］

すべての発火系列$\sigma"\in L(N")$について、$\phi(\sigma")\in L(N)$である。　□

（証明）$\sigma"=\lambda$のとき、$\phi(\lambda)=\lambda$より明らか。

$|\sigma_1"|=d$のとき成立すると仮定する。

$\sigma"=\sigma_1"t\in L(N")$とおくと、 $t\in T'\setminus T_{in}$のとき、$\phi(\sigma")=\phi(\sigma_1")\in L(N)$となり成立する。

$t\in(T\setminus T_S)\cup T_{in}$のとき、$\phi(\sigma")=\phi(\sigma_1)t\in L(N)$となることを示す。ただし$t=t_{i-in}\in T_{in}$のとき$\phi(\sigma")=\phi(\sigma_1)t_i$である。

補題3.2から、任意の プレース $p\in P$は$M_1"(p)\leqq M_1(p)$を満足する(ただし、$M_0"[\sigma_1>M_1",M_0[\phi(\sigma_1)>M_1)$。

従って、N"の$M_1"$において発火可能なトランジションはNのM_1においても発火可能である。よって、$\phi(\sigma")=\phi(\sigma_1")t\in L(N)$となる。　＜証明終＞

［補題3.5］

すべての発火系列$\sigma"\in L(N")$について、$\psi(\sigma")\in L(N_B)$である。　□

(証明）$\sigma"=\lambda$のとき、$\psi(\lambda)=\lambda$から明らかである。

$|\sigma_1"|=n$のとき成立すると仮定する。

$\sigma"=\sigma_1"t\in L(N")$とおくと、$t\in T\setminus T_S$のとき、$\psi(\sigma_1"t)=\psi(\sigma_1")\in L(N_B)$となり成立する。

$t \in T'$のとき、補題3.2から、任意の$p \in P'$は$M_1"(p)=M_{B1}(p)$を満足する(ただし$M_0"[\sigma_1">M_1"$, $M_{B0}[\psi(\sigma_1")>M_{B1}$) 。
$t \in T' \setminus T_{in}$のとき、N"の$M_1$において発火可能なトランジションは$N_B$の$M_{B1}$においても発火可能となる。よって$\psi(\sigma_1"t)=\psi(\sigma_1")t \in L(N_B)$となる。
$t=t_{i-in} \in T_{in}$のとき、補題3.1から$M_{B1}(s_i)=1$となるので、N_BのM_{B1}においても発火可能となる。よって、$\psi(\sigma_1"t)=\psi(\sigma_1")t \in L(N_B)$となる。　<証明終>

[補題3.6]
ペトリネットNにおいて、$Ts \subseteq T$はライブであるとする。このとき、すべての発火系列$\sigma \in L(N_B)$について、$\psi(\sigma")=\sigma$となる$\sigma" \in L(N")$が存在する。 □

(証明) Tsに関する無限部分系列が L(N)に存在することより、L(N")に$T_{in} \cup T_{out}$に関する無限部分系列が存在する。よって、すべての発火系列$\sigma \in L(N_B)$について、$\psi(\sigma")=\sigma$となる$\sigma" \in L(N")$が存在する。　<証明終>

上述の補題3.1から補題3.6を用いると、ネットの詳細化に関して次の定理が得られる。

[定理3.1] (性質の保存性)
変換条件を満足するペトリネット$N=(P,T,IN,OUT,M_0)$のトランジションの集合$Ts \subseteq T$を、well-behavedなサブネット$N'=(P',T',IN',OUT',M_0')$で置き換えて得られるペトリネットを$N"=(P",T",IN",OUT",M_0")$とする。
(1) N"が有界であるための十分条件は、N, N_Bが有界であることである。
(2) N"が安全であるための十分条件は、N, N_Bが安全であることである。
(3) N"がライブであるための必要十分条件は、N, N_Bがライブであることである。 □

(証明) (1)N, N_Bは有界であるとする。$\sigma" \in L(N")$, $M_0"[\sigma">M_1"$とおく。$M_0[\psi(\sigma")>M_1$とすると、すべての$p \in P$について、$M_1"(p) \leqq M_1(p) \leqq k$を満足するk

$\in Z^+$が存在する。また、$M_{B0}[\psi(\sigma'')\rangle M_{B1}$とすると、すべての$p \in P'$について、$k \geqq M_1''(p) = M_{B1}(p) \leqq k'$を満足する$k' \in Z^+$が存在する。したがって、すべての$p \in P''$に対して $M_1''(p) \leqq Max(k,k')$となりN"は有界となる。

(2) (1)の証明でk=k'=1とすればよい。

(3) 十分性：N,N_Bがライブであるとする。このとき、N_Bのwell-behaved-nessから直ちにN"がライブであることを証明できる。

必要性：N"がライブであるとする。このとき、N_Bがライブであることは明らかである。更に、N_Bのwell-behavednessを考慮するとNがライブであることを証明できる。　＜証明終＞

ロボットアームの詳細化の例を考える。図3.8のようにネットN_Bを構成すると、代置ネット図3.6(b)から構成されたネット図3.8(b)がwell-behavedでないので、図3.7(b)のように詳細化したネットのライブ性は保存されないことが検証できる。

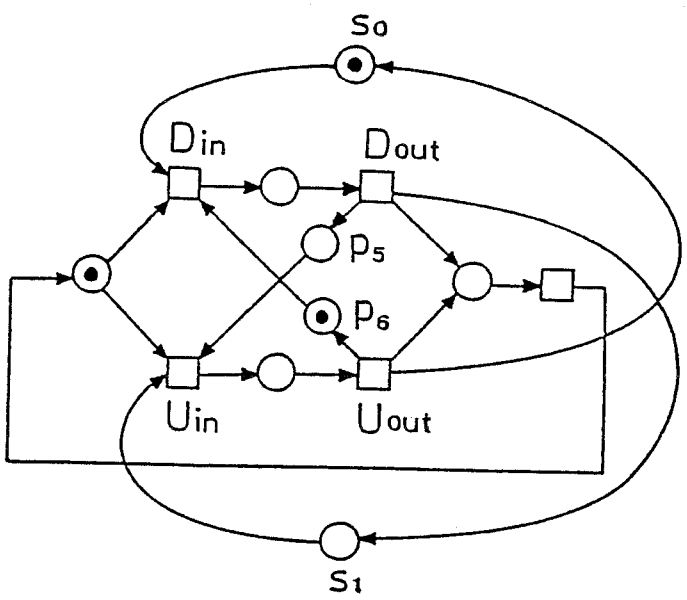
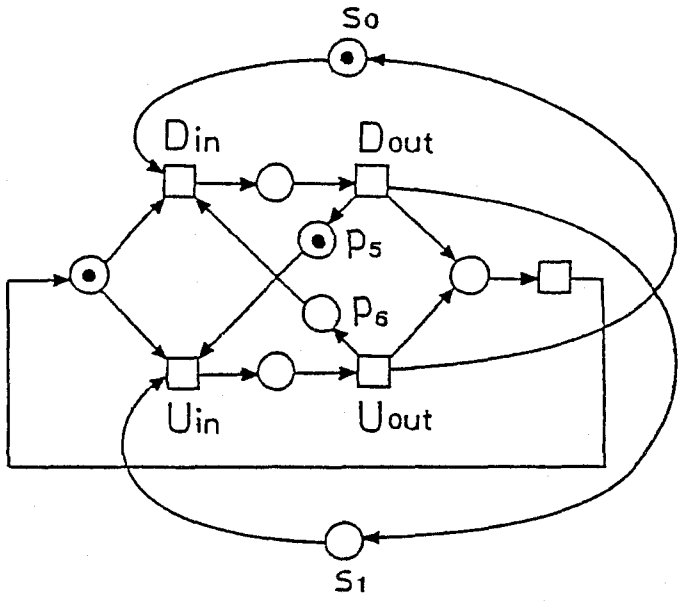


図3.8(a)　N_B：well-behaved



(b) N_B : not well-behaved

図3.8 Well-behavedness

Fig. 3.8 Well-behavedness

３．５ 大規模システムの動作検証

著者らは現在、離散事象システムの動作解析用シミュレータＰＣＳＳ(Petri net-based Concurrent System Simulator)[5]を開発中である。これは、ペトリネット上で発火可能なトランジションを自動的に発火させ、システムの並行動作をシミュレーション解析するものである。

ＰＣＳＳでは、ペトリネットモデルは図3.9 に示す階層構造をなしている。(k-1)層の各コンポーネントは、k層のあるコンポーネント中のトランジションを詳細化したサブネットである。シミュレーションは各コンポーネントを単独で実行することができ、また、必要に応じて下層のサブネットを結合し、シミュレーションを行うこともできる。

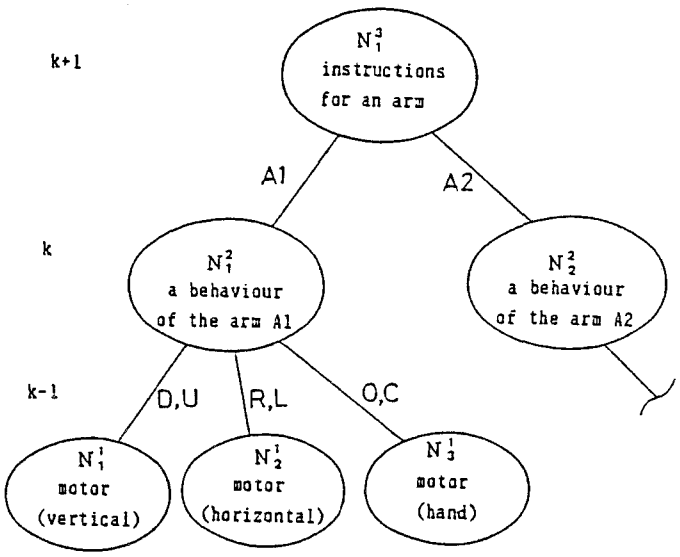


図3.9　ペトリネットモデルの階層構造

Fig. 3.9　Hierarchical structures of Petri net models

このような動作検証用ツールを用いてシステムを設計する際、詳細化のレベルが進むに従ってネットの規模が大きくなり、検証のための手続きも膨大になってくる。また、下層のサブシステムに頻繁に起こる仕様変更に対して全体のネットのシミュレーションを必要としない検証法が望ましい。前節に示した条件を満足するネットによって詳細化していけば各詳細化レベル間のシステムの動作が保存されるので、変更部分に対応する代置ネットの検証を行えば全システムの基本的性質に関する知見が得られ、検証のための手数を大幅に削減することができる。

