

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25年5月24日現在

機関番号：12102

研究種目：基盤研究（B）

研究期間：2009～2012

課題番号：21300005

研究課題名（和文）コード生成のためのプログラミング言語の基礎理論

研究課題名（英文）Foundation of Programming Languages for Code Generation

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

研究成果の概要（和文）：本研究課題は、コード生成機能を持つプログラミング言語に、「計算エフェクト」を加えて、効率が高いコード生成プログラムを自然に記述することができ、更に、生成されたコードが高い安全性を持つことが保証されるような言語体系を構築することを目的としたものである。本研究で、限定継続という先進的なコントロールオペレータをもつコード生成プログラミング言語体系の設計およびその安全性を証明、また、処理系および型推論器の実装を完了した。さらに、コード生成計算体系の論理的基礎付け、型主導部分計算器の構築、また、コード生成言語の高性能計算への応用例の作成などを行い、本研究の有用性を実証した。

研究成果の概要（英文）：The objective of this research project is to introduce computational effect to the programming language for code generation (or “staging”) so that one can write efficient code generators naturally, and at the same time the generated codes can be guaranteed to be safe. As research results of this project, we have obtained a language for code generation with multi-prompt delimited-control operators, proved type soundness which implies the safety of the language, and implemented a proto-type interpreter and a type inference system for our language. We have also given a logical foundation for the calculus of code generation, built a type-directed partial evaluator for delimited-control operators, and implemented several useful algorithms in high-performance computation, which proved usefulness and applicability of our language.

交付決定額

(金額単位：円)

| | 直接経費 | 間接経費 | 合計 |
|--------|-----------|-----------|------------|
| 2009年度 | 3,000,000 | 900,000 | 3,900,000 |
| 2010年度 | 1,700,000 | 510,000 | 2,210,000 |
| 2011年度 | 2,100,000 | 630,000 | 2,730,000 |
| 2012年度 | 1,500,000 | 450,000 | 1,950,000 |
| 年度 | | | |
| 総計 | 8,300,000 | 2,490,000 | 10,790,000 |

研究分野：プログラム論理

科研費の分科・細目：情報学・ソフトウェア

キーワード：プログラミング言語論、プログラミングパラダイム

1. 研究開始当初の背景

「プログラムによってプログラムを生成する」という考え方は、コンパイラやパーサ生成器など古くから様々な分野で利用されている。また、部分計算(プログラム特化)や動的コード生成という名前で、汎用性や保守性の高さと実効性能の高さを両立させる手法として注目されている。近年では、領域特化言語 (Domain Specific Language) の処理系の高速化への応用など、応用範囲を広げている。

これに対するプログラミング言語からのサポート機能として、古くは Lisp の擬似引用機構 (quasi-quotation) があり、型付き言語の世界ではそれを ML に取り込んで拡張した MetaML や MetaOCaml が提案されてきた。後者では、型システムを用いた安全性について研究がなされ、コード生成器の型の整合性を検査すれば、生成されるコードの安全性が保証される、という画期的な成果が得られていた。

しかし、本研究開始時点では、これらの言語や体系で生成されるコードの安全性が保証される範囲は、純粋な (計算エフェクトを持たない) 関数型言語に対するものに限定されていた。コード生成器の効率をあげるための種々の技法は、この範囲では自然に記述できないことが多く、また、実用的なプログラムは計算エフェクトを必要とすることから、これらの言語体系では不十分なことが明らかであった。現実世界では、コード生成機能を使ったプログラムは多く作成されていたが、それらは安全性の保証の範囲外のため、コードを生成するごとに安全性を確かめる必要があるという本質的な問題があった。

2. 研究の目的

本研究の目標は、コード生成機能を持つプログラミング言語に、「計算エフェクト」を加えて、効率が高いコード生成プログラムを自然に記述することができ、更に、生成されたコードが高い安全性を持つことが保証されるような言語体系を構築することである。

先行研究で、(ある範囲での) すべての計算エフェクトが、「限定継続コントロールオペレータ」で表現できることがわかったので、これをコード生成のための言語に追加し、なおかつ、安全性が保証されるような範囲を探ることを本研究の目的とした。その範囲が適切であることを示すため、対応するプログラミング言語の設計、安全性の保証、処理系の実装、そして応用例の構築をすることまでを含めて、本研究の具体的な目的とした。

