

科学研究費助成事業 研究成果報告書

平成 27 年 5 月 31 日現在

機関番号：12102

研究種目：挑戦的萌芽研究

研究期間：2013～2014

課題番号：25540023

研究課題名(和文) 必要呼び計算体系とコントロール抽象

研究課題名(英文) Calculi for Call-by-Need and Control Abstraction

研究代表者

亀山 幸義 (KAMEYAMA, Yukiyoshi)

筑波大学・システム情報系・教授

研究者番号：10195000

交付決定額(研究期間全体)：(直接経費) 1,200,000円

研究成果の概要(和文)：本研究は、コントロール抽象を提供するコントロールオペレータがどのように定式化され、また、それらを用いてどのような制御構造が表現できるかについて研究を行った。成果として、限定継続コントロールオペレータにおける「結果型変更」機能を使った全てのプログラムが、その機能なしのプログラムに変換できること、つまり、主流の関数型言語OCamlやHaskellにおいて実行可能となることを示した。これにより、コントロールオペレータの応用範囲を劇的に広げること成功した。この変換が型付けを保存することの厳密な証明を与え、さらに変換の実装を与え、本研究の成果を一般プログラマが利用できるようにした。

研究成果の概要(英文)：This research focused on control abstraction in programming languages. In particular, we have studied how control operators are formulated in terms of foundational calculi, and how they can represent interesting control structures in a sophisticated way. We have revealed that the mechanism of Answer-Type Modification for delimited-control operators is essential for a certain interesting programming idioms, and a program with this mechanism can be translated to a program without it. Our translation is novel and sufficiently general, in that any programs with this mechanism can be translated in main-stream functional programming languages such as OCaml and Haskell, thus widening the application area of control operators drastically. We have given a rigid proof for the type preservation property of this translation as well as a concrete implementation in the tagless-final style.

研究分野：プログラム論理

キーワード：コントロール抽象 関数型プログラム言語 計算モデル 型システム プログラム変換

1. 研究開始当初の背景

プログラミング言語における制御構造の抽象化、すなわち、コントロール抽象は、複雑で理解しにくいプログラムを、簡潔な構造を組み合わせて記述することにより、プログラムの理解・解析・検証を行いやすくすることを目的としている。Dijkstra の構造化プログラミングの提唱以来、様々な手段が提案されてきている。コントロール抽象を提供する有力な手段であるコントロールオペレータの研究は、主として関数型プログラム言語において研究され、ML 系言語の例外機構、Scheme の call/cc のほか、1990 年代から、限定継続コントロールオペレータの研究が盛んになっている。限定継続コントロールオペレータは従来からある call/cc を精密化したものであり、その振舞いが理論的に美しく説明されるため、質の高い基礎理論研究がなされるとともに、応用プログラムも多数提案されてきた。しかしながら、多くの応用は Answer-Type Modification (結果型の変更) という独特の機能を用いているため、OCaml や Haskell といった主流の関数型プログラム言語では使えず、研究論文における成果そのままのプログラムが、現実のプログラム言語では表現できない、という問題が生じていた。

また、コントロール抽象は制御構造に関するものである事から、必然的に、評価戦略と密接な関連があり、値呼び戦略以外に、名前呼び戦略や必要呼び戦略など、非標準の評価戦略におけるコントロール抽象のありかたについても関心が高まっていた。

2. 研究の目的

本研究は、このような問題意識のもと、コントロール抽象と必要呼びなどの評価戦略に関する様々な理論的成果を、現実に見えるようにする方法、言い換えれば、多くのユーザが日常的に利用するプログラム言語で、使える形にするためにはどうしたらよいかという問いに答えることを目的とした。

具体的には、限定継続コントロールオペレータの体系において計算戦略を定式化し、その体系の理論的性質を調べるとともに、この体系の応用範囲を調べることを第一の目的とした。必要呼び戦略は、Haskell 言語では既に採用されているが、その基礎理論研究は比較的新しく 2000 年以降に盛んになったものである。この体系における限