このように、本章で与えたシステムの階層表現法は大規模な離散事象システムの動作検証の効率を上げるのに貢献している。

３．６　結言

本章では、ペトリネットのトランジションを他のペトリネットで置き換え順次詳細化し、システムを階層表現する際の詳細化規則を与えた。その規則に従って詳細化すれば、元のネットの有界性、安全性及びライブ性等の諸性質が、詳細化後のネットにおいても保存されることを示した。そして、このように階層表現することによって、大規模離散事象システムの動作検証の高効率化が図れることを示した。詳細化すべきトランジション間に本文で設けたような順序関係を仮定しない場合については、動作の保存性についての定式化が難しく、今後の課題として残されている。

第４章　シーケンス制御系のモデル化と動作検証

４．１　緒言

ＦＡ，ＦＭＳで代表されるようにシーケンス制御の対象が大規模かつ複雑化・階層化してくるに伴い、仕様記述から具体的回路設計に至るまでの設計過程に一貫した階層的システムモデルと動作記述言語の必要性が生じてくる。シーケンス制御系設計の体系化の必要性が叫ばれて久しいが(1)、上記のような大規模系の計算機援用設計方式を確立するまでには至っていない。ところで、シーケンス制御系は、離散的な事象がある条件に従って並列・非同期で互いに干渉し合いながら動作する並列離散事象システムと考えることができるが、このような並列離散事象システムをモデル化するのに適したモデルとしてペトリネット(Petri Net)(2)がある。ペトリネットは、最近さらに一般化されてネット理論というシステム理論の一分野が形成されつつあり、このような理論的基盤のもとで具体的な並行システムをペトリネットでモデル化することにより、仕様記述からシステム設計、動作検証、故障診断、システム制御に至るまでの全過程を統一的に取り扱うことができると考えられる(3)。更に、時間付ペトリネット(Timed Petri Net) を用いれば、実システムにおける時間遅れも含めて考えることができる。シーケンス制御系へペトリネットを応用した例としては、文献(4)～(7)などがあり、シーケンス制御系設計方式の体系化の手段として、その有効性が次第に認識されつつある。

ＦＡ，ＦＭＳで代表される複合シーケンス制御系は、種々の制御対象と制御部を持つワークステーション、およびそれら相互間の通信網からなる分散システムと考えることができる。論理シミュレータなどでも各制御部の解析を行うことはできるが、分散システム全体の並行動作や階層的動作記述を行うことは困難である。一方、このような分散システムの各部およびその上位レベルとの通信機能をまとめてペトリネットでモデル化することにより、シーケンス制御系全体の構造的あるいは階層的な動作解析を行うことが可能となる。このためには、まず、シーケンス制御系での最下位レベルの構成単位である制御部および制御対象のペトリネットによる適当なモデル化が必要となる。この際、制御

対象のモデル化は比較的容易に行え、その動作も比較的単純である。

本章では、上記のような分散システムの階層的モデルを構築するための第一段階として、4.3.1小節において複雑な動作をすることが多い制御部のモデル化を行う。従来慣用されている記述法との親和性という応用上の観点から、特に、ＰＣ（Programmable Controller）のプログラムの際によく用いられる展開接続図（ラダー回路）や論理図のペトリネットへの自動変換を可能とする変換規則を提案する。4.3.2小節では、これまで提案されたモデル[7],[8]と比較を行い、本章のモデルがもとの制御システムでは存在しない競合がモデルにおいて生起するという不都合がない、モデルの規模が小さい、変更・保守およびモジュール化・階層化が容易である、などの特徴を持っていることを示す。更に、4.4節では不必要な動作の停止などが起きないように正しく設計された展開接続図のペトリネットモデルが有界かつライブであることを示す。また、ペトリネットでモデル化された実システムの動作検証を行う場合、システムの一部の検証だけが必要な場合が多々ある。そこで、4.5節において検証すべきプレースやトランジションの動作を変化させることなく、それ以外のネットの規模を小さく（縮約）する規則を導出する。また、ネットの縮約規則を設計段階で応用すれば、ネットモデルをより小さな規模で実現することも可能となる。最後に、4.6節において本文で提案したモデルの有効性をシミュレータ PCSS (Petri net-based Concurrent System Simulator)[9] によるシミュレーションにより確かめる。

４．２　記号と準備

本章で用いる記号・用語の説明を行う。接点の名称として、操作力を加えることにより閉路（開路）するもの、あるいはしゃ断器など主接点と開閉を同じくする（逆にする）補助接点をａ接点（ｂ接点）と呼ぶ。また、接点の中で、所定の時間だけ遅れて動作（復帰）する接点を限時動作（限時復帰）接点と呼び、時間遅れなく動作（復帰）する接点を即時動作（即時復帰）接点と呼ぶ。

次に、ペトリネットについて説明する。ペトリネットは、プレース（○で表す）とトランジション（｜で表す）と呼ばれる二種類の節点集合とその間の有向枝集合からなる二部グラフである。ｐからｔ（ｔからｐ）の向きに枝が存在

する場合に、pをtの入力（出力）プレースと呼び、その集合を ${}^{\bullet}t$（$t^{\bullet}$）で表し、tをpの出力（入力）トランジションと呼び、その集合を $p^{\bullet}$（${}^{\bullet}p$）で表す。

通常、プレースは事象生起に関するシステムの条件を表し、トランジションは離散事象の生起と完了を表現する。そして、プレースに対応する条件が現時点で成立している場合には、そのプレースにトークン（●で表す）を置き、トランジションの発火によるトークンの遷移によりシステムの状態遷移を表す。各プレースのトークン数を要素にもつ非負整数ベクトルをマーキングMと呼び、M(p)でプレースpのトークン数を表す。任意の $p \in {}^{\bullet}t$ について、$M(p) \geqq 1$ が成り立つとき、tはマーキングMで発火可能であるといい、Mで発火可能なトランジションtの発火により、Mは次の発火規則に従ってM'に変化する。

$$\begin{cases} M'(p)=M(p)+1 & : p \in t^{\bullet} \wedge p \notin {}^{\bullet}t \\ M'(p)=M(p)-1 & : p \in {}^{\bullet}t \wedge p \notin t^{\bullet} \\ M'(p)=M(p) & : \text{otherwise} \end{cases} \qquad (4.1)$$

更に、発火可能なトランジション t_1 , t_2 のうち、一方を発火させると他方が発火不可能となるようなトランジションは、競合（conflict）しているという。マーキングMから到達可能なマーキングの集合をR(M)で表す。任意の $M \in R(M_0)$ に対して、$M' \in R(M)$ が存在して、M'においてtが発火可能となるとき、M_0 においてtはライブであるという。また、すべてのtがライブであるとき、M_0 においてペトリネットはライブであるという。任意のマーキングにおいて、プレースpに対して $M(p) \leqq K_p$ を満足する正整数 K_p が存在するとき、pは有界であるという。また、すべてのpが有界であるとき、ペトリネットは有界であるという。

本文では、往復有向枝対 "⇄" を "⊸" で表し、この枝を制御枝と呼ぶ。制御枝を持つプレースはトランジションtの発火に対する許可条件を表し、tの発火によるトークンの変化を伴わない。

４．３　ペトリネットによるモデル化

4.3.1　展開接続図のモデル化

本節では、展開接続図（ラダー回路）に使用される主な要素の図記号とそのペトリネット表現について述べるとともに、その接続法について述べる。

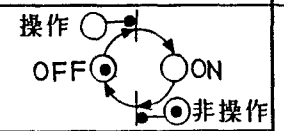
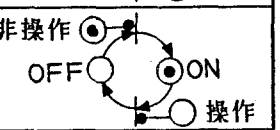
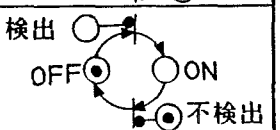
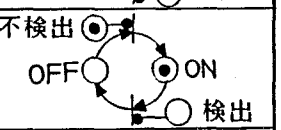
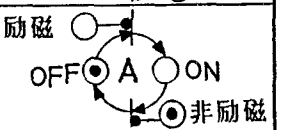
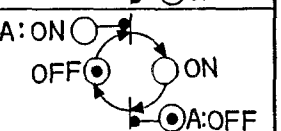
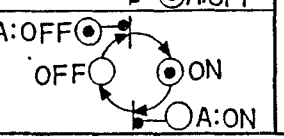
展開接続図で用いられるものは、スイッチ類、電磁コイル、リレー、などであるが、いずれの素子も ON（動作している）状態と OFF（動作していない）状態の二つの状態が存在するだけである。したがって、表4.1のようにONとOFFを表すプレースはトランジションを介して有向枝で結ばれ、有向閉路を形成していると考えるのが自然である。そして、ある条件が成立したとき（条件を表すプレースにトークンが置かれたとき）トランジションの発火により状態が変化すると考えればよい。このように考えると、表4.1のように、各プレースに対する状態の割り当て（意味づけ）を変更するだけで、各素子が同じようにモデル化できる。

