

## 論文の内容の要旨

論文題目 **Homomorphism-based Structured Parallel Programming**  
(準同形に基づいた構造化並列プログラミングに関する研究)

氏名 江本 健斗

近年、計算機に処理させたい問題の巨大化や、安価な並列計算機環境の普及に伴い、それら並列計算機を有効に活用するための並列プログラミングがプログラマにとって必須の仕事となってきている。しかしながら、並列プログラミングでは、これまでの逐次プログラミングには必要の無かった、計算対象のデータの複数計算機間での分散や、計算の仕事の分散と同期とを考慮する必要がある。そのため、並列プログラミングには、大きく以下の二点の難しい問題がある。ひとつは、問題を並列計算機で効率的に処理することが可能な並列アルゴリズムの開発が難しい。そして、もうひとつは、そのアルゴリズムを並列プログラムとして実装することが難しい。

これらの問題を解決するための枠組みのひとつとして、構造化並列プログラミング（スケルトン並列プログラミングとしても知られている）が提唱されている。この枠組みでは、スケルトンと呼ばれる並列計算の基本パターンを組み合わせる事で、プログラマは並列プログラミングを行うことができる。言い換えれば、並列プログラムをスケルトンで構造化することで、アルゴリズムの開発や実装の難しさの解決を図っている。並列処理の為に考慮すべき難しい並列性がスケルトンにより隠蔽されるため、この枠組みによる並列アルゴリズムの作成は、スケルトンの逐次的な組み合わせでアルゴリズムを設計する事にあたる。また、アルゴリズムの実装は、各スケルトンの並列実装を設計した組み合わせどおりに呼び出すだけとなり、容易である。

しかしながら、効率の良いアルゴリズムを系統的に作成できるような構造化並列プログラミングの枠組みを構築する事は、未だ大きな課題として残されている。スケルトンの組み合わせによるアルゴリズムの記述は容易であるが、一方で、愚直に組まれたアルゴリズムは組み合わせに含まれる無駄による非効率性の問題を持つ。例えば、連続するスケルトン間でやりとりされる中間データ構造がしばしば無駄である。そのため、組み合わせの無駄を省いて効率の良いアルゴリズムを合成できる枠組みが必要とされている。

本論文では、スケルトンの組み合わせの融合変換による最適化に注目して、効率の良いアルゴリズムを系統的に作成できるような構造化並列プログラミングの枠組みを提案する。我々の枠組みでは、融合の為の良い性質を持つ、準同形と呼ばれる特定の形の計算パターンでスケルトンを構造化する。このスケルトンの構造化により、我々の枠組みでは、愚直なスケルトンの組み合わせで記述された並列アルゴリズムから、融合変換によって効率の良い並列アルゴリズムを得ることができる。本論文の主な貢献は以下のとおりである。

- (a) 準同型に基づく融合変換に適した並列スケルトンの設計。
- (b) 設計されたスケルトンに関する融合規則による最適化の理論。
- (c) 設計されたスケルトン及び最適化の実装による理論の効果の検証。

特に、二次元配列を操作対象とするスケルトンの設計、それらを用いたプログラミング、及び、それらの最適化は、大きな課題とされていた問題である。本論文の成果により、二次元配列を操作対象とした構造化並列プログラミングの議論が可能になった。また、他のデータ構造（一次元配列）に関しても、いつかの融合変換に基づく最適化を定理の形で与え、より幅広い問題を最適化可能になった。さらに、愚直なプログラム記述から効率的なアルゴリズムを得るための自動最適化の仕組みを提案し、実際のプログラミング言語上に高レベルな最適化を実装したライブラリを構築した。この仕組みを用いることにより、プログラマは最適化理論の成果を容易に利用可能となる。

本論文の構成は以下の通りである。

第一章では、本論文の背景、目的、貢献を要約し、本論文の構成を述べる。

第二章では、スケルトン設計の元となる、操作対象のデータ構造を表現する代数系とその準同形が導入される。データ構造を代数系として捉え、その上の準同形を計算の基本パターンとすることで、単純な計算の組合せによる複雑な計算の記述が容易になる。また、それらの代数系に並列性の為の自由度を持たせることにより、準同形が並列スケルトンの基

礎として利用できる。本章は、特に二次元配列に重点を置きつつ、一次元配列と木に関する代数系と準同形を導入する。

第三章では、前章で導入した準同形を基に、並列スケルトンが定義される。準同形の性質を受け継ぐスケルトンは、自身は単純な計算パターンであるが、それらの組み合わせにより様々な計算を表現可能である。本章では、スケルトンで記述された様々な並列プログラムを通し、スケルトンの組み合わせによる記述力も示される。また、幾つかの行列の基本演算が定義されたスケルトンで記述可能であることが示される。

第四章では、本論文の提案するスケルトンの特徴である、融合変換による最適化の理論が構築される。まず、準同形の融合変換が導入され、それを基礎としてスケルトンの融合変換が構築される。また、それらの融合変換を用いて、愚直なスケルトンの組合せで記述されたアルゴリズムから、効率的なアルゴリズムを得ることができることが、具体的な問題例に関して示される。

第五章では、特定の問題クラスに特化した融合変換規則の構築や、効率的なアルゴリズムを与える最適化定理を構築する。これらは、準同形の融合変換の一般性から生じる変換の難しさを解決し、スケルトンプログラムの自動的な最適化を可能とする基盤を与える。本章で対象とする問題クラスは、入力の興味のある部分構造を生成し、それら部分構造の集まりを処理して結果を得るような計算問題である。例えば、二次元配列を入力とし、その中の矩形領域を考え、それらから最大の和を与えるものを探し出す、という問題（最大部分矩形和問題）が対象に含まれる。

第六章では、前章までに構築した理論を実際に実装し、本論文の提案する構造化並列プログラミングの枠組みで並列プログラムが現実に関数として動かせることが示される。本章ではふたつのスケルトンの実装が示される。ひとつはC++上の関数としてスケルトンの実装を提供するライブラリである。もうひとつは新しいプログラミング言語 **Fortress** 上のライブラリである。後者の実装では、スケルトンの理論を基礎にしたより使いやすい表記法との融合や、スケルトンの組合せにマッピングされた並列プログラムの自動的な最適化の手法が示される。この実装により、ライブラリが並列アルゴリズムの動的な辞典として機能する。

第七章では、本論文の内容がまとめられ、さらに、今後の展望についての提案がなされる。