また、コード生成機能に対する基礎と応用の両面からの研究を幅広く、かつ、深く推進して更なる発展を目指すため、基礎の方では

型理論や論理学の立場からの基礎付けを行い、応用の方では、型主導部分計算など深く関連する手法の探求等を行うことも目的とした。

3. 研究の方法

(1) 限定継続コントロールオペレータは、計算エフェクトの表現において万能性を有するが、1つの計算エフェクトに対して1つのコントロールオペレータを必要とする。そこで、任意の個数の異なる計算エフェクトを表現するため、コントロールオペレータに名前をつけて区別することができる「マルチプロンプト対応のコントロールオペレータ」を研究対象とすることにした。

コード生成のための基礎理論として最も広く使われていたものは、Walid Taha らによる MetaOCaml の基礎をなす計算体系 ($\lambda \alpha$ および λi) であったため、それを若干簡略化したものをベースとし、それに上記のマルチプロンプト対応のコントロールオペレータを追加して、安全性を保証できる範囲を探求した。

(2) 論理的基礎付けの方では、上記の基礎的計算体系 $\lambda \alpha$ を分析し、そこに現れる型付け規則を Curry-Howard の同型対応の観点から1つ1つ検査し、適当な対応関係を持たない規則について検討することにより、論理的見地から適切な計算体系の設計とその意味の考察を行うこととした。

(3) コード生成器において限定継続コントロールオペレータを使うことが本質的である、という視点に立ち、限定継続コントロールオペレータに対する型主導部分計算器の構築を試みた。部分計算においては、その正しさの保証が重要課題であるため、その保証を行う手法を検討した。

4. 研究成果

(1) 現在、最も先進的なコントロールオペレータである「マルチプロンプト対応の限定継続コントロールオペレータ」と、コード生成機能、および、生成されたコードの実行機能を持つ計算体系を設計した。この体系は、ある範囲 (モナディックに表現できる) あらゆる計算エフェクトを表現できるとともに、コード生成においても、たとえば「let 挿入」という効率的コード生成で必須の機能を記述するために十分な表現力を持つものである。また、上記の機能を制限なく組み合わせで構成したプログラムの中には、安全でないもの (生成されたコードが、コンパイルエラーとなるもの) が存在するため、そのような安全でないプログラムを排除するよう、型付

け規則において、計算エフェクトに関する制限を設定した。

(2) この体系が型安全性を満たすことを厳密に証明した。これにより、この体系のプログラムは、実行時に決して型エラーを起こさないこと、また、生成されたコードは構文が適切で、型エラーを起こさず、さらに、自由変数が出現しないことが保証され、コード生成の安全性が保証された。このような体系の提案は世界で初めてである。

(3) コード生成器のためのライブラリ関数の多くが多相型を持つため、本研究の体系を多相型に対応したものに拡張する研究を行った。ML 系言語では、計算エフェクトと多相型の組み合わせだけで安全性に問題が生じることが良く知られており、その解決策として値多相(value restriction)がある。本研究では、コード生成言語では、値多相によって安全な言語を設計することが不可能であることを示し、その解決策として本研究代表と分担者が 2007 年の共著論文で提案した手法を援用することにより、多相型と計算エフェクトを持つコード生成言語の安全性を保証することに成功した。

(4) この体系に対応するプログラミング言語が有用なものであることを示すため、処理系(インタープリタ)と型推論器を設計し、実装した。本実装は、proof of concept のためのものであるが、コード生成の例題を記述するには十分なものである。作成した型推論器は、ML 系言語の型推論器の拡張である。この処理系を利用して、MetaOCaml におけるプログラム例題の記述と同様な例題のほか、let 挿入などを行った効率的コード生成例題を記述し、確認した。

(5) コード生成計算体系の論理的基礎付けを世界で初めて行った。MetaOCaml の基礎理論として使われてきた体系は、コード実行機能を安全に行うための仕組みに特徴があるが、その部分は Curry-Howard 同型対応のもとでの論理学との対応がつかなかった。一方、本研究の体系は、従来体系における概念を整理し、コード実行機能に対する型付けをコード利用のための型付けと統合することにより、型に対する新しい解釈が可能となり、論理体系と対応する体系とすることに成功した。

(6) 限定継続コントロールオペレータを持つ計算体系に対する型主導部分計算器を構築した。さらに、限定継続コントロールオペレータが CPS 変換で意味を与えられることに着目し、CPS 変換を取り込んだ形での