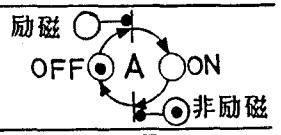
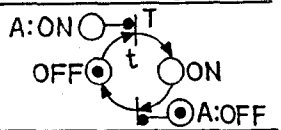
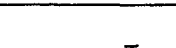
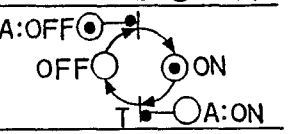
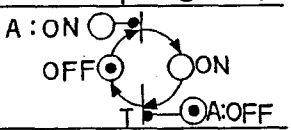
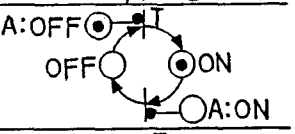
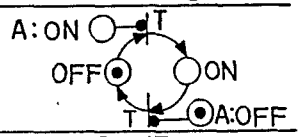
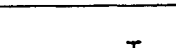
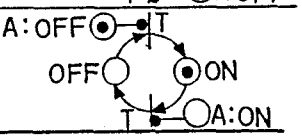
もしスイッチがOFFからONへ変化するのに時間を要するのであれば、限時動作のａ接点のモデルのように、トランジションｔの発火に時間遅れを導入した時間付ペトリネット(Timed Petri Net)[10]を用いればよい。本章のように、ONとOFFという背反な状態を伴うものをモデル化する場合には、ｔが発火可能となる条件が整ってＴ単位時刻後に依然発火可能であるならば、そのとき発火してマーキングが変化するものと考えるのが適当である。

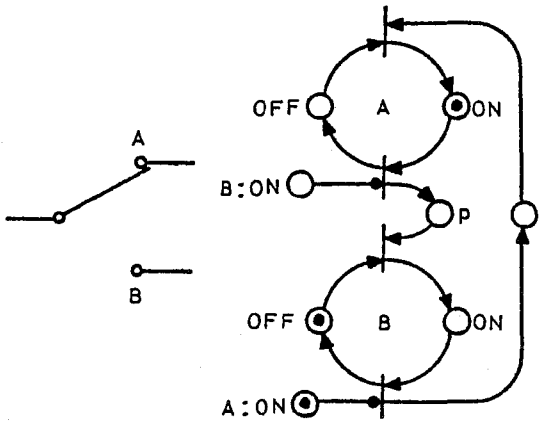
もう少し複雑な動作をする素子も各接点にONとOFF の状態が存在することに注意すれば、図4.1～4.4のように容易にモデル化できる。

たとえば、図4.1の切換接点の接点Ｂを"ON"にするためには、接点Ａが"OFF"にならなければならない。したがって、図4.1のように接点Ａが"OFF"になったことを示すプレースｐを付け加えることにより、その動作を実現している。逆の動作も同様である。

表4.1　基本構成素子とペトリネットモデル

名称		図記号	ペトリネットモデル
復帰形接点	a接点		
	b接点		
機械的接点	a接点		
	b接点		
電磁リレー	コイル		
	a接点		
	b接点		

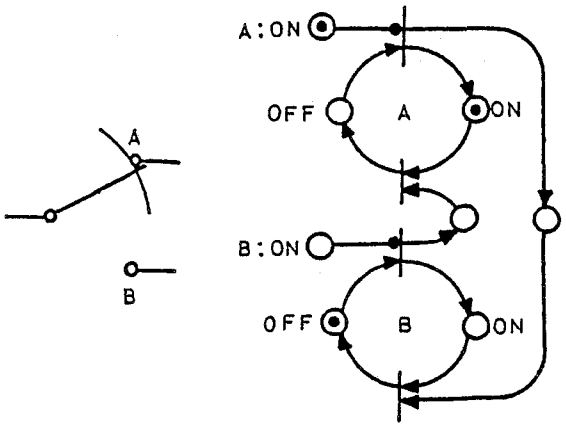
名称			図記号	ペトリネットモデル
限時リレー		コイル		
	限時動作即時復帰	a接点		
		b接点		
	即時動作限時復帰	a接点		
		b接点		
	限時動作限時復帰	a接点		
		b接点		



(i) A symbol　　(ii) A Petri net model

図4.1　切換接点のモデル化

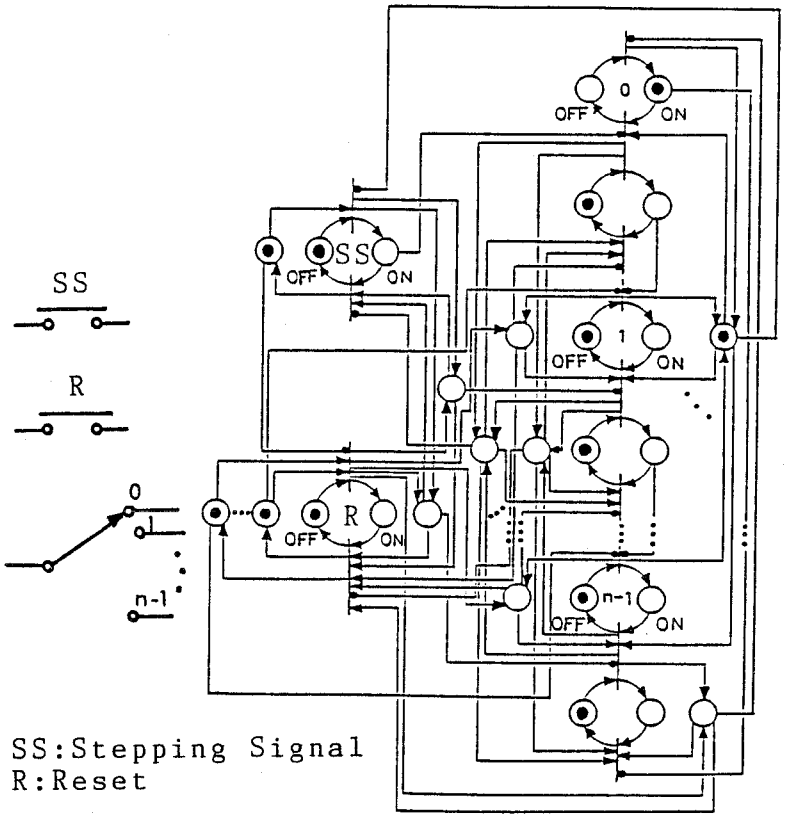
Fig. 4.1　A modelling of transfer switches



(i) A symbol　　(ii) A Petri net model

図4.2　ラップ接点のモデル化

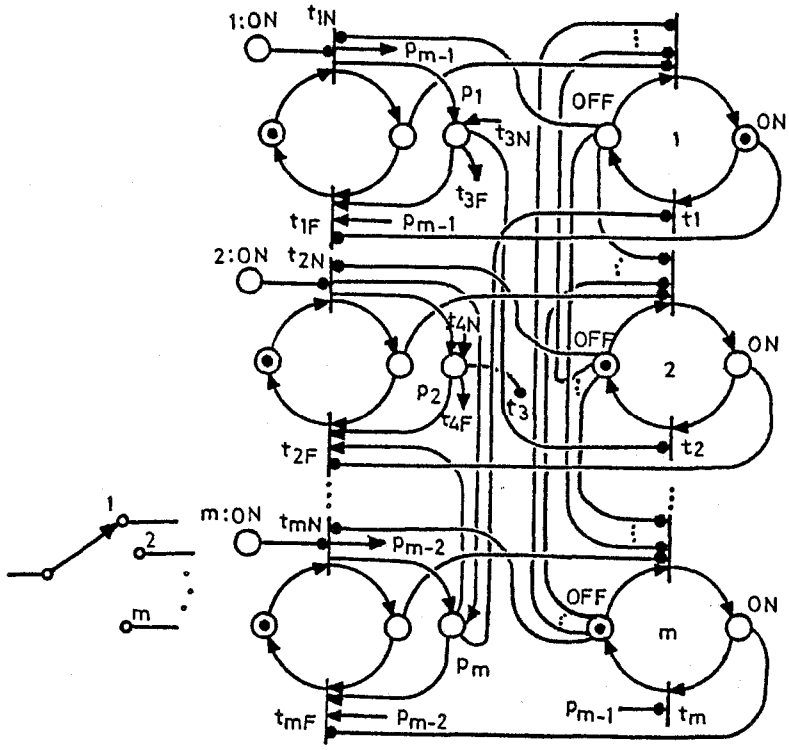
Fig. 4.2　A modelling of lap contacts



(i) A symbol (ii) A Petri net model

図4.3 ステッピングスイッチのモデル化

Fig. 4.3 A modelling of stepping switches



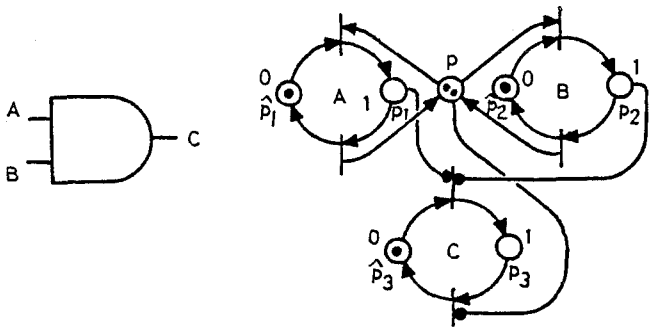
(i) A symbol (ii) A Petri net model

図4.4 ロータリースイッチのモデル化

Fig. 4.4 A modelling of rotary switches

次に、論理素子のモデル化を考える。論理素子の入力と出力には、０と１の二つの状態が存在するだけであるので、表4.1のように０のプレースと１のプレースが出力枝と入力枝をそれぞれ一本ずつ持ち有向閉路を形成していると考えることができる。

