氏名	田中章喜（たなかあきよし）
授与学位	工学博士
学位授与年月日	昭和53年3月24日
学位授与の根拠法規	学位規則第5条第1項
研究科，専攻の名称	東北大学大学院工学研究科 （博士課程）電気及通信工学専攻
学位論文題目	**並列処理システムに関する研究**
指導教官	東北大学教授 野口 正一
論文審査委員	東北大学教授 野口 正一　東北大学教授 城戸 健一 東北大学教授 木村 正行　東北大学教授 星子 幸男

論文内容要旨

第1章 序論

近年，計算機応用分野の拡大に伴ない，計算機複合体等の並列処理システムの発達は著しいものがある。並列処理システムでは従来の逐次形処理ではみられなかった様々な問題が生じ，理論的な検討が必要とされている。本論文では並列処理システムでは特に重要であるプロセス間相互作用の制御法を検討する。プロセス間相互作用は共通バッファを介する通信，共有ファイルへのアクセス，システム資源の排他的使用等がある。従来の研究で，与えられた並列プログラムがこれらプロセス間相互作用を満たしかつデッドロック・フリーであるか否かを証明する方法はプログラム理論の応用として帰納法，公理論的アプローチ，補助変数の導入による証明などいくつか提案されてきた。しかし，これらの方法は述語論理を用いて厳密に証明されるが具体的な証明にあたっては述語の適切な発見法や補助変数の適切な導入法がなくかなり複雑になる。一方，シス

テム設計者やプログラマにとっては，並列プログラムを直接に証明するよりも，プロセス間相互作用の具体的な制御法が与えられることが望ましい。すなわち並列プログラムの合成法である。本論文はプロセス間相互作用が与えられたとき，これら相互作用を同期基本命令を用いて制御する方法を具体的に示し，与えられた相互作用をすべて満たし，かつシステムがデッドロック・フリーとなることを保証するプログラム構成アルゴリズムを提案し，その正当性を与える。

第2章　並列処理システムの基礎的定義

本章では考察の対象となる並列処理システムの定義を行なう。まず，並列プログラムがプロセスの並列合成によって定義される。プロセスは作用と呼ばれる処理単位の直列合成であり，各作用はプロセスの進行を制御する制御部と具体的な処理を行なう実効部からなる。並列処理システムは並列プログラムとプログラム変数の初期値ベクトルの対によって定義される。そして，このシステムの特徴として相互作用の制御を論ずるには相互作用に寄与するプロセスの作用のみを考察すればよいことを示す。

次に，プロセス間相互作用を定義する。本論文では3種類の相互作用が定義されるが，いずれも相互排他的な相互作用で，作用の実効部間の関係として与えられる。第1番目は相互排除関係であり，これは相互作用を構成する作用が各プロセスで1つに限定されている。第2番目は遷移禁止関係で，1つのプロセスが連続する作用，他のプロセスが1つの作用から構成している相互作用である。第3番目は相互遷移禁止関係で，複数のプロセスが連続する作用から構成している相互作用である。相互排除関係は従来のクリティカル・セクションの相互排除問題に対応し，遷移禁止関係，相互遷移禁止関係は排他的な相互作用の一般化である。本章では，さらにこれら相互作用を制御する同期基本命令を定義する。プロセスの作用の制御部はこの同期基本命令の直列合成である。プロセス間相互作用は同期基本命令によってプロセスの進行をブロック（待ち状態に入る）したり，解放することにより同期がとられ制御される。従って，待ち状態に入るプロセスの待ち時間は最小であることが要求される。これを待ち時間最小条件といい，この条件もみたすように並列プログラムを構成する。さらに，システムの動作の記述が明解となる状態グラフを定義する。

第3章　2プロセスシステムの制御

本章では基礎的な場合として，2つのプロセスからなる並列処理システムにおける相互作用の制御法を明らかにする。まず実効部間の関係で与えられる相互排除関係が同期基本命令によって作用の制御部で制御できることを示し，相互排除関係のみが与えられたときの並列プログラムの構成法を示し，その結果得られるシステムがデッドロック・フリーとなることを示す。

