

## 審査の結果の要旨

氏名 尾上能之

本論文は、信頼性の高い計算機プログラムを開発する手法として提案されている構成的アルゴリズム論に基づくプログラミング方法論を実践するにあたり、プログラムの効率を改善するためのプログラム変換技術を扱ったものである。一般に、プログラムはモジュールで構成されるが、モジュール間で受け渡されるデータを除去して実行効率のよいプログラムを得るための実用的な方法を確立する必要があり、そこでは、理論的な成果である融合変換を実用的なコンパイラに実装して、その効果を評価することが課題となる。本研究は、その課題に対して、必要とされるアルゴリズムの開発とそれを既存のコンパイラに実装する手法、さらに、実用規模のプログラムに対するベンチマークによる評価を与えたものである。

第1章「序論」では、本論文で扱う関数型言語におけるプログラム変換について概説している。研究の背景として関数型言語を用いる利点であるプログラムの簡潔さやモジュール性を述べた後、短所である実行効率について言及し、それを改善するための既存研究を紹介している。とくに、そのような効率改善の一手法である融合変換(Hylo 融合変換)を実在するコンパイラに実装し、その効果を評価した本論文の貢献について要約している。

第2章「予備知識」では、関数プログラミングの基本的な概念や構成的アルゴリズム論に関する研究を概観し、構成的アルゴリズム論を用いて、プログラム中に内在する再帰構造を抽出しそれを操作する Hylo 融合変換の特徴に言及している。さらに、本論文中で用いる記法や背景となる定理を示している。

第3章「運算的融合変換」では、Hylo システムを実現するために必要ないくつかの重要なアルゴリズムを提示している。これらのアルゴリズムは従来の研究では示されていなかったものであり、自動変換システムを作る際の障壁となっていた部分でもある。ここではとくに、データ生成と消費を捕獲する多様型関数を導出するためのアルゴリズムに焦点をあてている。Hylo 再帰関数の導出や展開に関するアルゴリズムは、既存研究を拡張したものとなっている。

第4章「融合変換の実装」では、Hylo 融合変換の処理を、実際に広く用いられている関数型言語のコンパイラ内部に実装した方法について述べている。独自のシステムを構築するのではなく、既存コンパイラに埋め込むことによって、これが広く一般的に利用され、実用的な規模の例題に対して融合変換の有効性

を検証することが可能になる。また、WWW 上で、融合変換を埋め込んだコンパイラを利用できるインタフェースを用意し、融合変換の効果を視覚的に表示できる仕組みの実現についても触れている。

第5章「融合変換の性能評価」では、まず、インタプリタを基にしたプロトタイプ上で、プログラムを人手で融合変換したものの実行効率の比較実験を行なって融合変換の有効性を検証した結果を示している。ついで、前章で述べたコンパイラを用いて実用的規模のプログラムによるベンチマークでその長所と短所を広く調べている。そこでは、Hylo 融合変換と、別に提案されている `foldr/build` 融合変換の特長を比較し、後者の適用範囲よりも広い範囲のプログラムに対して Hylo 融合変換が適用され、効果的であると主張している。

第6章「関連研究」では、中間データとして受け渡しされるデータ構造を除去するための手法を広くサーベイしている。これらの手法は、`fold/unfold` 変換に基づく手法、リストのデータ構造に着目した `foldr/build` 法、それを一般の再帰データ型に拡張した本研究の中心的な課題である Hylo 融合変換、の3つに大別され、これらの長所短所の比較に加えて、それらの関係を明確にし、そのいくつかは型推論に基づく解析によって変換後の効率が向上することを示している。

第7章「結論」では、論文の全体を総括し、本論文の主な貢献である構成的アルゴリズム論に基づいた運算的融合変換のアルゴリズムが実在する関数型言語のコンパイラに対しても十分に利用可能であることを確認している。また、今後の課題として、現在のシステムにおける実装上の課題や Hylo 融合変換と他手法との組み合わせた場合の相乗効果などに触れている。

以上、これを要するに、本研究は関数プログラムの融合変換に関する理論的な成果をもとに、それを実用的なコンパイラに実装するためのアルゴリズムを開発し、それを用いたコンパイラによって、融合変換の効果を実用規模のプログラムに対して実証したものであり、情報工学の研究に貢献するところ大である。よって本論文は博士(工学)の論文として合格と認められる。