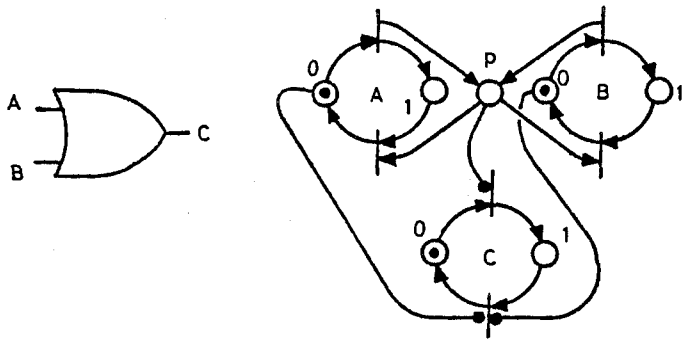
入力をＡ，Ｂ、出力をＣとする。まず、AND素子を考える。ＡとＢが共に１になったときにＣが０から１へ変化するので、図4.5に示すように、p_1とp_2からp_3の入力トランジションへ制御枝を結べばよい。逆に、ＡとＢのうち少なくともどちらか一方が０になったとき、Ｃは１から０へ変化する。そこで、ＡとＢのうち少なくともどちらか一方が０であることを示すプレースｐを図4.5のように付け加え、そこからp_3の入力トランジションへ制御枝を付け加えると、Ｃの１から０への変化が実現できる。図4.5は、Ａ，Ｂ共に０の状態を示しており、ｐにはＡが０であることを示すトークンとＢが０であることを示すトークンが置かれる。また、OR素子も同様にして得られる。これらの素子のモデルが機能的に正しいことは、次節において示す。



(i) A symbol　　(ii) A Petri net model

図4.5　ＡＮＤ素子のモデル化

Fig. 4.5　A modelling of AND elements



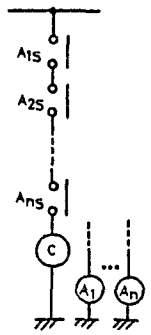
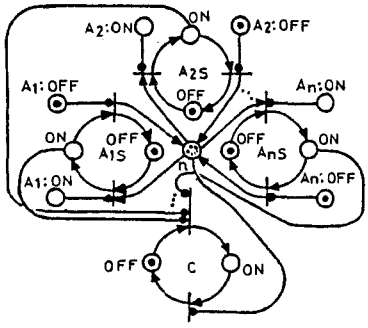
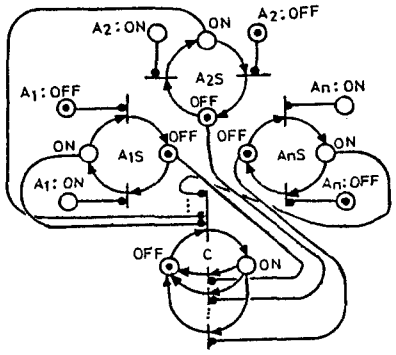
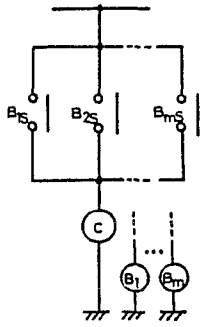
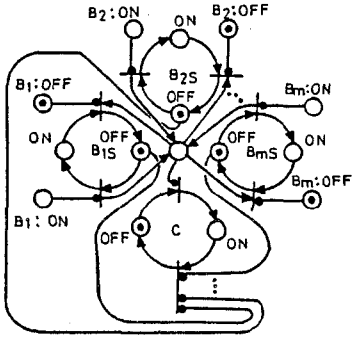
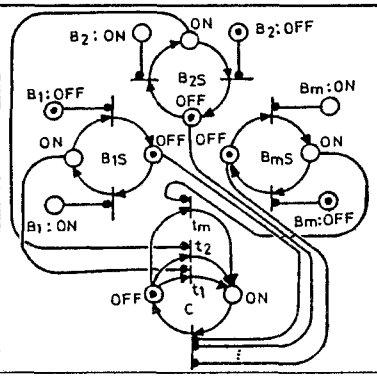
(i) A symbol　　(ii) A Petri net model

図4.6　ＯＲ素子のモデル化

Fig. 4.6　A modelling of OR elements

最後に、各素子の接続法を示す。まず、表4.2のようにスイッチA_{1s}～スイッチA_{ns}が直列に接続されている場合を考える。このとき、Ｃに電流が流れるのはA_{1s}～A_{ns}が共にONのときに限る。つまり、"ON"を"１"、"OFF"を"０"と考えれば、AND素子と同様にして、表4.2のようにモデル化できる。また、並列接続はOR素子と同様にして、モデル化できる。その他の接続、例えばコイルＡがONになるとスイッチA_sがONになる場合をモデル化するには、表4.1のように、コイルＡがONであることを表すプレースからスイッチA_sがONであることを表すプレースの入力トランジションへ制御枝を接続すればよい。したがって、各素子は、表4.2のように接続されるか制御枝で接続される。

表4.2　直・並列接続のモデルの比較

	本文で用いるモデル	文献(7),(8)で用いられているモデル
		
トランジションの数	2n+2	3n+1
プレースの数	4n+3	4n+2
		
トランジションの数	2m+2	3m+1
プレースの数	4m+3	4m+2

4.3.2 他のモデルとの比較

文献(7)や(8)に示されている論理素子のペトリネットモデルと比べて、本章で提案したモデルはトランジションの数が少なくて済む（表4.2）ほか、本来存在しない競合を生じることがない。たとえば、表4.2の並列接続において、すべてのスイッチがONになった状態を考えると、文献(7)や(8)のモデルでは、トランジション$t_1,\ldots,t_m$が競合となる。

次に、展開接続図のシミュレーションについて考える。展開接続図を論理素子に置き換えると、論理シミュレータなどによりシミュレーションが行える。しかし、少なくとも次の点でペトリネットモデルによる方が優れていると考えられる。

- 制御部のペトリネットモデルを制御対象のペトリネットモデルと結合することにより、シーケンス制御系全体を解析することが可能である。
- 表現形式がわかりやすく、各素子が同じような形式（AND,OR等も同じ○，，，）で表現できる。
- 展開接続図等への復元が容易である(経験則を用いない）。
- 初期状態Moでのエラーチェックが可能である（実際では起こり得ないMoが与えられた場合には、Moで発火可能なトランジションが存在する）。

以上のほかに、ペトリネットモデルは、仕様記述と動作記述の一貫性をもたらし、プレース、トランジションによる実システムの抽象度を設計段階に応じて変えることにより、階層設計にも適した長所を有している。特に、本文で提案するモデルのように、同種のネット構造を持つサブネットを接続することによって全体のネットが構成される場合には、より階層設計が容易になると考えられる。

4．4 ペトリネットモデルの機能検証

本節では、前節で提案したペトリネットモデルの構造的性質を調べ、そのモデル機能の検証を行う。

トランジションの数をm、プレースの数をnとし、$(W^+)_{ij}$はt_iからp_jへの枝

の数、$(W^-)_{ij}$はp_jからt_iへの枝の数を表す。

$$(A^+)_{ij} := \begin{cases} (W^+)_{ij}, & t_i \in {}^\bullet p_j \\ 0, & \text{otherwise} \end{cases} \qquad (4.2a)$$

$$(A^-)_{ij} := \begin{cases} (W^-)_{ij}, & t_i \in p_j^\bullet \\ 0, & \text{otherwise} \end{cases} \qquad (4.2b)$$

とする。接続行列は、$A = A^+ - A^-$ と表すことができる。ここで、マーキングMから発火系列 $r_0, r_1, \cdots, r_k$を加えてM' に到達したとすると、

$$M' = M + A^T r, \quad r = r_0 + r_1 + \cdots + r_k \qquad (4.3)$$

と表すことができる[11]。ここで、riは $m \times 1$のベクトルで、発火させるトランジションに対応した成分が1、その他の成分が0である。このとき、次の性質を満足することが知られている。

[Property 4.1] [12]

次の(i) と(ii)は等価である。

(i) 任意の初期マーキングに対して、ペトリネットは有界である。

(ii) $AX \leqq 0$ を満足する正ベクトルXが存在する。 □

[Property 4.2] [12]

(i) が成り立てば(ii)が成り立つ。

(i) $AX = 0$ を満足する非負ベクトルXが存在する。

(ii) M_0 から到達可能な任意のマーキングMにおいて、$X^T M = X^T M_0$ を満足する。 □

まず、有界性を考える。展開接続図のペトリネットモデルは、各素子(スイッチ等)を表すペトリネットモデルを、直列(並列)接続するかあるいは制御枝で結ぶことにより構成される。また、式(4.2) から明らかなように、t_iとp_jの

間に制御枝があっても$(A)_{ij}$には影響を与えないので、展開接続図のペトリネットモデルから制御枝を取り除くことにより得られる各部分ネット(各素子を直列(並列)接続したモデル)が有界であることを示せばよい。説明を具体的にするために図4.7のような$S_1,S_2,\cdots,S_n$の直列接続を例にとって考えよう。

