

氏 名	はま だ まさ よし 濱 田 真 美
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	プログラムおよびデータのモジュール化に関する 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

電子計算機の利用方法が高度化するに伴い，大型のプログラムを実行する要求が強くなった。プログラムが実行される時，ある短時間内に使用されるプログラムおよびデータは小さな部分に集中する傾向が有る。従って，計算機系を有効に利用する目的で，仮想記憶と呼ばれる記憶管理方式が一般的になりつつある。当面使用される部分を高速記憶上に，残りを低速記憶上に配置し，必要に応じて両記憶間でデータ等を交換する。処理に中断は生じるが，処理内容には支障がない。この時，データ交換の頻度が高いと処理速度が低下する。速度低下を抑止する 2 つの独立な方法が考えられる。1 つは，処理系管理プログラムが記憶の割当て量を加減するシステム論的方法である。もう 1 つは，言語処理系を使ってプログラムおよびデータの局所性を増加させるプログラム論的方法である。

本論文では、後者の立場に立ち、プログラムおよびデータをモジュール化する方法とモジュールの表記能力を論じる。

プログラムに関しては、第2章から第4章において、実行の流れをグラフ論的に議論する。一般にはプログラムの流れとデータの参照関係は交錯するが、データの使用域が実行の滞留域と密接な関係を持ち、データ集合とプログラムが同時にモジュール化される。従ってデータ集合に対しては、それ内部での局所的配置が可能かを第5章で考察する。特に、局所性が乏しいと言われているポインタデータについて実験とモデルを対比しながら論じる。

第2章 流れ図の制御構造

第4章まではプログラムの動作モデルである流れ図形を対象とする。本章では、議論を進める際に必要となる基礎的事項について定義を行い、その性質を考察する。

第4章において流れ図形のモジュール化方法が示される。この時、帰着できるモジュールの大きさ n に応じて流れ図形を分類でき、LP n チャートと呼ばれる。LP n チャートで表記できる関数と従来研究されている流れ図形族で表記できる関数との比較によって、モジュールの大きさと表記できる計算の関係を表わす。表記能力の比較方法を定義すると共に、比較をより容易にする方法が議論される。すなわち、制御構造族間で表記能力を比較する場合には Ianov 流れ図形で議論できることが示される。更に Ianov 流れ図形における実行の推移関係をチャートと呼ばれるグラフで表現して取扱いを簡単にする。第4章でのモジュール化の議論はチャート上で行われる。

部分チャートを複製分離するチャート上の演算である nodes split を定義する。本論文では、新しい演算子を導入しない、かつ等しい関数を計算する、という制限によって定義された演算子非増加変換の下での変換可能性に従って表記能力を比較する。nodes split 変換はより制限の強い変換であるが、変換前後の構造に明確な対応関係が存在し、対象を限定できる。しかも両変換の下で表記能力が同等である制御構造族の条件が示される。第4章で扱われる流れ図族は全てこの条件を満たし、nodes split 変換下での表記能力比較結果が、演算子非増加変換の下での表記能力関係でもある。

第3章 EBJチャート族の表記能力

LPチャート族の表記能力は従来定義されているBJ (Böhm & Jacopini) チャート族とRE (Repeat End) チャート族の中間に位置する。BJチャート族はnodes split変換と演算子非増加変換でその表記能力結果が異なる。一方、BJチャート族のループ構造を拡張したEBJ (Extended BJ) チャート族を導入すると、それは両変換の下で同一の表記能力を持つ。

EBJ, LP, REチャート族に関しては第4章で表記能力を比較する。本章では、構造要素の構造上での包含関係および特定の解釈を与えて表記不可能な流れ図例を作る方法を用いて、BJ

チャート族とEBJチャート族の間の表記能力を比較する。

第4章 チャートのモジュール化

流れ図中を実行が移動する経路という観点に立つ時、流れ図形の複雑さをそれに含まれる述語の数で規定できる。更に局部的に見た複雑さを考える目的で、流れ図形Cを入口点と出口点が各々1個の部分である流れ図ブロックに分割し、各流れ図ブロック中の内部ブロックを全て関数呼出しで置換えてモジュール集合を作る。モジュールが含む述語数で最大の値をCのモジュールサイズと言う。与えられた流れ図を nodes split してモジュールサイズが最小の流れ図を作る問題について本章で論じる。ただし、流れ図のモデルであるチャートを使って議論を進める。縮約ブロックがモジュールに、複雑度がモジュールサイズに相当する。

