

科学研究費助成事業 研究成果報告書

平成 27 年 5 月 26 日現在

機関番号：12611

研究種目：基盤研究(C)

研究期間：2010～2014

課題番号：22500025

研究課題名(和文)自己反映言語の効率的実装とそれを支える基盤技術

研究課題名(英文)Foundations and efficient implementation of reflective languages

研究代表者

浅井 健一 (ASAI, Kenichi)

お茶の水女子大学・大学院人間文化創成科学研究科・准教授

研究者番号：10262156

交付決定額(研究期間全体)：(直接経費) 3,300,000円

研究成果の概要(和文)：従来の自己反映言語において生じていた不具合の原因を特定した。その回避法として、言語の意味は直接形式に保ちつつ実装のみを非関数化した継続渡し形式にするという方法を提示した。これに基づいて Meta0CamI で自己反映言語を実装し、ユーザによって変更されたメタレベルインタプリタのコンパイルを実現した。一方、変更されたメタレベルインタプリタのもとでのコンパイルは現時点では難しいことを指摘し、その実現には Meta0CamI に束縛時解析が必要なことを示した。

研究成果の概要(英文)：The cause of the level-shifting anomaly is identified. The anomaly can be avoided by implementing the reflective language in the defunctionalized continuation-passing style while keeping the user-observable interpreter in direct style. Using this method, the reflective language is implemented in Meta0CamI, which achieves compilation of the user-defined metalevel interpreter. On the other hand, compilation under the modified metalevel interpreter is difficult, due to the lack of the binding-time analysis in Meta0CamI.

研究分野：プログラミング言語

キーワード：自己反映言語 部分評価 プログラム変換 Meta0CamI

1. 研究開始当初の背景

自己反映言語とは、プログラム実行中にそのプログラムの動作の一部、変更できるような言語のことである。自己反映言語のすぐれている点は、実用的にはプログラムをその時々状況に応じて最適化できる点、また理論的には様々な言語処理系の概念を内包しているためプログラミング言語に対する深い洞察を与えてくれる点である。

しかし、自己反映言語の効率的な実装は、自己反映言語の高い柔軟性ゆえに難しい。一般に自己反映言語で書かれたプログラムをコンパイルするには、言語の意味を与えるインタプリタをユーザプログラムについて特化するしか方法がない。その実現のためには、強力な部分評価器と自己反映言語の実装に不可欠な継続に関する知識が必要となる。

一方、限定継続の理論や限定継続を操作するプログラムの部分評価手法などが開発されてきたことから、この技術を組み合わせれば自己反映言語のプログラムのコンパイルが達成される可能性が出てきた。

2. 研究の目的

本研究の目的は、3つの基盤技術、すなわち2種類のコードの共存法の確立、プログラム変換技術の開発と発展、部分継続に関する理論の発展を通して、これまで難しいとされてきた自己反映言語の効率的な実装を行う。

2種類のコードの共存法の確立では、インタプリタで解釈実行されるコード(ベースレベルコード)とコンパイルされて直接実行されるコード(メタレベルコード)を明確に区別し、自己反映言語の構造を明らかにする。そして自己反映言語のコンパイルを前者を後者に変換するという形で定式化する。プログラム変換技術の開発と発展では、部分評価を中心として、自己反映言語をコンパイルできるくらい強力なプログラム変換技術を開発するとともに、実行時にコンパイルできるようその高速化も目指す。また、部分継続に関する理論の発展では、これまでの研究では扱って来なかった再帰と部分継続の関わりや、部分継続命令の型主導による高速な部分評価手法の開発を試みる。

3. 研究の方法

(1) すでに提案されている継続渡し形式の自己反映言語の分析を行う。この言語では継続が特別な役割を果たしているため、ユーザの自己反映機能も継続渡し形式で実現するものに限られる。全ての自己反映機能を許すためには直接形式の自己反映言語が必要だが、これまでの直接形式の自己反映言語はレベル遷移時に不具合が生じていた。そこで、まず継続渡し形式の実装を詳細に分析しその構造を明らかにする。その上で、解釈実行

