

論文審査の結果の要旨

氏名 高野 明彦

ソフトウェアの生産性と信頼性を飛躍的に向上させるためには、信頼性の保証された共通ソフトウェア部品を整備し、それらを組み合わせて高信頼で高性能のソフトウェアを作成する新しいプログラミング方法論の確立が求められている。

本論文は、このような背景のもとで、汎用部品プログラムを組み合わせで記述されたプログラムを、機能的には等価で、より効率の良いプログラムへ自動的に変換する方法について述べている。本論文では、関数プログラムを対象とする3つのプログラム変換法を提案しており、それぞれ第2章、第4章、第5章で定式化され、その適用範囲や変換例について述べられている。

第2章では「関数プログラムの一般部分計算法」を提案している。一般部分計算法とは、プログラム解析に定理証明系を用いる新しい部分変換法であるが、本論文では、遅延評価の関数プログラムを対象に一般部分計算法を定式化している。変換例として、素朴な文字列検索プログラムから KMP アルゴリズムと同等の効率の良いプログラムを導出している。さらに、射影 (projection) により表現された評価要求情報を用いて無駄な変換を避ける方法を示している。本変換法は、プログラム解析に抽象実行を用いる従来の部分計算法よりも高い変換能力を持っている。

第4章では「構成的アルゴリズム論に基づく Shortcut Deforestation」について述べている。中間データ構造として使われるリスト構造を自動的に除去するプログラム変換として、foldr-build 融合規則に基づく Shortcut Deforestation が知られている。本論文では、その foldr-build 融合規則を一般の代数データ型に拡張した酸性雨定理 (Acid Rain Theorem) を証明した。本定理は、foldr-build 規則と双対な場合、いわば unbuild-unfold と呼ばれるべき規則の拡張も含んでいる。さらに、Hylomorphism と呼ばれる再帰構造が、プログラム融合変換に好適なプログラム表現方法であることを示し、それを3つ組で表す新たな表示方法を提案している。3要素はそれぞれ、消費するデータ構造の分解、分解されたデータ成分ごとの変換、生成するデータ構造の構築に対応している。この表現により、酸性雨定理を3つ組表現された Hylomorphism の融合規則として捉え直し、それらに基づく融合変換法を定式化している。このように定式化された

融合変換法は、Shortcut deforestation 等の従来法よりも強力な融合変換法であることが、変換例を通じて示されている。

第5章では、本論文で提案する3つ目のプログラム変換法である「文脈保存条件を用いた並列プログラムの系統的導出法」について述べられている。逐次プログラムから並列プログラムの導出を一種のプログラムの一般化と見なし、4ステップからなる並列化方法を示している。変換対象とする線形自己再帰関数 (Linear Self-Recursive Functions) というクラスを定義し、プログラム中での再帰呼出しの類似性を議論するための概文脈 (Skeltal Context) という概念を導入し、再帰構造により概文脈がどのように複写され保存されるかについて述べている。Cons-List 上の Catamorphism を表す逐次プログラムが、Append-List 上の並列プログラムへ変換できるための十分条件を、再帰呼出しの概文脈保存条件の形で与えている。前節の定理の条件を満足する概文脈を4種類 (Simple, Nested, Conditional, Tupled) 与え、それぞれどのような並列プログラム構造へ変換されるかについて述べている。本変換法は、従来の自動並列化法では取扱いが困難なプログラムについても適用可能な系統的な変換法である。

以上のように、本論文は、関数プログラムに基づくプログラム作成法に関して、部品プログラムの保持に好適な表現方法を新たに提案するとともに、それらから構築されたプログラムの性能を改善するプログラム変換法を示して、その有効性を示したものである。

なお、本論文の第4章は Erik Meijer 氏との共同研究、第5章は Wei-Ngan Chin 氏、Zhenjiang Hu 氏との共同研究であるが、論文提出者が主体となって分析及び検証を行ったものであり、論文提出者の寄与が十分であると判断する。

したがって、博士 (理学) を授与できると認める。