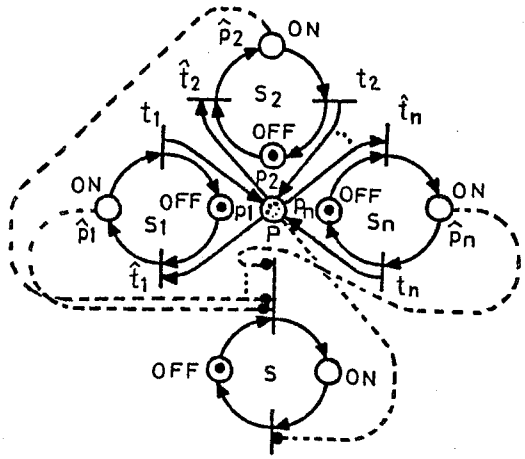


図4.7　直列接続 (S=S1∧S2∧…∧Sn)

Fig. 4.7　A series connection (S=S1∧S2∧…∧Sn)

図4.7の接続行列は次式で表すことができる。

$$A^T = \begin{array}{c} \begin{matrix} t_1 & \hat{t}_1 & t_2 & \hat{t}_2 & \cdots & t_n & \hat{t}_n \end{matrix} \\ \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & -1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \\ 1 & -1 & 1 & -1 & \cdots & 1 & -1 \end{bmatrix} \end{array} \begin{matrix} p_1 \\ \hat{p}_1 \\ p_2 \\ \hat{p}_2 \\ \vdots \\ p_n \\ \hat{p}_n \\ p \end{matrix} \qquad (4.4)$$

このとき、ＡＸ＝０を満足する一次独立なベクトルＸは、次のように n+1個存在する。

$$X_1=\begin{pmatrix}1\\1\\0\\0\\0\\\\0\\0\end{pmatrix},\ X_2=\begin{pmatrix}0\\0\\1\\1\\0\\\\0\\0\end{pmatrix},\cdots,\ X_{n+1}=\begin{pmatrix}0\\1\\0\\1\\0\\\\1\\1\end{pmatrix} \qquad (4.5)$$

ここで、$Y=X_1+X_2+\cdots+X_{n+1}$ とすると、$AY=0$，$Y>0$、を満足するので、 Property 4.1 より、図4.7に示されたペトリネットは有界である。以上の図4.7についての有界性の議論は一般化できて、制御枝を取り除いた他の素子モデル（部分ネット）に対しても、同様に有界であることが示される。

したがって、次の結果を得る。

［構造的性質4.1］

展開接続図のペトリネットモデルは有界である。 □

更に、Property 4.2 を用いると、図4.7の初期マーキングの配置の仕方から、Moから到達可能な任意のマーキングＭにおいて次式を満足する。

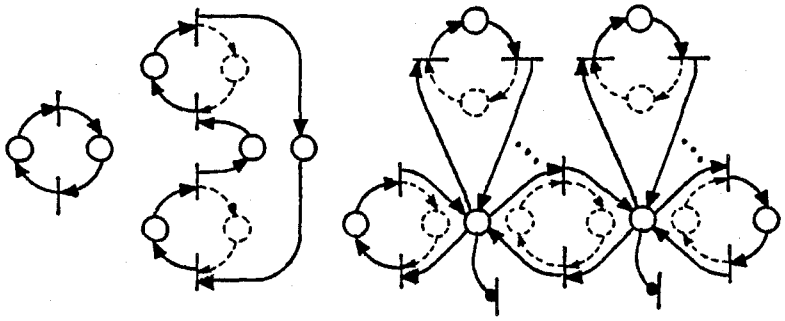
$$X_1^T\cdot M=X_1^T\cdot Mo\ ,\ \cdots\ ,\ X_{n+1}^T\cdot M=X_{n+1}^T\cdot Mo \qquad (4.6)$$

つまり、次の関係が成り立つ。

$$\begin{cases} M(p_1)+M(\hat{p}_1)=1 \\ \quad\vdots \\ M(p_n)+M(\hat{p}_n)=1 \\ M(\hat{p}_1)+\cdots+M(\hat{p}_n)+M(p)=n \end{cases} \tag{4.7}$$

式(4.7)からも、図4.7のプレースｐにトークンが存在するということは、M(p_1)>0 or ... or M(p_n)>0を示していることがわかり、このモデルが AND機能を正しく表現していることがわかる。その他の素子（OR素子、切換接点、ロータリースイッチなど）のモデルも同様に、Property 4.2 を用いることにより、各素子の機能を正しく表現していることがわかる。

次に、本章で提案したモデルのライブ性について考察しよう。そのためいくつかの準備をしておく。まず、強連結で、$|t^*|=|^*t|=1, t\in T$,を満足するペトリネットを SCSM(Strongly-Connected State-Machine)ネット[13]と呼ぶことにする。展開接続図のペトリネットモデルの部分ネットとしてSCSMネットとなる場合の例を図4.8に示す。



(a)　　(b)　　(c)

図4.8　SCSMネット

Fig. 4.8　A SCSM-net

いま図4.8(a)を見ると、初期マーキングの与え方から、同図(a)に存在するトークンは必ず一つである。また、同図(a)の二つのプレースから伸びる各制

御枝が同一のトランジションｔに接続されることは、ｔの発火条件が"ON"かつ"OFF"であることを意味し、このようなことは起こり得ない。同図(b),(c)に対しても同様の事が言える。以上の考察から一般に次の結果を得る。

[構造的性質4.2]

展開接続図のペトリネットモデルにおいて、同じSCSMネットからのびる制御枝が同一のトランジションに接続されるとき、SCSMネットのトークン和は制御枝の数以上存在する。 □

Ｄをプレースの集合とするとき、${}^{\bullet}D \subseteq D^{\bullet}$（$D^{\bullet} \subseteq {}^{\bullet}D$）を満足するプレースの集合をデッドロック（トラップ）と呼ぶ。あるマーキングＭにおいてデッドロックにトークンが存在しないとき、Ｍから到達可能なすべてのマーキングにおいてデッドロックにトークンは存在しない。また、あるマーキングＭにおいてトラップにトークンが存在するとき、Ｍから到達可能なすべてのマーキングにおいてトラップにトークンは少なくとも一つ存在する。

また、初期マーキングMo に対し、デッドなトランジションのR(Mo)に関する極大集合$Dead_M$を

$$Dead_M := \sup\{t \in T \mid t\text{はM'} \in R(Mo)\ \text{でデッド}\} \tag{4.8}$$

と定義する。ただし、Ｍは右辺を与えるマーキングを表す。

[補題4.1]

展開接続図のペトリネットモデルを考える。このとき、任意のトランジションｔ$\in Dead_M$ に対して、任意のマーキングM'$\in$R(M)において、 M'(p)=0 を満足するプレースｐ$\in$ ｔが存在する。 □

（証明） 文献(13),(14) と同様にして、

$$\forall t \in Dead_M,\ \exists s \in {}^{\bullet}t : M'(s)=0,\ \forall M' \in R(M) \tag{4.9}$$

が成立しないと仮定すると、ある $t \in Dead_M$ に対して、集合 $L := \{s \in {}^\bullet t \mid$ sは少なくとも一つのライブな入力トランジションをもつ$\}$ が存在する。すべてのs∈Lに対して、M(s)>0を満足するマーキングMが存在しないので、s∈Lを少なくとも二つ含むSCSMネットが存在する。このとき、構造的性質4.2から、SCSMネット内にトークンはtに接続されている制御枝の数よりも多く存在する。更に、図4.8を見てもわかるように、sがLの要素であれば、あるマーキング M'∈R(M)が存在して、すべてのsに対して、M'(s)>0 となる。これは矛盾であり、式(4.9)が成り立つ。　　＜証明終＞

ペトリネットモデルにおいて、トラップを含まないデッドロックが構成される場合を考える。

図4.9(a)のように一つの素子Bの"ON"と"OFF"の両方にORを表すプレース p_1, p_2が接続している場合を考える。式(4.7) からもわかるようにp_1,p_2のうち少なくともどちらか一方が必ずマーキングされているので、このようなプレースを含むデッドロックからトークンはなくならない。したがって、図4.9(a)のようなプレースp_1,p_2を含まない場合を考えると、同図(b)～(d)のようなデッドロックが得られる。

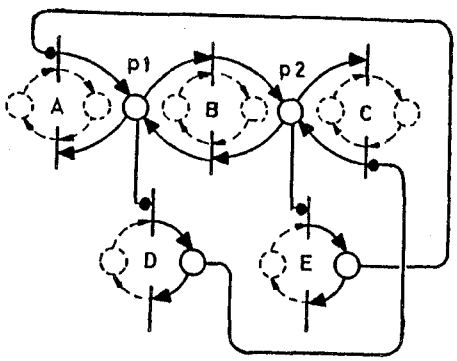


図4.9(a) $D=A\bar{V}\bar{B}, E=B\bar{V}\bar{C}$

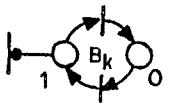
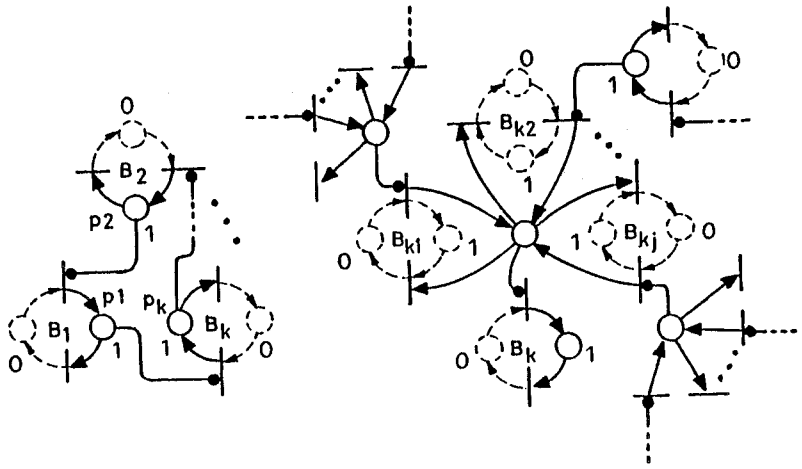