定継続コントロールオペレータの定式化は、我々の知る限り、他にはなく、このような体系の基礎理論および応用について先駆的かつ萌芽的な研究を行った。

第二の目的は、このように理論的に厳格に定式化した限定継続コントロールオペレータの体系について、それを主流のプログラミング言語で利用可能とするための方法を解明することである。こちらの目的は研究開始当初はあまり意識していなかったが、研究の進捗につれ、このための障害となる部分について同定し、その障害を取り除くことが極めて重要であると認識されたため、本研究の主要な目的の1つとした。

3. 研究の方法

第一の目的を達成するため、必要呼び体系についての基礎理論として有力な体系の1つである Ariola らの体系を取り上げ、その体系に shift/reset と呼ばれる限定継続コントロールオペレータを追加する研究を行った。体系の設計においては、極めて多くの設計上の岐路があることがわかり、理論的に良い性質を持つこと、既存体系の自然な拡張であること、及び、応用プログラムの表現力・表現のしやすさの観点でふるいかけ、体系として適切なものを特定した。この体系に対して、定式化、理論的性質の解明、抽象機械の導出、実装という一連の作業を行い、「shift/reset を持つ限定継続コントロールオペレータの体系」として世界で初めて必要呼び戦略と両立するものを作成した。

第二の目的を達成するため、Answer-Type Modification の機能の応用例と、定式化について様々な文献およびプログラム実例集を収集し検討をおこなった。その結果、「型付き printf」とよばれるプログラム例を OCaml で実現する手法があることが判明し、これを一般化して、任意のプログラムを変換することが可能ではないか、という着想に至った。この着想に基づき、体系の定式化、変換の定式化、その性質の証明、実装という一連の作業をおこない、研究成果を得ることに成功した。

4. 研究成果

本研究の研究成果は、以下のとおりである。

- (1) 必要呼び体系と限定継続コントロールオペレータを持つ体系の定式化：

「研究の方法」で述べたとおり、Ariola らの体系に基づいて限定継続を扱うことのできる体系の定式化を行い、望ましい理論的性質を持つ体系の作成に成功した。本研究では、インタープリタから抽象機械を導出する際に、Danvy らの「関数的対応」のアイデアに基づき、正しさが保証される形で導出することに成功しており、これは世界的にも、本研究が初めてのものであると考えている。この体系の応用先として、Curry などの関数論理型プログラム言語における探索のカプセル化機能の定式化をおこなうことを計画している。これについては関数論理型言語における意味論自体が、まだ十分に整理されていないことがわかり、その部分から研究を行う必要があったために、本研究での体系を用いた定式化には至っていないが、今後、この研究を進展させて、理論面のみならず、実用的にも意義がある成果としたい。(この成果についての論文発表については準備中である。)

(2) 限定継続コントロールオペレータと future を持つ並列計算戦略：

当初目的にはなかったものであるが、限定継続コントロールオペレータで記述したプログラムを効率的に実行するため、並列計算を行いたい、という考えから、このような研究を行った。具体的には、Scheme 言語における並列化プリミティブである future オペレータを導入し、スレッド単位での並列計算を導入した。ここでの研究課題は、単に2つのオペレータを併せ持つ体系を設計することではなく、その体系が、「コントロール抽象」の名にふさわしい体系であることを示すことである。すなわち、future を導入して並列化したプログラムの意味が、並列化前のプログラムの意味と同一であるという性質(透過性)が極めて重要であると考えた。このため、体系の厳密な定義を与え、透過性を厳密に証明することに尽力した。この証明は、当初想定していたよりはるかに困難なものであったが、最終的に、「模倣」(simulation)の関係を用いた厳密な証明に成功した。さらに、この体系による並列化の効果を測定するため、プログラム言語として設計し直し、この言語の処理系を実装した。実装にあたっては Scheme のプログラムを、並列化機能が備わっている Erlang のプログラムに変換する手法を用いて行い、実際にいくつかのプログラムに対して、本実装による並列化の