チャートGに対する nodes split は分離する部分チャートの選び方と、それに入る孤の分割法に従った変形が存在する。また nodes split 回数は任意に可能である。従って、チャートGを nodes split して得られる複雑度が最小なチャート G_0 を求める有限手続きは自明でない。しかし確定的に G_0 を作る方法が以下の議論によって示される。

出点数が1である区画と呼ばれる部分チャートを選択し、区画の入口点毎に孤を分割するという制限を付けた nodes split を複連接化と呼ぶ。複連接化を実行することにより、区画がブロックとなってチャートの複雑度を減らせる。チャートの任意の部分チャートが2個以上の出点を持つ時、複連接(doubly coupled)であると定義する。nodes split の前後で複連接性が保存される重要な性質が有る。縮約ブロック全てが複連接であるチャートを既複連接化チャートと呼ぶ。既複連接化チャートは区画を含まず、複連接化が定義されない。有限回の複連接化で必ず既複連接化チャートが得られること、既複連接化チャートにおいて複雑度が極小であること、任意の複連接化過程で作られた既複連接化チャートが同形であり、途中に nodes split が加わると複雑度が増すことが示される。ゆえに、任意の区画に対して複連接を行って作った最初の既複連接化チャートが最小の複雑度を与える。これが最小なモジュールに分割する方法である。

nodes split によって得られる最小複雑度 n を使ってチャートを分類し、 LP_n チャートと呼ぶ。入口点のみが異ってチャート本体が同一な複連接チャート G_1, G_2 は、頂点数がより小さい F 中に G_2 を入れ子とした構造へ G_1 を変換できる。従って、(1)複連接で、(2)頂点数が n 以下で、かつ(3)チャート本体が異なるチャート集合 $\epsilon(LP_n)$ の元を入れ子にして、任意の LP_n チャートの計算を記述できる。