図4.9(b)



(c) (d)

図4.9 トラップを含まないデッドロック

Fig. 4.9 A deadlock containing no trap

図4.9(b)のように、プレースに名前をつけると、トラップを含まないデッドロックは、一般に図4.9(c)のような接続のプレースの集合{$p_1,\cdots,p_k$}として得られる。この接続を論理式で表すと

$$\begin{aligned} B_k &\leftarrow \bar{B}_k \cdot B_1 \\ &\quad \vdots \\ &\leftarrow \bar{B}_k \cdot \bar{B}_1 \cdots \bar{B}_{k-1} \cdot B_k = 0 \end{aligned} \tag{4.10}$$

となり、意味のない接続となる。更に、AND(OR)で接続されているものも含めて図4.9(d)のように考えると、

$$\begin{aligned} B_k &\leftarrow \bar{B}_k \cdot (B_{k1} + \cdots + B_{kj}),\ k,j=1,2,\cdots \\ &\quad \vdots \\ &= 0 \end{aligned} \tag{4.11}$$

が得られ、これも論理式的に意味のない接続となる。以上のことより、一般に次の結果を得る。

［構造的性質4.3］

展開接続図のペトリネットモデルにおいて、トラップを含まないデッドロックが存在するとき、その接続は論理式的に意味のない接続であるかまたは、意味のある接続である場合はデッドロック内のトークンはなくならない。　□

＜注4.1＞

本章では、便宜上、図4.10のようにペトリネットを変換し、不必要な競合が起こらないようにしている。　□

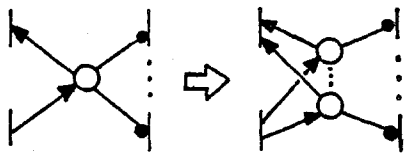


図4.10　競合の回避

Fig. 4.10　An avoidance of conflicts

このとき、次の性質が得られる。

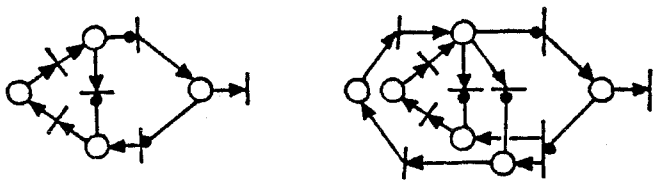
［構造的性質4.4］

展開接続図のペトリネットモデルがライブであるための必要十分条件は、初期マーキングから到達可能な任意のマーキングにおいて、トークンを持たないデッドロックが存在しないことである。　□

（証明）　ライブであるための必要性は、対偶を考えれば明らかである。

ライブであるための十分性を示すため、ライブでないと仮定すると、補題4.1より、文献(14)のTheorem 1 と同様にして矛盾が生じる。　＜証明終＞

構造的性質4.3より、展開接続図のペトリネットモデルに存在するデッドロックがトラップを含まない場合には必ずデッドロック内にトークン存在がしている。したがって、デッドロックがトラップを含む場合を考え、このトラップが必ずマーキングされていることを示す。デッドロックがトラップを含む場合には、SCSMネットの他に図4.11のようなトラップを含む場合がある。



(a)　　(b)

図4.11　SCSMネット以外のトラップを含むデッドロック

Fig. 4.11　A deadlock containing trap excepting SCSM-nets

しかし、展開接続図のペトリネットモデル中に存在するネットを考えると、各素子のモデルおよびそれらの接続方法から、図4.11のような接続のネットは存在しない。したがって、デッドロックがトラップを含むとすれば、それはSCSMネット（トラップ）である。

したがって、構造的性質4.4とSCSMネットは初期マーキングで必ずマーキングされていることより、次の性質が得られる。

[構造的性質4.5]

展開接続図のペトリネットモデルはライブである。　□

４．５　ペトリネットモデルの縮約

ペトリネットでモデル化された実システムの動作検証を行う場合、システムの一部の検証だけが必要な場合が多々ある。この場合、検証すべきプレースやトランジションの動作を変化させることなく、それ以外のネットの規模を小さく（縮約）する規則を導出しておくと便利である。また、ネットの縮約規則を

設計段階で応用すれば、ネットモデルをより小さな規模で実現することも可能となる。

シーケンス制御系で用いられているプログラマブルコントローラのプログラミングに使用されるラダー回路のペトリネットモデル（前節参照）の縮約を考える。縮約規則の一部を列挙すると次のようになる。

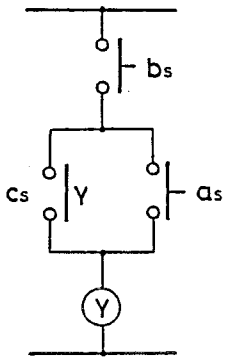
［縮約規則］

(1) "→|←" によってつくられる閉路C上に、制御枝 "|" 以外に枝を3本以上含むプレースpが唯一存在する場合には、pの入力枝e及びeが接続している素子モデルに接続するpの出力枝を取り除くことができる。

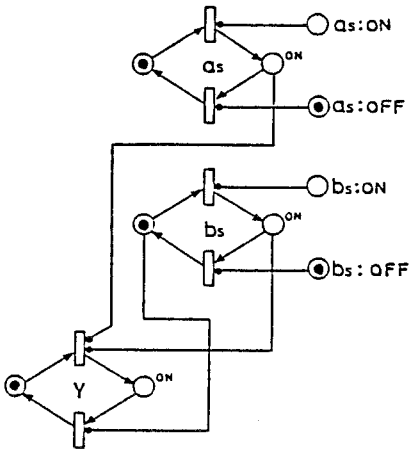
(2) 上述のpの枝の数が3本でかつ制御枝を持たないプレースp'がpと並列に接続されている場合には、pの制御枝をp'につけかえpを取り除くことができる。

(3) 制御枝及び"↑"枝からなる有向路をIとする。検証すべき素子モデルのトランジションに入力する有向路I'∈Iを持たない素子モデルは取り除くことができる。 □

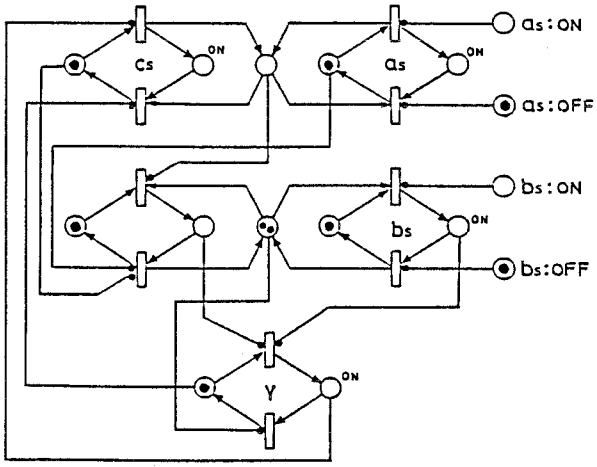
たとえば、図4.12(a)のラダー回路でa_s, b_s, Yの動作が知りたい場合、論理式を利用してもこれ以上簡約化して動作解析をすることはできない。一方、同図(a)のペトリネットモデル（同図(b)）を考え、これに縮約規則を適用すると、より小さな規模のネット（同図(c)）でa_s, b_s, Yの動作検証を行うことができる（c_sも省かれている）。これは、論理式表現では真になる条件と偽になる条件は背反でなければならないのに対し、ネットモデルではその必要がないことによるものである。また、これらの縮約はすべてPCSS上で実行できるようになっている。



(a) A ladder diagram



(c) A reduced model of (b)



(b) A Petri net model of (a)

図4.12　ネットモデルの縮約

Fig. 4.12　Reductions of Petri net model

４．６　例題

図4.13の展開接続図をペトリネットでモデル化すると、図4.14のようになる。これを実際にシミュレータPCSSで実行すると図4.15のようなタイムチャートが得られ、展開接続図の動作を正しくモデル化していることが確かめられる[3],[9]。

また、展開接続図は大規模であることが多いため、図4.16(i)のスイッチAsのような意味のない素子が設計者の意図に反して混入する場合が多々ある。このような素子は、ペトリネットがライブであるための条件を用いれば容易に発見することができる。たとえば、同図(ii)のペトリネットモデル中のプレース{p_1,p_2}は空なデッドロックであるので、構造的性質4.4から、シミュレーションすることなくAsがONの状態から変化しないことを確認できる。