部分計算器の正しさの保証を試み、証明することに成功した。さらに、この成果を形式検証のためのシステムである Coq において形式化して証明を与えることに成功した。

(7) 上記の成果をもとに、さらに新しい発展の方向性を展望するため、2012 年 5 月に NII 湘南ミーティング “Bridging the theory of staged programming languages and the practice of high-performance computing” を主催した。この集会には、本研究課題のメンバー 3 名を含め、コード生成(ステージ化計算)と高性能計算のそれぞれの第一線専門家が約 30 名参加し、最先端の研究成果の発表と新たな発展のための研究討論をおこなった。特に、現在のところ理論面で大きな発展を遂げてきたコード生成言語の成果が、他の研究分野や実世界にどう影響を及ぼし得るかについて、討論を行った。

この集会の成果の 1 つとして、高性能計算で実際に使われているコードを、コード生成言語を用いて生成することを目的とした国際チャレンジである “Shonan Challenge” がある。これは、高性能計算のためのコードの投稿と、そのコードを生成するプログラムの投稿から構成され、後者としてコード生成言語における優れた解であるかどうかを評価対象となる。通常のコテキストと違い、性能ではなくプログラムの質が評価対象であるという点で新しいタイプのチャレンジである。Shonan Challenge のウェブサイトは全世界に開放されており、現在でも新しい問題(生成したいコード)とコード生成言語による解の投稿がされている。この 2 つの分野の相互作用による発展を加速させることに成功している。

(8) 本研究課題の 4 年間に、いくつかの最先端テーマに関する研究集会の主催、共催等を行い、本研究に関連するプログラミング言語研究の発展に貢献した。

2011 年 9 月には ACM Continuation Workshop (「継続」に関するワークショップ)を国立情報学研究所において ICFP 国際会議に併設して開催した。これはアブストラクト投稿による査読プロセスを持つ公式の会議として開催し、10 件程度の発表があった。また、Workshop 前日には、本研究分担者らによるチュートリアルを開催した。

また、2013 年 2 月には、Workshop on Staged Computation (ステージ化計算に関するワークショップ)を筑波大学において 2 日間にわたって開催した。

いずれも、国内外の第一線の研究発表を集めて、活発な討論が行われた。

5. 主な発表論文等

[雑誌論文] (計 12 件)

- ① Baris Aktemur, Yukiyoshi Kameyama, Oleg Kiselyov and Chung-chieh Shan, "Shonan Challenge for Generative Programming", Proc. ACM SIGPLAN 2013 Workshop on Partial Evaluation and Program Manipulation (PEPM 2013), pp. 147-154, Jan., 2013. (査読有)
- ② Asami Tanaka and Yukiyoshi Kameyama, "A Call-by-Name CPS Hierarchy", Proc. International Symposium on Functional and Logic Programming (FLOPS 2012), Kobe, Japan, Lecture Notes in Computer Science 7294, pp. 260-274, May, 2012. (査読有)
- ③ Yuichiro Kokaji, Yukiyoshi Kameyama, "Polymorphic Multi-Stage Language with Control Effects", Proc. Ninth Asian Symposium on Programming Languages and Systems (APLAS 2011), Kenting, Taiwan, Lecture Notes in Computer Science 7078, pp. 105-120, Dec. 2011. (査読有)
- ④ Yukiyoshi Kameyama, Oleg Kiselyov, Chung-chieh Shan, "Shifting the Stage - Staging with Delimited Control", Journal of Functional Programming, 21(6), pp. 617-662, Nov., 2011. (査読有)
- ⑤ Koji Nakazawa, Makoto Tatsuta, Yukiyoshi Kameyama, Hiroshi Nakano, "Undecidability of Type-checking in Domain-free Typed Lambda-Calculi with Existence", Theoretical Computer Science 412, pp. 6193-6207, Oct., 2011. (査読有)
- ⑥ Kensuke Kojima and Atsushi Igarashi, "Constructive linear-time temporal logic: Proof systems and Kripke semantics", Information and Computation, 209(12), pp. 1491-1503, 2011. (査読有)
- ⑦ 杉浦啓介, 亀山幸義, 「コード実行機能と計算エフェクトを持つ型付きマルチステージ言語」、コンピュータソフトウェア, Vol. 28, No. 1, pp. 217-229, 2011. (査読有)
- ⑧ Moe Masuko, Kenichi Asai, "Caml Light + shift/reset = Caml Shift," Theory and Practice of Delimited Continuations (TPDC 2011), pp. 33-46, May, 2011. (査読有)
- ⑨ Kanako Sakurai, Kenichi Asai,
⑩ Takeshi Tsukada, Atsushi Igarashi, "A logical foundation for environment classifiers", Logical Methods in Computer Science, 6(4:8), pp. 1-43, Dec. 2010. (査読有)
- ⑪ Cynthia Kustanto, Yukiyoshi Kameyama, "Improving Error Messages in Type System", IPSJ Transactions on Programming, Vol. 3, No. 4, pp. 43-56, Sep. 2010. (査読有)
- ⑫ Yukiyoshi Kameyama and Asami Tanaka, "Equational Axiomatization of Call-by-Name Delimited Control", Proc. 12th ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP 2010), pp. 77-86, Jul. 2010. (査読有)