基本構造要素が計算の特性を内包するので、 $\epsilon(LP_n)$ の性質を利用してEBJ, LP, REチャート族間の表記能力を比較した(図1)。

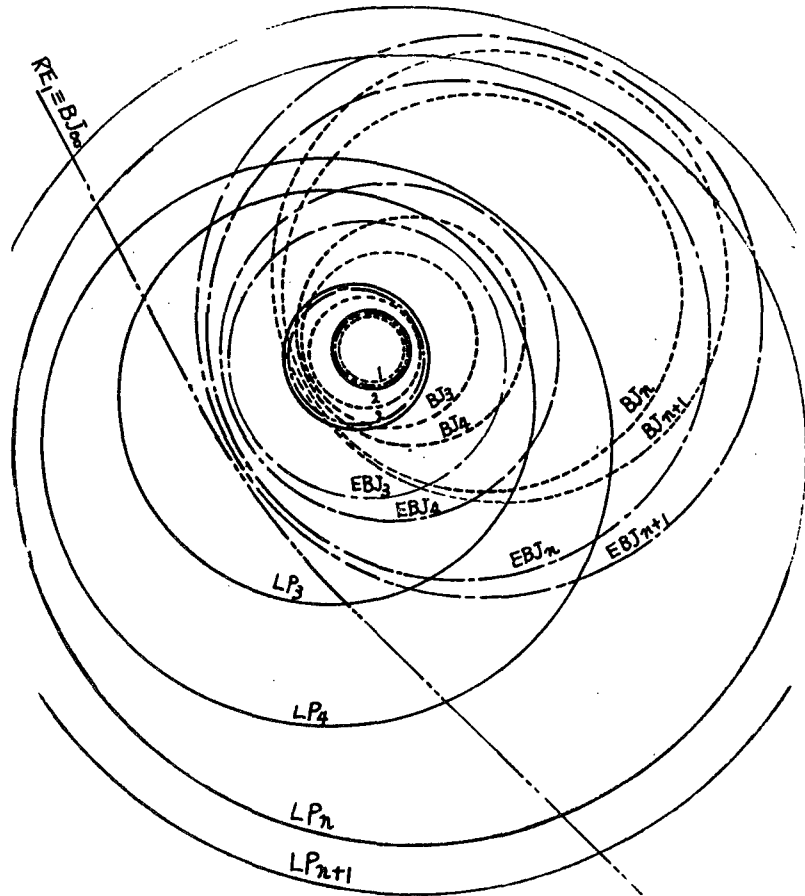


図1. BJ, EBJ, LP, RE チャート族の表記能力比較

第5章 データの局所性

データ型の違いによって、データの局所化配置に難易が生じる。特に、ポインタデータは参照関係が動的に決定されるために、局所化については問題を含む。本章においては、ポインタデータ処理系として代表的なLispインタプリタを例として、実験から得た測定値と理論的モデルに基づく解析を比較しながらポインタデータの局所化について議論する。

ポインタデータの配置機構から、指示距離に対し指数的に指示密度が減少するモデルを設定できる。このモデルから得た指示距離分布の解析結果と実測値が一致する。従ってポインタデータも局所性を持つと言える。ポインタを局所化するコンパクションの効果について、このモデルを利用して解析し実測と比較した。更に指示距離分布の式から記憶交換要求頻度をも計算した。

ポインタ処理系では記憶交換と共に、廃セル回収にも無駄時間を費す。また回収時にコンパク

ションが実施されるので、回収頻度が局所化頻度と一致する。以上の理由から回収頻度および回収時間に関して解析を行っている。

第 6 章 結 論

計算機系の有効的利用を目標として仮想記憶方式の記憶管理法が普及した。本論文では、仮想記憶系での処理速度を向上させる一方法であるプログラムおよびデータを局所化、モジュール化する方法について論じた。プログラムに関してはプログラム形理論およびグラフ理論的に、データに関してはモデルによる解析と実験で得た結果を対比して議論した。

並列処理を行える計算機が実用段階に近づいている。この時各処理装置に仕事を分担するかが問題となる。並列性を意識せずにプログラミングできて、言語処理系が自動的に並列分割するのが理想であろう。この時、並列に実行される仕事間には相互作用が少ない要請があり、本論で言うモジュールに対応する。データ流解析をも加味する必要があるが、本論でのモジュール分割法を並列処理用プログラム分解等、広範囲のプログラム処理に応用できると思われる。

審査結果の要旨

情報処理システムの普及に伴い、極めて大規模なソフトウェアシステムを必要とする処理が多くなり、このため、計算機システムに巨大な記憶装置が要求されるようになってきた。しかしながら計算機システムの高速記憶装置は高価なため、能率の良いシステムを構成するためには、低速大容量記憶装置を付加し、それと高速記憶装置間の情報の授受を巧みに行う処理方式が必要となる。これに対処する有効な方法は、プログラムとデータをできるだけ局所的にモジュール化し、高速記憶装置と低速記憶装置間の情報の授受の回数を少なくすることである。著者はこの立場から、プログラム及びデータのモジュール化により、効率よいシステムを設計するための基礎的研究を行った。本論文はその成果をまとめたもので全編6章より成る。

第1章は序論である。

第2章では従来の研究と本研究の基礎となる諸概念を説明し、第3章以降で用いる新しい用語の定義を与えている。

第3章では著者により与えられたEBJチャート族とプログラムスキーマの中で基本的な役割を果すBJチャート族間の関係を調べ、各チャート族間の記述能力の比較を行っている。

第4章ではプログラムをモジュール化したときの複雑度について考察し、プログラムの複雑度が本論文で新しく導入したLPチャート族によって直接表現できることを示している。ついで、与えられた流れ図を第2章で示したノード分割法を用いて逐次変形し、プログラムの複雑度を最小にするアルゴリズムをグラフの複連接性の性質を用いて与え、プログラムの最小ブロック化を行う方法等を示している。

第5章ではデータのモジュール化を行うため、情報処理の中で基本的な役割を果すリスト構造をもつデータの局所性を理論的及び実験的に調べ、その局所性を明らかにしている。ついでリスト処理系の動作について評価を行い処理に伴って生ずるガベージコレクションの時間、コンパクトションの効果等を詳細に調べている。

第6章は結論である。

以上要するに本論文は、仮想記憶管理方式を能率よく行うために不可欠なプログラム及びデータのモジュール化に関する研究を詳細に行い、大規模な計算機システムにおけるソフトウェア設計の上に重要な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。