効果を確認した。この成果は、情報処理学会論文誌において発表した。

(3) 結果型変更機能のシミュレーション：

本研究の第2の目的である「コントロール抽象を、主流のプログラム言語で使えるようにする」ことについて、目覚ましい成果を得ることができた。すなわち、限定継続コントロールオペレータの数々の技巧的なプログラミング・イディオムで用いられている Answer-Type Modification (結果型の変更) 機能について、それを含む任意のプログラムを、それなしのプログラムに全自動で変換することに成功した。この変換は、Kiselyov による printf プログラムに対する変換を一般のプログラムに拡張したものに相当し、変換後の言語としては、結果型変更機能のない限定継続コントロールオペレータのみを利用している。本研究では、この変換アルゴリズムを一般的に定式化し、これがプログラムの型付けを保存することを厳密に証明した。(この結果は、国際会議 WoC で発表した。)

(4) 結果型変更機能を実現するシステムの作成：

上記(3)の理論的成果を、一般のプログラマが自由に使えるようにするため、変換アルゴリズムを実装した。実装にあたっては、埋め込み領域特化言語の実現に関する最新の成果である「タグなし最終スタイル」(tagless-final style)を用いることにより、型付けが適切におこなわれることを自動的に保証したものである。この実装は、結果型変更機能を用いた任意のプログラムが与えられると、この機能を使わないプログラムとして解釈して実行するものであり、理論的成果を理解する必要なく、一般のプログラマが手軽に利用可能なシステムとなっている点に価値があると考えている。

(5) 変換の効率化：

実際にシステムを作成し、具体的な例題に適用する作業を通じて、上記(3)の変換アルゴリズムが、必ずしも効率的でない部分があることが発見されたため、より効率的な変換について研究を行った。これは、結果型変更の機能を用いない部分まで、一律に一般的な変換をほどこしていたため、実際には必要のないデータの受け渡しをしてしまっていたためである。このため、効率的な変換のアルゴリズムを設計し、それに

についても、型付けの保存の性質に関する理論的保証と、システム実装を行った。このシステム実装は、従来の型システムの範囲を超えており、GADT (Generalized Algebraic Data Type)を利用する必要があることが判明した。GAD は最新の OCaml や Haskell などの言語で利用可能であるため、これを用いた効率化を行っても、本システムの有用性を減ずることはないと判断した。(この成果は、上記(4)の成果とあわせて、WoC 論文の改定版として論文投稿するよう、現在準備中である。)

5. 主な発表論文等

[雑誌論文] (計2件)

[1] Ikuo Kobori, Yukiyoshi Kameyama, Oleg Kiselyov, ATM without Tears: Prompt-Passing Style Translation for Typed Delimited-Control Operators, Electronic Proceedings of 2015 Workshop on Continuations (WoC'15), 12 pages, 2015.
(<http://ebooks.au.dk/index.php/aul/catalog/book/4>) 査読有

[2] 田中麻峰、亀山幸義: 限定継続機構と future を持つ計算体系の透過的意味論、情報処理学会論文誌、Vol. 54, No. 8, pp. 1996-2011, 2013. 査読有

[学会発表] (計1件)

[1] 小堀育男、亀山幸義、オレックキセリョーフ、shift/reset による結果型変更の実現(ポスター発表)、第17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、2015.3.5, 道後プリンスホテル、愛媛。

[その他] ホームページ等
<http://logic.cs.tsukuba.ac.jp/~kam/programming/>

6. 研究組織

(1) 研究代表者

亀山幸義 (KAMEYAMA, Yukiyoshi)
筑波大学 システム情報系・教授
研究者番号: 10195000

(2) 研究協力者

田中麻峰 (TANAKA, Asami)

筑波大学システム情報工学研究科・博士
後期課程(当時)

小堀育男 (KOBORI, Ikuo)

筑波大学システム情報工学研究科・博士
前期課程(当時)