様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成 24 年 5 月 11 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

研究成果の概要（和文）：ポインタ操作や例外処理のような進んだ計算機構を含むプログラムに対してプログラムの段階的詳細化の手法を拡張することによって、プログラムの正しさを保証しながらプログラム変換を行う手法を与えた。論理体系としては、ポインタ操作には separation logic, 例外処理には 4 値論理がそれぞれ対応することを示し、計算機上でそれぞれの論理体系に基づく形式的証明を行うことによってプログラム変換の正当性を保証することができることを示した。

研究成果の概要（英文）：We have developed a method, which is an extension of the stepwise refinement method, for correctness-preserving transformation of programs that make use of advanced features such as pointer manipulations and exceptions. We have shown that each feature has a corresponding logical system, namely, separation logic for pointer manipulations and four-valued logic for exceptions. It has been shown that the correctness of programs is formally guaranteed by developing formal proofs on computers, based on each particular logical system.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2008 年度	1,300,000	390,000	1,690,000
2009 年度	700,000	210,000	910,000
2010 年度	700,000	210,000	910,000
2011 年度	700,000	210,000	910,000
総　計	3,400,000	1,020,000	4,420,000

研究分野：総合領域
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．研究開始当初の背景

プログラム変換は古く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。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許されないことである。

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フレームワークとして refinement calculus の手法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

できるのではないかと考えた。形式的技法では、プログラムの仕様から始めて、これに詳細化の規則を繰り返し適用することによって目的プログラムを導出するが、このような段階的詳細化の過程で現れる途中結果は仕様でもプログラムでも良い。したがって、上記の手続きを仕様からではなくプログラムから始めることによって、元のプログラムに対して正しさを保証しながらプログラム変換を行うことができる。
このような詳細化関係を形式的定理証明系の上で定義することによって、プログラムの正しさを破壊してしまうかもしれないプログラムの変換手続きを補足し、これがプログラムの動作に影響するかどうか検証することによって変換の正しさを厳密に論証することができる。

２．研究の目的

(1) プログラム変換が正しいことを保証するための手法の確立
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の理論的基礎を確立する。特に、ポインタ操作などの従来のプログラム変換では正しさの保証が難しかったプログラムを扱う手法を開発する。
(2) 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の構築
上記理論に基づいたプログラム変換を形式的定理証明系上で実現する。これによって、プログラム変換及び変換の適用過程の正しさを形式的に検証可能なものとする。

３．研究の方法
従来の Refinement calculus が対象とするのはごく基本的な計算機構の組み合わせで書かれたプログラムに限定されている。これをより複雑な論証を必要とするような計算機構を含むようなプログラムにも適用できるように拡張する。特に、ポインタ操作や例外処理の機構を含むプログラムについて重点的に拡張を試みる。
ポインタ操作や例外処理などの進んだ計算機構を扱うには、refinement calculus の基礎となる論理体系の拡張が本質的に必要である。それぞれの計算機構に即した論理体系の拡張を検討し、これらを refinement calculus の体系と統合する。
拡張された論理体系と統合された refinement calculus のフレームワーク上で、正しさの保証されたプログラム変換が可能であることを形式的証明系の上で、実際にプログラム変換の段階的詳細化を形式的に行うことによってこの手法の有効性を示す。

４．研究成果
(1) ポインタ操作やヒープ領域の確保や開放を含むプログラムに関するプログラム変換を形式的に導出可能な枠組みを、refinement calculus の拡張として与えた。この拡張では、従来の古典論理に代えて、ヒープ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した。この手法により、ヒープやポインタに関する操作が、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は、ヒープの確保と開放に対応するものであり、これらを用いて他のポインタ操作を表現できるだけでなく、これらがお互いに打ち消しあう性質を持つことを示した。このような基本的な述語演算子を用いて表されたプログラムに、いくつかの変換規則を繰り返し適用することにより、ヒープやポインタ操作を含むようなプログラムのプログラム変換がその正しさを保障した形で導出可能であることを示した。
　この結果を、国際会議 Mathematics of Program Consruction (MPC'08)において発表した。発表論文で示した、プログラム変換を計算機上の形式的証明系で具体的に導出する例はホームページ（下記参照）からダウンロード可能である。

(2) 例外の発生と例外の補足を許すようなプログラムに関するプログラム変換を形式的に導出可能な枠組みとして、４値論理に基づく refinement calculus を提案した。Refinement calculus は、古典論理をその基礎としており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。実際のプログラムでは、実行時エラー(例えばゼロ除算など)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が、これらは従来の Refinement Calculus では扱うことができなかった。今年度の研究では、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て扱うことを可能とし、これによって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。
　この結果を、国際シンポジウム Logic-Based Program Synthesis and Transformation (LOPSTR 2009)と国内研究集会において発表した。国際シンポジ

ウムで発表した内容はシンポジウムの selected paper として選定され出版された。

(3) 上記の研究成果から発展した内容として、研究当初の構想には含まれていなかった計算機構を含むようなプログラム変換にも同様の手法が適用できるようにするための理論的基礎付けとして、次のような研究成果が得られた。