されるコードと直接実行されるコードの関係を明らかにする。それを通して、自己反映言語の実行時コンパイルを前者を後者に部分評価を使って最適化する問題として定式化する。

(2) 自己反映言語では言語の意味(メタレベルインタプリタ)がプログラム実行途中で変化する。したがって、プログラムをコンパイルするには、メタレベルインタプリタをユーザプログラムで部分評価する必要がある。その第1段階として、まだ変更されていない標準的なインタプリタの部分評価を行う。次に第2段階として、ユーザによって変更されたインタプリタの部分評価(コンパイル)を行う。ユーザによる変更すべてに対応するのは原理的に難しいが、自己反映言語を使う上でよく行われる代表的な変更に対して部分評価によるコンパイルを試みる。

(3) 自己反映言語を効率的に実装するために必要な部分継続に関する理論の発展を行う。実行時に部分評価(コンパイル)を行うことを目指して、高速であるとされる型主導の部分評価手法を研究する。また、必要に応じて再帰と部分継続の関わりを検討する。さらに、部分継続を使った応用も検討する。それらを通して自己反映言語の効率的な実装に必要な技術を整備する。

4. 研究成果

(1) 既存の継続渡し形式の自己反映言語を詳細に分析した結果、これまでのシステムは「一旦、メタレベルで仕事をした後、ベースレベルに戻ってくる際、上手に継続をつなぐことで自己反映機能を実現している」ことがわかった。さらに、直接形式で自己反映言語を実装すると上手に継続をつなぐことができずに余計なフレームが入り込むことを突き止めた。その余計なフレームは、以下に示すふたつの技術を使うと取り除くことができ、それを使うとレベル遷移時に不具合を生じることなく直接形式の自己反映言語を実装できることがわかった。余計なフレームを取り除く際に必要なひとつ目の技術は、ユーザから見えるメタレベル言語は直接形式に保ったまま、実装を継続渡し形式に変更することである。これにより(ユーザから直接は見えていないものの)実装レベルで継続を扱うことができるようになる。もうひとつの技術は、その継続を非関数化して1階のデータにすることで、余計なフレームかどうかを判定できるようにすることである。これを通して、余計なフレームだったら取り除くという処理を書くことができるようになる。

直接形式の自己反映言語を実現するにあたり、実装を継続渡し形式にするのでは、一見、従来の継続渡し形式の自己反映言語と変わらないように見えるが、そうではない。実装

は継続渡し形式でもユーザからは直接形式のメタレベルインタプリタが見えているので、この言語は直接形式の自己反映言語を実現している。ユーザから見えているインタプリタと実際のインタプリタが同じものにならないのは普通のことで、従来の継続渡し形式の自己反映言語でも同じことが起きている。

以上の成果は国際会議にて発表した。この結果は、大きくふたつのことを意味している。まず、ここで作成した自己反映言語の実装が継続渡し形式になっているため、限定継続を正面から扱わなくてもコンパイルを行える見通しとなった。これは、コンパイルを行う際に必要な部分評価器が限定継続を扱えなくても構わないことを意味している。もうひとつは、ここでの結果により、部分評価器を使ったコンパイルなど効率的な自己反映言語の実装を行う土台が整った。この先は、ここで作成した自己反映言語を使ってコンパイルを行っていくことになる。

(2) 前項の結果、限定継続を扱わない部分評価器でも自己反映言語をコンパイルできる見通しとなった。一方で、従来の部分評価器では部分評価結果がプログラムテキストの形でしか得られず、実行時にその場の状況を使ってコンパイルを行うことができないという問題があった。そこで、その場の状況を使って部分評価と同じ効果を得られる多段階言語 MetaOCaml を使ってコンパイルを試みた。

MetaOCaml を使って部分評価を行うには手動でインタプリタに注釈を付けなくてはならない。標準的なインタプリタに対して注釈を与え、それを使って自己反映言語のコンパイルを行ったところ、かなりの速度向上を得られることがわかった。実際、ほとんどの解釈実行部分は最適化され、直接実行しているのとの差は原始演算のインライン展開が今のところできていないなど一部のみとなった。その結果、ユーザ定義されたメタレベルインタプリタのコンパイルがある程度、可能になった。これまで、自己反映機能を使ってメタレベルインタプリタを変更すると、それ以後、ユーザプログラムは2段分のインタプリタによって解釈実行されていたが、そのうちの1段分を最適化して、1段のみのインタプリタによって解釈実行されているのとほぼ変わらないようにすることができたことになる。この成果は国内のワークショップ、及び国際会議にて発表した。