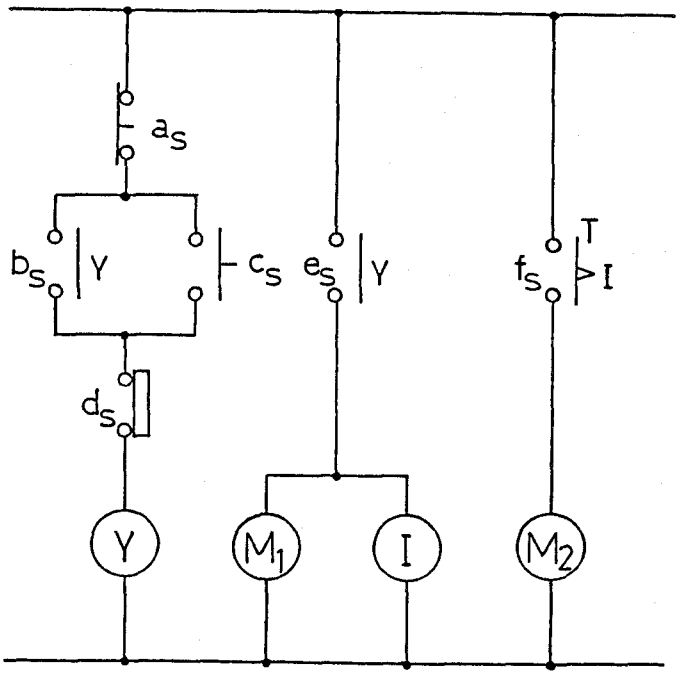


図4.13　シーケンス制御回路の例

Fig. 4.13　An example of sequential control circuits

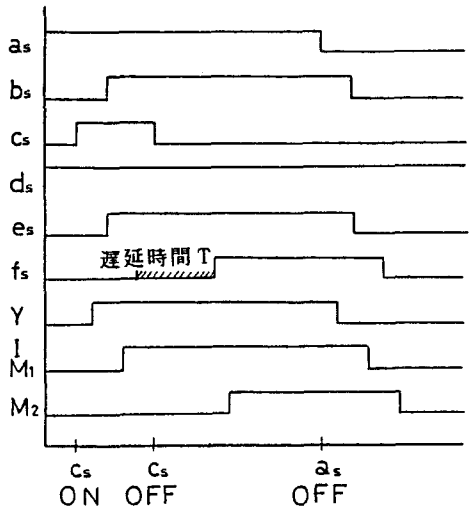


図4.14　シーケンス制御回路のタイムチャート

Fig. 4.14　A time-chart of sequential control circuits

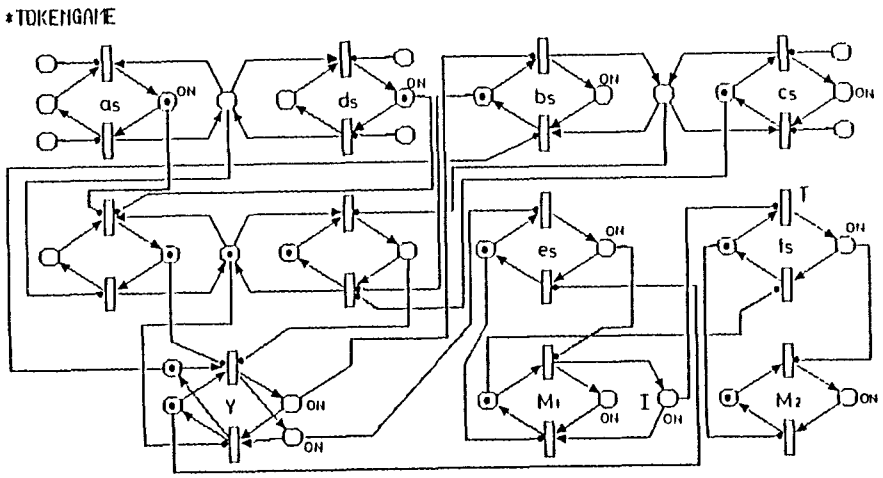
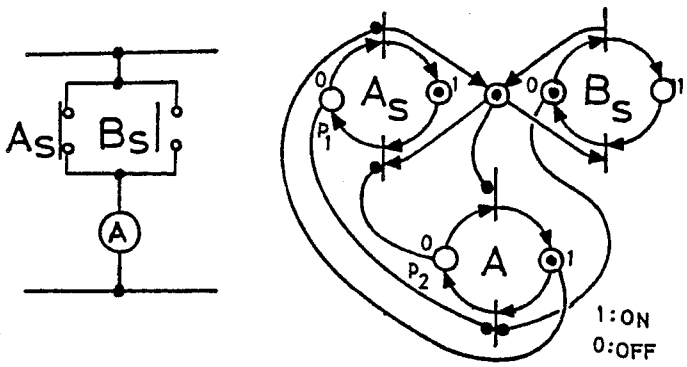


図4.15　シーケンス制御回路のペトリネットモデル

Fig. 4.15　A Petri net model of sequential control circuits



(i) A ladder diagram (ii) A Petri net model

図4.16 意味のないスイッチを含む展開接続図

Fig. 4.16 A ladder diagram including unworthy switches

4．7 結言

本章では、シーケンス制御系の設計によく用いられる展開接続図（ラダー回路）や論理図をペトリネットによってモデル化するための一つの方法を示した。そして、本章で提案したモデルは、従来あるモデルと比べて、トランジションの数が少なくて済み、モデル化によって不必要な競合が起こらない等の特徴を有していることを示した。更に、不必要な動作の停止などが起きないように正しく設計された展開接続図のペトリネットモデルが有界でありかつライブであることを示した。次に、ラダー回路のペトリネットモデルの縮約規則を示した。この規則を用いると、システムの一部の検証だけが必要な場合やネットモデルをより小さな規模で実現する場合に有用である。また、ペトリネットがライブであるための必要十分条件を用いれば、設計者が見落しがちな意味のない素子も容易に検出できることを示した。

第５章　結　論

本論文では、複雑な動作をする大規模な並行離散事象制御系の動作解析がペトリネットを用いて厳密な数学的基盤のもとで行えることを示した。特に、システムのデッドロックに関連する主要な問題の一つであるライブ性に関する理論的考察を通じて、ネット理論に基づく離散事象システムの設計理念の確立に対する一つの基礎を与えた。本論文で得られた結果の大略を述べると、応用上有用であるペトリネットのサブクラスの構造的性質とその有界性・ライブ性との関係の導出、ペトリネットを用いて階層的設計を行う場合のライブ性などの諸性質を保存する詳細化規則の導出、およびシーケンス制御系のペトリネットによるモデル化と動作検証である。以下、各章で得られた結果の内容をやや詳しく説明する。

第２章では、ペトリネットのサブクラスであり、並列処理などを表現するモデルとして応用上有用であるマークグラフに制御枝（ペトリネットの往復枝に相当）と呼ばれる枝を付加した拡張マークグラフのあるサブクラスの解析を行った。そして、ライブであるための必要十分条件を導出した。特に、ネット構造と初期マーキングに関する適用しやすい条件として導出した。更に、本章で得られた結果が、従来の条件ではライブ性を調べることが困難であったネットに対しても有効であることを示した。また、本章で導いた定理を用いれば、マルチタスクシステムなどの場合、シミュレーションすることなしに実行不可能なタスクを発見することができることを示した。本章で扱った拡張マークグラフは、マークグラフではモデル化できなかった選択性（競合性）をモデル化することができ、更に、従来論議されている拡張マークグラフよりも広いクラスの拡張マークグラフである。

第３章では、従来の設計方法が、いくつかのサブシステムを組合わせて一つの統合されたシステムを構成する方法や、システムの大枠から設計を行い各部分を順次具体化してゆくという階層的設計方法であることから、ペトリネットモデルのトランジションをそのトランジションが表す動作を詳細表現した他のペトリネットで逐次置き換えるというトップダウン形式のシステム表現を考えた。そして、どのような規則を用いて詳細化すれば、詳細化する前のペトリネ

ットの動作が詳細化後のペトリネットにおいても完全に保存されるかということを考えた。特に、複数のトランジションを詳細化したとき、各々のサブネットの動作に相互関係が生ずる場合が多々あり、それらのサブネットを接続する必要が生ずる。そこで、従来の単一トランジションの詳細化に対して得られていた性質の保存性に関する結果を、システムの任意の部分（トランジションの集合）の詳細化に対して一般化した。この規則に従って詳細化すれば、元のネットの有界性、安全性及びライブ性等の諸性質が、詳細化後のネットにおいても保存されることを示した。そして、このように階層表現することによって、大規模離散事象システムの動作検証の高効率化が図れることを示した。本章で示した条件を満足するネットによって詳細化していけば各詳細化レベル間のシステムの動作が保存されるので、変更部分に対応する代置ネットの検証を行えば全システムの基本的性質に関する知見が得られ、検証のための手数を大幅に削減することができる。したがって、本章で与えたシステムの階層表現法は大規模な離散事象システムの動作検証の効率を上げるのに貢献するものと考えられる。

第４章では、並行離散事象制御系、特に、シーケンス制御系のモデル化と動作検証を行った。シーケンス制御系は、種々の制御対象と制御部を持つワークステーション、およびそれら相互間の通信網からなる分散システムと考えることができ、制御対象のモデル化は比較的容易に行え、その動作も比較的単純である。分散システムの階層的モデルを構築するための第一段階として、複雑な動作をすることが多い制御部のモデル化を考えた。特に、従来慣用されている記述法との親和性という応用上の観点から、ＰＣ(Programmable Controller)のプログラムの際によく用いられる展開接続図（ラダー回路）や論理図のペトリネットへの自動変換を可能とする変換規則を提案した。本章で提案したモデルは、これまで提案されたモデルに比べて、もとの制御システムでは存在しない競合がモデルにおいて生起するという不都合がない、モデルの規模が小さい、変更・保守およびモジュール化・階層化が容易である、などの特徴を持っていることを示した。更に、不必要な動作の停止などが起きないように正しく設計された展開接続図のペトリネットモデルが有界でありかつライブであることを示した。次に、システムの一部の検証だけが必要な場合やネットモデルをより

小さな規模で実現する場合に有用であるラダー回路のペトリネットモデルの縮約規則を示した。また、ペトリネットがライブであるための必要十分条件を用いれば、設計者が見落しがちな意味のない素子も容易に検出できることを示した。