次に，遷移禁止関係の制御法を論じる。遷移禁止関係が与えられた場合には，閉塞状態が生じうる。閉塞状態はそれ自身ではデッドロック状態ではないが，相互作用が満たされない状態，すなわち個々の相互作用が同期基本命令によって制御された後，必ずデッドロック状態に陥入る状態である。従って，システム中に閉塞状態は存在してはならず，閉塞状態も同期基本命令によって制御されなければならない。本章では，まず個々の遷移禁止関係が作用の制御部によって制御されることを示し，遷移禁止関係が与えられたとき，それらによって生じる閉塞状態を求める操作を与え，この操作によってすべての閉塞状態が求められることを証明する。次に，遷移禁止関係のみが与えられたときの並列プログラムの構成アルゴリズムを提案し，その結果得られるシステムがすべての相互作用及び待ち時間最小条件をみたし，かつデッドロック・フリーであることを証明する。

相互遷移禁止関係が与えられたときも閉塞状態は生じる。しかし，相互遷移禁止関係は遷移禁止関係に帰着されることが示されるので，相互遷移禁止関係は遷移禁止関係の制御アルゴリズムによって制御される。

最後に3種類の相互作用が一般的に与えられた場合の並列プログラムの構成アルゴリズムを提案し，その結果得られるシステムが与えられたすべての相互作用をみたしかつ待ち時間最小条件をみたし，デッドロック・フリーとなることを証明する。

第4章　2プロセスシステムの最短実行時間

本章では第3章のアルゴリズムによって得られたシステムにおいて，各プロセスが非同期に実行されるときの最短実行時間を求める方法を示す。ただし，プロセスの各作用の実行時間はすべて等しいものとする。最短実行時間は状態グラフから求めることができる。システムの実行時間は実際の処理に要する時間とプロセス間相互作用をみたすために同期基本命令によって制御される待ち時間の和になることが示される。前者はプロセスが与えられると一意に定まる値である。従って，待ち時間の最小値を求めることにより最短実行時間が得られる。最小の待ち時間は状態グラフから容易に得られる。本章ではこの最短実行時間を状態グラフから求めるアルゴリズムを示し，その正当性を証明している。

第5章　Nプロセスシステムの制御

本章は第3章の結果の一般のNプロセスシステムへの拡張である。2プロセスシステムと大きく異なる点はローカル・デッドロック状態が存在することである。ローカル・デッドロック状態というのは，その状態でシステム全体ではデッドロックではないが，いくつかのプロセスのみが永久待ち状態となっている状態である。永久待ち状態となっているプロセスは永久待ち状態となっていないプロセスが任意の状態となっても永久待ち状態が解除されないのがローカル・デッド

ロック状態の特徴である。また，システムがローカル・デッドロック・フリーであればシステム全体もデッドロック・フリーとなることが示される。

相互排除関係のみが与えられた場合の制御法は第3章の方法の一般的な場合への単純な拡張によって得られ，相互排除関係を制御する同期基本命令をその実行順序が一定の規則に従うようにプログラムへ組込むことによって，ローカル・デッドロック・フリーであることが保証される。

次に，遷移禁止関係のみが与えられたときの制御法を与える。相互作用の制御手段として本論文で用いる同期基本命令はその変数（同期変数）のとる値が0か1の2値である。従って，この同期基本命令を組込んだシステムは同期変数を資源とする従来研究されてきた多種類単一資源システムに対応づけることができる。多種類とは同期変数が複数個用いられることであり，単一資源とは各同期変数のとる値が前述の2値であることに対応する。従来得られている結果として，多種類単一資源システムがデッドロック・フリーである必要十分条件は資源待ちのプロセスがサイクル待ちとならないことが示されている。従って，まず第3章の方法によってすべてのプロセス対間での相互作用の制御と，それらから生じる閉塞状態を求めて，多種類単一資源システムに帰着させる。次に本章で与える条件によって，同期基本命令の実行に関するサイクル待ち状態を検出する。サイクル待ちの検出は各同期変数が2つのプロセスにのみ出現するように制限されているので容易である。この方法によって，すべての閉塞状態が求められることの証明も与えている。このようにしてすべての閉塞状態が求められたのち，具体的に同期基本命令をプログラムに組込む。この方法によって制御されたシステムが与えられた遷移禁止関係をすべてみたし，かつローカル・デッドロック・フリーとなることの証明が与えられている。