一方で、このアプローチではあらかじめメタレベルインタプリタに注釈をつける必要があるため、第2段階のコンパイル、すなわちユーザによって変更されたインタプリタの部分評価(コンパイル)は難しいことがわか

った。これを行うためには、第1段階のコンパイルで得られたプログラムに対して注釈を付ける必要があるが、そのプログラムは実行時にしか得られず、あらかじめ注釈を付けることができないためである。この問題に対処するためには、与えられた任意のプログラムに対し、自動で注釈を付ける技術、つまり束縛時解析が必要となる。MetaOCaml に束縛時解析を実装するのは面白い研究課題であり、本研究期間終了後に取り組み予定である。

(3) 部分継続を扱わなくても自己反映言語のコンパイルを(ある程度)行えることがわかったので、部分継続の理論の発展のウェイトは下がったが、上記の研究と並行して各種の研究を行った。まず、限定継続を含む体系に対する型主導の部分評価手法について研究した。入力プログラムを2度、継続渡し形式に変換した上で従来の手法を適用することで型主導の部分評価器を得るとともに、その正当性を定理証明系を使って示した。しかし、今のところこの部分評価器は複雑で、その本質を理解するには至っていない。この部分評価器の整理と理解も今後の課題である。

また、部分継続の実装について、部分継続がスタックの一部に対応していることを理論的に示した。さらに、例外処理が入った場合にどうなるかについても検討した。また、応用として printf と限定継続の関わりを深化させるとともに、探索ライブラリを限定継続を使って実装すると(prolog のような感じで)簡潔に探索問題を記述できることを示した。

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 6 件)

K. Asai "Compiling a Reflective Language using MetaOCaml," Proceedings of the 2014 International Conference on Generative Programming: Concepts and Experiences (GPCE '14), pp. 113-122 (2014), 査読あり, DOI: 10.1145/2658761.2658775

浅井 健一「MetaOCaml を使った自己反映言語のコンパイル」、第16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 (2014)、査読あり、<http://pplab.is.ocha.ac.jp/~asai/jpapers/ppl/>

増子 萌、浅井 健一「例外と限定継続命令をサポートする評価器からの仮想機械とコンパイラの導出」、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、15 ページ (2013)、査読あり、<http://pplab.is.ocha.ac.jp/~asai/jpape>

rs/pp/

K. Asai, O. Kiselyov, and C.-c. Shan
“Functional un|unparsing,” Higher-
Order and Symbolic Computation, Vol. 24,
No. 4, pp. 311-340 (2011), 査読あり、
DOI 10.1007/s10990-012-9087-2

K. Asai “Reflection in Direct Style,”
Proceedings of the Tenth Conference on
Generative Programming and Component
Engineering (GPCE '11), pp. 97-106
(2011), 査読あり、DOI: 10.1145/2189751.
2047882

K. Asai and A. Kitani “Functional
Derivation of a Virtual Machine for
Delimited Continuations,” Proceedings
of the 2010 Symposium on Principles and
Practice of Declarative Programming
(PPDP '10), pp. 87-97 (2010), 査読あり、
DOI: 10.1145/1836089.1836101

[学会発表](計 2 件)

中野 祥、浅井 健一「データベースを用い
た Miki の拡張に向けて」、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、2013年3月5日、御
宿東鳳(福島県・会津若松市)

金子 ちひろ、浅井 健一「探索アルゴリズム
のための非決定性オペレータの実装」、
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、2011年3
月9日、定山溪ビューホテル(北海道・札幌市)

6. 研究組織

(1) 研究代表者

浅井 健一 (ASAI, Kenichi)
お茶の水女子大学・大学院人間文化創成科
学研究科・准教授
研究者番号: 10262156

(2) 研究分担者 なし

(3) 連携研究者 なし