

## 研究課題別評価

1. 研究課題名：プログラムの性能を理論的に考慮したコンパイラの検証と構築

2. 研究者氏名：南出 靖彦

3. 研究の狙い：

プログラムが期待したように動作することを保証するためには、ソース・プログラムからわかるプログラムの実行時間、メモリーの使用状況などプログラムの性能に関連した様々な性質が、実行コードでも成り立つ必要がある。そこで、本研究では、プログラムの性能を理論的に扱い、性能を考慮した上で期待したように動作する実行コードを生成するコンパイラを構築できるようにする。そのために、まず、プログラムの性能を理論的に議論するためのプログラミング言語の意味論を確立する。その上で、コンパイラで用いられる様々なプログラム変換が、性能に関して期待される性質を満たしていることを厳密に証明していく。

また、コンパイラの性質の証明を厳密に行う場合には、その証明は非常に複雑かつ冗長なものになり、今度は、その証明自体の正しさを確認することが非常に困難になると思われる。そこで、本研究では、プログラム変換の正当性をより確実に示すために、証明をコンピュータのプログラムとして実現されている定理証明システムで行い、証明の検証を行うことも目指す。

4. 研究結果：

コンパイラの性能に関する正当性を議論するためのプログラムの意味論を構築し、正当性の基準を示した。その意味論の上で幾つかのプログラム変換について実行時間や必要な記憶領域について正当性を証明し、正当性の証明法を明らかにした。さらに、プログラムの性能に関する性質を型情報として推論し、その情報に基づきプログラムをコンパイルする手法を提案した。また、プログラム変換の正当性の証明を定理証明システムで行う場合の幾つかの問題点を明らかにし、その解決策を示した。以下にそれぞれの研究の成果を述べる。

### 1) プログラムの意味論と正当性の基準

実行時間、スタック空間、ヒープ空間をモデル化した操作的意味論を与えるとともに、コンパイラの正当性の基準として幾つかの性質を定義した。実行時間、スタック空間に関する意味論は、既存の意味論の拡張として得ることができたが、スタック空間に関してはコンパイラによる実行、インタプリタによる実行のどちらをモデル化するかで、2種類の意味論が得られた。この2種類の意味論については、スタックの大きさがプログラムの大きさと関係づけられることを示した。ヒープ空間に関しても、幾つかの意味論を与えたが、まだ、どの意味論が正当性の証明に適しているか十分に議論できていない。

また、コンパイラが満たすべき性能に関する正当性の基準として、性能を定数倍の範囲で保存する、計算量を保存する、など幾つかのレベルの性質を定義した。これらの基準の中で、どの基準が適当であるかは、プログラミング言語や議論する性能の種類（実行時間や記憶領域）などに依存することを示した。

## 2) 正当性の証明法

プログラム変換の正当性の証明法及びプログラミング言語の型システムの正当性の証明法が、プログラム変換の性能に関する正当性の証明法に拡張できることを示した。型付のプログラミング言語のプログラム変換の正当性の証明で用いられる論理関係 (logical relation) を拡張し、実行時間、スタック空間に関する正当性を証明した。ヒープ空間に関しては、より詳細な議論が必要になり論理関係を用いることはできなかった。しかし、参照型を持つプログラミング言語の型システムの正当性の証明法に基づき、ヒープ領域に関する正当性の見通しのよい証明を与えることができた。

## 3) 型情報に基づくコンパイル法

最悪の場合の性能を保証するようにコンパイラを作ると高性能な実行プログラムを生成することが困難になる。この問題の一つの解決策として、プログラムの性能に関する情報をプログラムの型として推論し、その情報に基づきプログラムをコンパイルする方法を提案した。この方法に関数型プログラミング言語 ML から Java のバイトコードへのコンパイラに実装し、末尾呼び出しを効率的に、かつ性能に関して正当に実現できることを示した。

## 4) 定理証明システムによる検証

関数型プログラミング言語のコンパイラで用いられる CPS 変換と呼ばれるプログラム変換の検証を Isabelle/HOL と呼ばれる定理証明システムで行った。この検証で、変換によって導入される変数の扱いとプログラムのアルファ同値の扱いが問題になることがわかった。これらについて定理証明システムによる検証に適した形式化を見つけ、検証を行った。この形式化によって、CPS 変換の正当性の検証では、多くの補題の証明を自動化することができた。

## 5. 自己評価：

性能を理論的に考慮したコンパイラの検証と構築に関して、様々な面で研究を進めることができた。コンパイラで用いられるプログラム変換の性能に関する正当性の証明法に関しては、いくつかのプログラム変換の正当性の証明を行い、どのように正当性を証明するか明らかにできたと考えている。特に、ヒープ空間に関する正当性の証明は、これまで、非常に複雑になり困難であると考えられていたが、本研究で十分に可能であることを示せたことは大きな成果であったと考えている。しかし、本研究で用いた証明法を容易に他のプログラム変換に応用するためには、さらに証明法の体系化が必要であると考えている。

研究構想では、Java などオブジェクト指向言語のコンパイラに研究成果を応用していくことを目指していた。しかし、さきがけ研究の中では、関数型プログラミング言語のコンパイラで用いられるプログラム変換の検証のみしか行うことができなかった。しかし、本研究の成果は Java などの高度なコンパイル技術が必要となる言語のコンパイラへ直ちに应用できるものであると考えている。

定理証明システムに関する研究では、コンパイラで用いられるプログラム変換の検証が比較的容易にできることを示せた。このことから、コンパイラ全体を定理証明システムで検証することも可能であると考えており、今後、そのような方向で研究を進めれば、完全に検証されたコンパイラが構築できるようになると考えている。

6. 研究総括の見解：

性能について正しいコンパイラを構築する上で重要となる基礎理論について研究成果をあげており、高く評価できる。ソフトウェアの信頼性の向上は、ますます、重要になってくる課題であり、本研究はコンパイラの信頼性向上からこの課題に大きく貢献するものである。正しさが検証されたコンパイラを構築する道筋ができつつあり、今後の研究の推進に期待できる。

7. 主な論文等：

1. 大熊浩示, 南出靖彦, 定理証明システムを用いた CPS 変換の正当性の検証, ソフトウェア科学会第 17 回大会, 2000.
2. Yasuhiko Minamide, A New Criterion for Safe Program Transformations, In Proceedings of the Forth International Workshop on Higher Order Operational Techniques in Semantics (HOOTS), ENTCS 41, 2000.
3. Yasuhiko Minamide, Runtime Behavior of Conversion Interpretation of Subtyping, In Proceedings of the 13th International Workshop on the Implementation of Functional Languages, LNCS 2312, pages 155-167, 2001.
4. Yasuhiko Minamide, Selective Tail Call Elimination, In Proceedings of the 14th International Workshop on Implementation of Functional Languages, 2002.