以上のように、並行・非同期動作を特徴とする離散事象制御系の数学的解析を可能とするペトリネットを用いたモデル化や動作検証および解析を行い、特に、そのライブ性に関する詳細な考察を行った。得られた結果は、並行離散事象システムの階層設計における重要な理論的基礎を与えるものである。しかし、一般にペトリネットを実システムの制御へ応用する場合には、ペトリネットの拡張が必要になる。例えば、システムの性能評価などを行おうとすれば時間の概念の導入が必要であるし、実システムを表現しようとするとトークンに何らかの属性を持たせる必要が生じてくる。前者に対しては時間付ペトリネット（Timed Petri Net）[1]、後者に対しては色付ペトリネット(Coloured Petri Net)[2]と呼ばれる拡張されたペトリネットが知られている。しかし、これらのネットの可到達性やライブ性などの基本的性質についての決定問題は未解決であり、今後の研究課題として残されている。

謝辞

本研究の全過程を通じて、直接理解ある御指導を賜り、常に励まされ御助言頂いた大阪大学工学部電子工学科児玉慎三教授ならびに熊谷貞俊助手、第３章の階層的詳細化について貴重な御意見を頂いたダイキン工業㈱の山田智彦氏に心から感謝致します。

大学院工学研究科後期課程において御指導、御教示を頂いた大阪大学工学部電子工学科寺田浩詔教授、西原浩教授、浜口智尋教授に対し、厚く御礼申し上げます。

また、種々の面で御指導を頂いた大阪大学工学部電子工学科白川功助教授、前田肇助教授ならびに築山修治助手に心から感謝致します。

筆者の所属する研究室の楠岡英雄助手、太田快人助手ならびに朴和子事務補には種々の面で御協力を頂きました。心から感謝致します。

最後になりましたが、親切な御援助を頂いた電子工学教室の諸氏、児玉研究室の諸氏に御礼申し上げます。

参考文献

第1章

(1) J.L.Peterson:" Petri Net Theory and the Modelling of Systems," Prentice-Hall,(1981).

(2) C.Ramchandani : " Analysis of asynchronous concurrent system by Petri Nets," Project MAC, TR-120, MIT, Cambridge, (1974).

(3) E.W.Mayr:" An algorithm for the general Petri net reachability problem,"Proc. of the 13th. ann. ACM Symp. on Theory of Computing, Milwarkee, (1981).

(4) M.Jantzen & R.Valk:" Formal properties of place/transition nets," Net Theory & Appl., Lecture Notes in Computer Science, 84, Springer-Verlag, (1980).

(5) T.Murata:"Circuit theoretic analysis and synthesis of marked graphs," IEEE Trans. on Circuits & Systems, CAS-24-7, pp.400-405, (1977).

(6) T.Taguchi, S.Kodama,and S.Kumagai:"Analysis of marked graphs with safe condition,"Trans. IECE JAPAN, J63-D, 4, pp.342-348, (1980).

(7) S.Kumagai, S.Kodama, and M.Kitagawa:"Submarking reachability of marked graphs," IEEE Trans. on Circuits & Systems, CAS-31-2, pp. 159-164, (1984).

(8) T.Murata:"Modeling and analysis of concurrent systems," Handbook of software Engineering, ch.3, Van Nostrand, (1984).

(9) S.Kumagai, S.Kodama, T.Naito and T.Sawai:" On the structural properties of extended marked graphs," Proc. of ISCAS84, pp.1386-1389, (1984).

(10) R.Valette : " Analysis of Petri Nets by Stepwise Refinements," J. Comput. & Syst. Science, 18, pp.35-46 (1979).

(11) I.Suzuki and T.Murata : " A method for stepwise refinement and abstraction of petri nets," J. Comput.& Syst. Science, 27, 1,

(1983).
(12) シーケンス制御工学体系化調査専門委員会：" シーケンス制御の工学体系化に関する調査研究," 電気学会技術報告，第138号，(1982).
(13) 長谷川健介："マークフローグラフとＦＡの応用," 計測と制御，22-11, pp.946-945 (1983).
(14) D.Azema, R.Valett and M.Diaz："Petri Nets as a common tool for design verification and hardware simulation," 13th Design Automation Conf., San Flancisco, pp.109-116 (1976).
(15) N.Komoda et al："An autonomous, decentralized control system for factory automation Computer," IEEE Computer Society, 17-12, pp. 73-83 (1984).
(16) 中村，阿部，武田："ペトリネットを用いたシーケンス制御系の一解析手法," 電気学会システム制御研究会資料，SC-83-48, pp.21-30 (1983).
(17) J.Hozjajenok and T.Huhtanen：" Introduction of combinational extended Petri Nets," Report B84, Digital Systems Laboratory, Helsinki University of Technology, (1981).
(18) 山本，山田，内藤，辻，熊谷，児玉："ペトリネットに基づいた並行システムシミュレータPCSSについて," 第24回計測自動制御学会学術講演会予稿集，pp.917-918 (1985).

第２章

(1) S.Kumagai, S.Kodama, T.Naito, and T.Sawai:" On the structural properties of extended marked graphs," Proc. of ISCAS84, pp.1386-1389 (1984).
(2) E.W.Mayr:"An algorithm for the general Petri net reachability problem," Proc. of the 13th ann. ACM Symp. on Theory of Computing, Milwaukee, (1981).
(3) J.L.Peterson:"Petri Net Theory and the Modeling of Systems," Prentice-Hall, (1981).
(4) T.Murata:"Modeling and analysis of concurrent systems," Handbook

of Software Engineering, ch.3, Van Nostrand, (1984).
(5) 田口，児玉，熊谷：”セイフ条件を考慮したマークグラフの解析” 電子通信学会論文誌，J63-D, 4, pp.342-348 (1980).
(6) S.Kumagai, S.Kodama, and M.kitagawa:"Submarking reachbility of marked graphs," IEEE Trans. on Circuits & Systems, 31, 2, pp.159-164 (1984).
(7) M.Jantzen and R.Valk:"Formal properties of place/transition nets, Springer Lecture Notes in Computer Science 84, Net Theory and Applications, Springer-Verlag, pp.165-212 (1980).
(8) S.Kumagai, S.Kodama, T.Naito and T.Sawai:"On the structural properties of extended marked graphs," Proc. of ISCAS84, pp.1386-1389 (1984).

第3章

(1) J.L.Peterson : "Petri Net Theory and the Modeling of the Systems," Prentice Hall (1981).
(2) R.Valette : " Analysis of Petri Nets by Stepwise Refinements," J. Comput. & Syst. Science, 18, pp.35-46 (1979)
(3) I.Suzuki and T.Murata : " A method for stepwise refinement and abstraction of petri nets," J. Comput.& Syst. Science, 27, 1, (1983).
(4) L.Ya.Rozenblyum:" Petri Nets," Scripta Publ. Co., pp.19-43 (1984).
(5) 山本，山田，内藤，辻，熊谷，児玉："ペトリネットに基づいた並行システムシミュレータPCSSについて," 第24回計測自動制御学会学術講演会予稿集，pp.917-918 (1985).

第4章

(1) シーケンス制御工学体系化調査専門委員会：” シーケンス制御の工学体系化に関する調査研究," 電気学会技術報告，第138号，(1982).
(2) J.L.Peterson : " Petri Net Theory and the Modeling of Systems,"

Prentice-Hall, New Jersey (1981).

(3) 児玉，熊谷："離散事象システム -ネットモデルによるアプローチ-," 計測と制御，24-7，pp.41-50 (1985).

(4) 長谷川健介："マークフローグラフとＦＡの応用," 計測と制御，22-11，pp.946-945 (1983).

(5) D.Azema, R.Valett and M.Diaz : "Petri Nets as a common tool for design verification and hardware simulation," 13th Design Automation Conf., San Flancisco, pp.109-116 (1976).

(6) N.Komoda et al : "An autonomous, decentralized control system for factory automation Computer," IEEE Computer Society, 17-12, pp. 73-83 (1984).

(7) 中村，阿部，武田："ペトリネットを用いたシーケンス制御系の一解析手法," 電気学会システム制御研究会資料，SC-83-48，pp.21-30 (1983).

(8) J.Hozjajenok and T.Huhtanen : " Introduction of combinational extended Petri Nets," Report B84, Digital Systems Laboratory, Helsinki University of Technology, (1981).

(9) 山本，山田，内藤，辻，熊谷，児玉："ペトリネットに基づいた並行システムシミュレータPCSSについて," 第24回計測自動制御学会学術講演会予稿集，pp.917-918 (1985).

(10) C.Ramchandani : " Analysis of asynchronous concurrent system by Petri Nets," Project MAC, TR-120, MIT, Cambridge, (1974).

(11) T.Murata:"State equation, controllability, and maximal matchings of Petri nets," IEEE Trans. Autom. Control, AC-22-33, pp.412-416 (1977).

(12) G.Memmi and G.Roucairol:"Linear algebra in net theory," Springer Lecture Notes in Computer Science84, Net Theory and Applications, Springer-Verlag, pp.213-223 (1980).

(13) M.Jantzen and R.Valk : " Formal properties of place/transition nets," Springer Lecture Notes in Computer Science 84, Net Theory and Applications, Springer-Verlag, pp.165-212 (1980).

(14) S.Kumagai, S.Kodama, T.Naito and T.Sawai:" On the structural properties of extended marked graphs," Proc. of ISCAS 84, pp. 1386-1389 (1984).

第5章

(1) C.Ramchandani : " Analysis of asynchronous concurrent system by Petri Nets," Project MAC, TR-120, MIT, Cambridge, (1974).

(2) K.Jansen : " Coloured Petri nets and the invariant-method," DAIMI PB-104, Computer Science Department, Aarhus University, Denmark, (1979).

No.25113474056468103816011614123114611l