[学会発表] (計 12 件)

- ① 廣田知子, 浅井健一, 「shift/reset 付き TDPE の抽出」, 第 15 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論文集, 14 ページ, 御宿東鳳 (会津若松市), 2013 年 3 月 6 日
- ② 金子ちひろ, 浅井健一, 「shift/reset によるモナドトランスフォーマの提案と実装」, 第 15 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論文集, 14 ページ, 御宿東鳳 (会津若松市), 2013 年 3 月 4 日.
- ③ 宮部浩太郎, 亀山幸義, 「結果型を変更可能な限定継続の模倣」, 情報処理学会第 93 回プログラミング研究会 (PRO-2012-5), 国立情報学研究所 (東京都), 2013 年 2 月 28 日.
- ④ 岩井亜里紗, 浅井健一, 「メタラムダ計算の定式化」, 第 14 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論文集, ポスター発表, p. 47, 南紀白浜 むさし (和歌山県), 2012 年 3 月 8 日.
- ⑤ 小山内幸一, 五十嵐淳, 「低水準コード生成を行う λ O 仮想機械の融合変換を使った系統的導出」, 日本ソフトウェア科学会第 28 回大会論文集, 沖縄県市町村自治会館 (沖縄県那覇市), 2011 年 9 月 27 日.
- ⑥ Naoki Takashima, Tatsuya Nishiyama, Yukiyoshi Kameyama, "Visualizing

- Continuations (short talk)", ACM SIGPLAN Continuation Workshop, 一橋記念講堂 (東京都), 2011 年 9 月 24 日.
- ⑦ 高島尚希, 亀山幸義, 『コントロールオペレータの表現力の比較』, 情報処理学会第 84 回プログラミング研究会 (PRO84), サン・リフレ函館 (北海道函館市), 2011 年 6 月 14 日.
 - ⑧ 小鍛治雄一郎, 亀山幸義, 「エフェクトを持つマルチステージ計算体系の型推論」, 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論文集, pp. 258-272, 定山溪ビューホテル (北海道札幌市), 2011 年 3 月 11 日.
 - ⑨ Cynthia Kustanto, Yuki Yoshi Kameyama, "Improving Error Message in Type System", 情報処理学会プログラミング研究会, 電気通信大学, 2010 年 3 月 16 日.
 - ⑩ 田中麻峰, 亀山幸義, 「限定継続に基づくスケーラブルなウェブアプリケーション構築手法」, 電子情報通信学会ソフトウェアサイエンス研究会, 鹿児島大学, 2010 年 3 月 8 日.
 - ⑪ 対馬かなえ, 浅井 健一, 「限定継続のための TDPE に向けて」, 第 12 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論文集, pp. 64-76, 琴参閣 (香川県琴平市), 2010 年 3 月 3 日.
 - ⑫ 杉浦啓介, 亀山幸義, 「コード実行機能と計算エフェクトを持つ型付きマルチステージ言語」, 日本ソフトウェア科学会第 26 回大会, 島根大学, 2009 年 9 月 18 日.

[その他]

ホームページ等

Shonan Challenge:

<https://github.com/StagedHPC/shonan-challenge/>

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) 研究分担者

浅井 健一 (ASAI KENICHI)
お茶の水女子大学・人間文化創成科学研究科・准教授
研究者番号: 10262156
五十嵐 淳 (IGARASHI ATSUSHI)
京都大学・情報学研究科・教授
研究者番号: 40323456