様式 C-7-2

自己評価報告書

平成２３年４月１２日現在

機関番号：１４３０１
研究種目：基盤研究（C）
研究期間：２００８　～　２０１１
課題番号：２０５０００１１
研究課題名（和文）　形式的に検証可能なプログラム変換フレームワークの構築

研究課題名（英文）　Development of formally verifiable framework for program transformation

研究代表者
西村 進（NISHIMURA SUSUMU）
京都大学・大学院理学研究科・准教授
研究者番号：10283681

研究分野：プログラム言語理論
科研費の分科・細目：情報学・情報学基礎
キーワード：プログラム理論、プログラム変換、形式的証明

１．研究計画の概要

プログラム変換は以前から研究されてきているテーマであるが、これまで提案されてきた個々の変換手法が本当に正しいかどうかは必ずしも明らかではない。特に、プログラム変換の対象をポインタ操作・大域的制御・並行計算プリミティブなどの実際のプログラムに含まれるものにまで拡張したときについては研究が進んでいない。正しさが必ずしも保証されないということは、プログラム変換を適用することによってプログラムがうまく動かなくなってしまう可能性があるということであり、これは高度な安全性の要求される分野においては絶対許されないことである。

本研究は、プログラム変換が正しいこと、すなわち意図したプログラムの動作がプログラム変換の前後で変化しないことを保証するため、プログラム変換工程の形式的検証のためのフレームワークを定理証明支援系等の上に構築することを可能とするような理論的基礎を確立することを目的とする。このような目的を達成するため、プログラム単体の意味(プログラムの動作)および変換前後のプログラムの対応関係を厳密に定義し、これらを形式的に記述できるようにする。このため、本研究では、形式的技法(formal methods)の分野で仕様からプログラムを導出するための詳細化と呼ばれる技法を基本とし、これをプログラム変換の過程において適用することによってプログラム変換の正しさを保証するフレームワークを構築する。具体的には、詳細化技法の基礎となっている述語変換子意味論を、従来の古典論理による形式化から拡張することによって、上記のような実際的なプログラミング概念に対応する。

２．研究の進捗状況

(1) ポインタ操作やメモリ管理を含むプログラムの変換

プログラムの段階的詳細化を行うための形式的な枠組である refinement calculus を、従来の古典論理に代えて、メモリに関する言明を表現可能な separation logic を用いてプログラムを述語変換子として表すことにより実現した。この手法により、メモリの確保と開放に関する操作が、separation logic の論理演算子に対応する基本的な述語変換子の組み合わせに分解できることを明らかにした。これら基本的な述語変換子が互いに打ち消しあう性質を持つことを利用して、プログラム変換の正しさを導出可能であることを示した。

(2) 例外の発生と補足を許すプログラムの変換

従来の refinement calculus の枠組みは、プログラムが正常に終了するかどうかの区別しかできないが、実際のプログラムでは、実行時エラーによって実行を異常終了させたり、異常終了を補足して正常実行に復帰する仕組みが用意されている。古典論理に代えて４値論理を用いることによって正常終了と異常終了を区別し、上記のような実行時エラーやエラーからの回復機構をもつプログラムについてプログラムの段階的な詳細化を形式的に導出することが可能となった。

(3) XPath の等価性証明(共同研究)

XPath は、構造化文書 XML におけるデータ検索を構造に関する相対的な位置関係によって行うためのプログラミング言語の一種である。XPath の式をその入力と出力の間の

2 項関係として表現することにより、XPath の包含関係(プログラムの詳細化に対応する)および等価性（ふたつの関係が相互に包含することによって示せる）を、関係に関する少数の代数規則を運用することにより示す枠組みを提供した。

３．現在までの達成度
②おおむね順調に進展している。
実際的なプログラミング要素を含むプログラム変換の正当性を保証する技法については、当初予定していた Refinement calculus のポインタおよびメモリ管理操作への拡張だけでなく、例外処理や XPath の等価性判定についても達成され、予定以上の成果が出ている。一方で、これら成果に基づいたフレームワークの構築は進展がやや遅れ気味である。

４．今後の研究の推進方策
これまでの成果をより発展させるため、並行計算プログラム(特に並行計算プリミティブを直に扱う必要のある細粒度並行計算プログラム)にこれまでの研究で得られた知見の適用を試みたい。細粒度並行計算プログラムの形式化とその正当性の検証は、プログラム変換の形式的検証がもっとも望まれる分野であるが研究が遅れている課題であり、これが実現したときの収穫は大きいものがある。またこれと同時に、これまでの基礎的な研究成果に基づいたフレームワークの構築も進めていく。

５． 代表的な研究成果
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計２件）
① Susumu Nishimura, “Refining Exceptions in Four-Valued Logic”, 19th International Symposium, LOPSTR 2009, Revised Selected Papers (Lecture Notes in Computer Science), 査読有, vol. 6037, 2010, pp. 113-127.
② Shin-ya Katsumata, Susumu Nishimura, “Algebraic Fusion of Functions with an Accumulating Parameter and Its Improvement”, Journal of Functional Programming, 査読有, vol. 18(5-6), 2008, pp. 781-819.

〔学会発表〕（計３件）
① Susumu Nishimura, “Calculating Tree Navigation with Symmetric Relational Zipper”, The 20th ACM SIGPLAN 2011 Workshop on Partial Evaluation and Program Manipulation (PEPM'11), 2011 年 1 月 25 日, Austin, Texas, アメリカ合衆国
② Susumu Nishimura, “Refining Exceptions in Four-Valued Logic”, 19th International Symposium on Logic-Based Program Synthesis and Transformation LOPSTR 2009, 2009 年 9 月 10 日, Coimbra, ポルトガル
③ Susumu Nishimura, “Safe Modification of Pointer Programs in Refinement Calculus”, International Conference on Mathematics of Program Construction (MPC '08), 2008 年 7 月 15 日, CIRM, Marseille, フランス

〔図書〕（計０件）

〔産業財産権〕
○出願状況（計０件）

○取得状況（計０件）

〔その他〕

http://www.math.kyoto-u.ac.jp/~susumu/mpc08pvs/index.html