

氏名	うがわともはる 鵜川 始陽
学位(専攻分野)	博士(情報学)
学位記番号	情博第165号
学位授与の日付	平成17年3月23日
学位授与の要件	学位規則第4条第1項該当
研究科・専攻	情報学研究科通信情報システム専攻
学位論文題目	一級継続のスタックベース処理系向け実装手法

(主査)
論文調査委員 教授 湯浅太一 教授 富田真治 教授 奥乃博

論文内容の要旨

本研究では、関数フレームをスタック上に生成するプログラミング言語処理系に、一級継続を効率良く実装するための手法を開発している。本論文はその成果をまとめたものであり、以下の7章から構成されている。

まず第1章では、研究の背景および動機として、一級継続が有用であることと、それを効率良く実装することが難しいことを述べている。また、第2章以降の構成を述べ、さらに、研究で得られた主要な成果を要約している。

第2章では、継続について説明している。継続の概念について説明し、継続を一級オブジェクトとして扱うことで可能となるプログラミングについて述べている。ここでは、例外処理とコールチンを例に挙げて一級継続を持つプログラミング言語の記述性の高さを示している。さらに、既存の一級継続の実装戦略についても述べている。

第3章では、他言語との相互呼出しが可能な処理系のための一級継続の実装手法を示している。継続の実装には、プログラム実行中に存在する全ての関数フレームを操作する必要がある。しかし、他言語と相互呼出しをするプログラムでは、関数フレームが一級継続を持つ言語のスタックと他言語のスタックの両方に分散して存在し、他言語のスタック上にある関数フレームは操作できないことがある。このような場合、無限の寿命を持つ完全な一級継続を実現することはできない。そこで、完全ではないが、このような処理系でも実現可能で、かつ多くの場面で有効に利用できる一級継続の実装手法を提案している。

第4章では、一級継続を効率良く実装する手法として、遅延スタックコピー法を提案している。関数フレームをスタック上に生成する処理系では、関数からリターンすると、関数フレームが解放されてしまう。そこで、従来の一級継続の実装では、継続がキャプチャされたときに関数フレームのコピーを生成し、将来の継続の呼出しに備えていた。しかし、関数フレームが解放されるより前にしか呼び出されない継続では、関数フレームのコピーは必要ない。遅延スタックコピー法では、関数フレームが解放される直前までコピー操作を遅延させる。もし、関数フレームをコピーする前に継続が必要なくなったことが分かれば、コピーはしない。これにより、関数フレームのコピーに要する時間や、コピー先の記憶領域を節約することができることを、ベンチマークプログラムを用いて確認している。

第5章では、遅延スタックコピー法を拡張した実装手法として、一つの関数フレームのコピーを複数の継続で共有するスタックコピー共有法を提案している。関数フレームのコピーを共有する一級継続の実装戦略は、既にいくつか提案されている。しかし、これらの実装戦略は特殊な処理系の構造を必要とするため、関数フレームをスタック上に生成する一般的な処理系に導入するには、処理系の構造を大きく変更しなければならなかった。スタックコピー共有法を使えば、一般的な処理系でも、局所的な変更で関数フレームのコピーを共有することが可能となる。

第6章では、遅延スタックコピー法のオーバーヘッドであるエスケープバリアを、静的解析により実行前に削減する手法を提案している。遅延スタックコピー法では、ヒープへの書き込みがある場合、毎回、書き込まれる値をチェックするエスケープバリアが働く。このため、一級継続を使わないプログラムにもオーバーヘッドがかかる。そこで、プログラムを実行前に

解析し、動作に影響しないバリアを除去する。これにより多くの不要なバリアが除去できることを確認している。

第7章はまとめであり、本研究で得られた主要な成果について要約している。

論文審査の結果の要旨

本論文は、関数フレームをスタック上に生成するプログラミング言語処理系に、一級継続を効率良く実装するための手法に関する研究成果を取りまとめたものであり、得られた主な研究成果は次の通りである。

- (1) Scheme 言語のような一級継続を持つ高級言語が、他言語で記述された関数と相互呼出しする機能を持つ場合、高級言語の処理系は高級言語のスタックは操作できても他言語のスタックは操作できない場合がある。このような場合、一般には無限の寿命を持つ完全な一級継続を実現することができない。本論文では、このような処理系でも実装可能な一級継続の実装手法を示した。これにより実現される継続は、完全ではないが、実用上役に立つ場合が少なくない。
- (2) 従来、スタックベースの処理系では、継続がキャプチャされる際にスタック上の関数フレームをヒープにコピーすることで一級継続を実現していた。本論文で提案する遅延スタックコピー法では、継続の実行に必要な関数フレームがスタック上に残っている間は、関数フレームのコピーを遅延させる。そのうえで、実際に関数フレームがコピーされる前に継続が不要になった場合にはコピーしないようにすることで、従来の処理系で行われていた無駄な処理を省くことを可能にした。
- (3) 継続が複数回にわたってキャプチャされる場合でも、原理的には同じ関数フレームは1回コピーすれば十分である。しかし、従来の継続の実装手法では、この原理を実装できない場合があった。本論文では、遅延スタックコピー法を実装することで、この原理を一般的な処理系に実装できることを明らかにし、スタックコピー共有法として実装モデルを示した。遅延スタックコピー法およびスタックコピー共有法を実装することにより、多くの一級継続を使うプログラムを高速に実行することが可能となった。
- (4) 遅延スタックコピー法をコンパイラに応用する際に問題となると予想されるオーバーヘッドの多くを、プログラム変換により除去する手法を示した。

以上要するに本論文は、一級継続を様々な処理系に効率良く実装する手法を提案するものであり、学術上、実際上寄与するところが少なくない。よって、本論文は博士（情報学）の学位論文として価値あるものと認める。また、平成17年2月24日実施した論文内容とそれに関連した諮問の結果合格と認めた。