① 構造化文書 XML において、文書中におけるデータを XML の構造に関する相対的な位置関係で指定するための標準的な言語である XPath に関して、その等価性を形式的に推論するためのフレームワークを与え、いくつかの自明でない等価性を導出できることを示した。(池田雄太氏との共同研究) XPath の相対的な位置関係による当該データの検索は、XML が表す木構造上での計算を行うプログラムの表現であると考えられ、この等価性を論じることはプログラム変換の正しさを論じることに他ならない。XPath の式をその入力と出力の間の2項関係として表現し、異なるふたつの XPath 式の等価性を関係間の相互の包含関係を示すことによって示した。(ここで関係の包含関係はプログラムの詳細化と対応する。) また、XPath でよく用いられる、条件によるデータの絞り込みを、統一的な方法で 2 項関係によって表現することができることも示した。これによって、XPath の等価性および詳細化を、関係に関する少数の代数規則を運用することにより、簡潔に証明することが可能となった。実際に証明した等価性にはすでに知られていたものだけでなく、否定命題によるデータの絞り込みを伴うものなども含まれる。

この結果を、プログラム変換に関する国際研究集会 ACM SIGPLAN 2011 Workshop on Partial Evaluation and Program Manipulation (PEPM'11)において発表した。

② 並行計算プログラムのプログラム変換は、並行計算に本質的に内在する複雑さのため、逐次プログラムのそれよりもずっと困難な課題である。困難の原因は並行計算プログラムの適切な形式的モデルの欠如にあると考え、プログラミング言語の意味モデルを具体的かつ正確に与えることができるとして近年研究が進んできているゲーム意味論に基づく形式化を試みた。この目的のため、プログラミング言語 Algol に基づいた並行言語を設計し、そのゲーム意味論を与えた。ゲーム意味論では、プログラムの実行システムとプログラムの 2 者で交互に手を打つ一種のゲーム上の対局として定式化するが、並行プログラムをゲーム意味論で定式化する際、素直に複数のプログラムの実行を混ぜ合わせる(interleave)と上述の 2 者が交互に手を打つという条件が崩れてしまうことが問題となる。この問題を、プログラムが一時的に他の並行実行されているプログラムに実行権を譲り、実行権が回復され次第再開する、という動作を表す打ち手を導入することにより解決した。このような定式化が並行実行に関する完全抽象モデルを与えること、すなわち 2 つの並行プログラムの等価性が、それぞれのプログラムに関する可能な打ち手の履歴 (戦略) の等価性に帰着できることを示した。(渡辺敬介氏との共同研究) また、ある一定の条件を満たすプログラムについては対応する戦略が正規言語で表され、これによって並行プログラムの等価性が判定できることを示した。この結果は、並行プログラムの安全なプログラム変換の正当性の形式的検証に向けての重要な方向づけになると考えられる。

この結果を、情報処理学会プログラム研究会(PRO)において発表した。

５．主な発表論文等

(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, pp. 113-127, LNCS 6037, 2010.
http://dx.doi.org/10.1007/978-3-642-12592-8_9

② Shin-ya Katsumata, Susumu Nishimura, "Algebraic Fusion of Functions with an Accumulating Parameter and Its Improvement", Journal of Functional Programming, vol. 18(5-6), pp. 781-819, 2008.
http://dx.doi.org/10.1017/S095679680800693X

〔学会発表〕（計６件）

① Keisuke Watanabe, Susumu Nishimura, “May&Must-Equivalence of Shared Variable Parallel Programs in Game Semantics”, 情報処理学会プログラミング研究会 (PRO2011-5)

② Yuta Ikdeda, Susumu Nishimura, “Calculating Tree Navigation with Symmetric Relational Zipper”, 第13回プログラミングおよびプログラミング言語ワークショップPPL2011

③ Yuta Ikdeda, Susumu Nishimura, “Calculating Tree Navigation with Symmetric Relational Zipper”, The 20th ACM SIGPLAN 2011 Workshop on Partial Evaluation and Program Manipulation (PEPM'11), 2011
http://dx.doi.org/10.1145/1929501.1929521

④ Susumu Nishimura, “Refining Exceptions in Four-Valued Logic”, 代数，論理，幾何と情報科学研究集会 (ALGI20), 2009

⑤ Susumu Nishimura, “Refining Exceptions in Four-Valued Logic”, 19th International Symposium, (LOPSTR 2009)

⑥ Susumu Nishimura, “Safe Modification of Pointer Programs in Refinement Calculus”, 9th International Conference on Mathematics of Program Construction, MPC 2008, pp.284-304, LNCS 5133, 2009.
http://dx.doi.org/10.1007/978-3-540-70594-9_16

〔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〕
ホームページ等
① MPC 2008 発表論文に関する、形式的証明のダウンロード
http://www.math.kyoto-u.ac.jp/~susumu/mpc08pvs/index.html

６．研究組織
(1)研究代表者
西村 進 (NISHIMURA SUSUMU)

研究者番号：10283681