次に相互遷移禁止関係の制御であるが，一般のNプロセスの場合も相互遷移禁止関係は遷移禁止関係に帰着されるので，遷移禁止関係の制御と同様な方法によって並列プログラムが構成される。

最後に3種類の相互作用が一般的に与えられた場合のプログラム構成アルゴリズムを与え，その正当性を証明する。相互排除関係と他の相互作用によっては閉塞状態が生じないという結果を用いて，本アルゴリズムではまず遷移禁止関係と相互遷移禁止関係によって生ずる閉塞状態を求め，サイクル待ちを検出した後に，各相互作用及び閉塞状態を制御する同期基本命令をプログラムに組込み，並列プログラムを構成する。

第6章　結　　論

本論文では並列処理システムにおいて，プロセス間相互作用が与えられた場合の制御法を明らかにし，並列プログラム構成アルゴリズムを具体的に与え，その結果得られるシステムがすべて

の相互作用をみたし，かつデッドロック・フリーとなることを証明した。

本研究で考察の対象としたプロセス間相互作用は，従来のクリティカル・セクション間の相互排除のみでなく，連続する作用間での相互排他的な相互作用（遷移禁止関係，相互遷移禁止関係）にまで拡張した一般的な排他的相互作用であり，並列処理システムの制御プログラムや，オペレーティング・システムへの広範囲な応用が可能である。

審 査 結 果 の 要 旨

電子計算機の急速な発達に伴い，情報処理の内容はますます多様化し，複雑なものとなっている。このため，より高度の処理能力をもつ計算機システムの設計法が強く望まれている。これに対処するには複数個のプロセスを有機的に結合し，能率よく処理を行う並列処理システムが必要であるが並列処理システムの統一的な設計理論は未だ確立されていない。著者はこの問題を研究し，並列処理システムにおける相互作用を制御する一般的方法を示し，並列処理システムの設計の基礎を与えた。本論文はこれらの成果をまとめたもので全編6章よりなる。

第1章は序論であり，本論文の意義と目的を述べている。

第2章ではまず並列処理システムの厳密な定義を与え，複雑なプロセスの動きを明確に表現する方法を示し，ついでプロセスの相互作用を3つのクラスに分類して，各相互作用の制御法について一般的に論じている。

第3章では第2章の結果を用い，2つのプロセスから成る処理系の個々の相互作用の制御法を詳細に調べ，制御に用いる同期基本命令数及びプロセスの待ち時間を夫々最小にする方法を導き，ついで，これを用いてプロセス間全体の相互作用の制御法を与えている。さらにこの方法によって構成された並列処理系はデッドロックのない系であることを示している。

第4章では第3章で得られた並列プログラム処理が非同期に実行されるとき，実行計算系列のすべてを求める有効な方法を示し，ついでこの処理系の最短実行時間，及びこれに対応する計算系列を求めるアルゴリズムを導いている。

第5章では第3章で考察した2プロセス並列処理システムの構成法をさらに一般のNプロセス並列処理システムに拡張している。まずNプロセス並列処理システムと2プロセス並列処理システムとが本質的に異なる点はローカルデッドロックであることを示し，ローカルデッドロックを検出するアルゴリズム，ローカルデッドロックを生じないための制御方法を導き，並列処理システムが全体としてデッドロックに陥らない効率よいシステムの設計法を与えている。

第6章は結論である。

以上要するに本論文は，将来の情報処理方式の基礎となる並列処理方式について研究し，複数個のプロセスを非同期的に同時に実行させる並列処理方式設計の基礎を与えたもので，情報工学の発展に寄与するところが少なくない。

よって，本論文は工学博士の学位論文として合格と認める。