3K-1

オブジェクト計算における変更可能な継続

久間恵美子 [†] 松本大介 [‡] 西崎真也 [†] 渡部卓雄 [†]

[†] 東京工業大学大学院情報理工学研究科計算工学専攻　[‡] 東京工業大学工学部情報工学科

1　はじめに

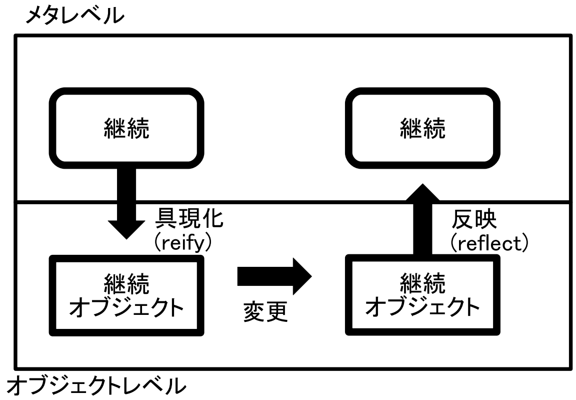
オブジェクト計算にファーストクラス継続を導入した継続オブジェクト計算 [1, 2] が提唱されている。

これは、元々Abadi と Cardelli により提唱されたオブジェクト計算 [3] を拡張した計算体系である。オブジェクト計算とは、簡潔な構文と意味論でセルフパラメータの遅延束縛などオブジェクト指向の特徴を形式化した計算体系である。もっとも基本的なものとしては型なしオブジェクト計算からはじまって，命令型オブジェクト計算や一階型体系や部分型をそなえたものなど，さまざまな体系が展開されている。

また、継続とは、式を評価している途中のある時点での後に続く計算を表現する概念で、ファーストクラス継続とは、メタレベルにおいて存在する継続をオブジェクトレベルのデータとして保存 (具現化，reify) し、また，具現化した継続を再びメタレベルに戻し利用することを可能とする計算機構である。継続は Felleisen ら [4] により評価文脈として形式化され，大域脱出の他，コルーチンやバックトラックなど高度な制御構造を実装することに利用される。

本研究では、オブジェクト計算、ファーストクラス継続という計算機構を加え，さらに，具現化された継続を変更する (modify) ことを可能とする計算体系 $\varsigma\kappa m$ 計算を提唱する。

継続オブジェクト計算の体系 [1, 2] や Scheme[5] のようなファーストクラス継続をもつプログラミング言語においては，継続オブジェクト内に保存されている評価文脈内のメソッドの変更は不可能である。そのため，継続を呼び出して評価文脈を再利用するとき評価文脈の一部分を変更して扱いたいような場合があったとしたら，継続オブジェクトを作り直さなければいけなかった。これを、直接変更できるように体系を再定義し直すことで、オブジェクト計算で表現できる幅を広げる。



本論文では、最初に、セルフパラメータをより形式的に扱うために、ラムダ計算で提唱されているデュブルーイン表記を継続オブジェクト計算に導入する。これにより、シグマ束縛だけでなく継続束縛も代入操作時の検査の対象となるため、継続オブジェクトがより形式的になる。その上で、型無し継続オブジェクト計算へ継続オブジェクトに継続の変更を可能とするような拡張を定義する。

2　型なし $\varsigma\kappa^m$ 計算

まず構文を紹介する。

$$
\begin{array}{rcl}
a, b & ::= & x \\
& | & [l_i = \varsigma b_i^{i \in 1..n}] \\
& | & a.l \\
& | & a.l \Leftarrow \varsigma b \\
& | & \kappa(l_{val}, l_{arg})a \\
& | & \mathcal{A}(a) \\
& | & \text{"|"}
\end{array}
$$

ここでは，セルフ変数の束縛子 ς はデブリューイン記法に従うものとし，変数は直近のものから順に $1, 2, 3, \ldots$ を使うこととする。多くの場合は，直近で束縛されるセルフ変数が参照されるので，簡潔に記載できる。上記の構文は，$\varsigma\kappa$ 計算 ([1, 2]) とほとんど同

Modifiable Continuations in Object Calculus

Emiko KUMA[†], Daisuke MATSUMOTO[‡], Shin-ya NISHIZAKI[†], Takuo WATANABE[†]
[†]Department of Computer Science,
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology.
[‡]Department of Computer Science,
Faculty of Engineering,
Tokyo Institute of Technology.
2-12-1-W8-69, O-okayama, Meguro-ku, Tokyo, 152-8552, JAPAN.
emy@psg.cs.titech.ac.jp, matsumoto@lambda.cs.titech.ac.jp, {nisizaki, takuo}@cs.titech.ac.jp

じである。違いは，ラベル l に対応して文字列という構文 "l" が用意されている点である。文字列はラベルが具現化されたものである。継続を変更する機構は，操作的意味論を拡張することにより実現されている。評価文脈は次のように定義される。

$$E[\,]::=[\,]\mid E[\,].l\mid E[\,].l\Leftarrow\varsigma b$$

そして，具現化された評価文脈は次のような項として定義される。

$$\begin{aligned}\lceil E[\,]\rceil ::= &\ [\,]\mid [mtd=\text{"l"}, obj=\lceil E[\,]\rceil]\\ &\mid [mtd=\text{"l"}, body=\varsigma b, obj=\lceil E[\,]\rceil]\end{aligned}$$

両者の三種はそれぞれ互いに対応していている。例えば $\lceil[\,].l\rceil$ は，$[mtd=\text{"l"}, obj=[\,]]$ に対応し，これの穴のところに a を埋めたもの $\lceil[a].l\rceil$ は，$[mtd=\text{"l"}, obj=a]$ を表す。

そして，簡約は，従来のオブジェクト計算と同様に，

$$\frac{a\rightarrowtail a'}{E[a]\to E[a']},$$

$$[l_i=\varsigma(x_i)b_i^{i\in 1..n}].l\rightarrowtail b\{\!\{x_j\leftarrow[l_i=\varsigma(x_i)b_i^{i\in 1..n}]\}\!\},$$

$$\begin{aligned}&[l_i=\varsigma(x_i)b_i^{i\in 1..n}].l_j\Leftarrow\varsigma(y)b\\ &\quad\rightarrowtail\ [l_j=\varsigma(y)b,\ l_i=\varsigma(x_i)b_i^{i\in 1..n-\{j\}}],\end{aligned}$$

の他，次のような継続に関する簡約が与えらえる。

$$E[\mathcal{A}(\lceil a\rceil)]\rightarrowtail a$$

$$\begin{aligned}E[\kappa(l_{val},l_{arg}l_{data})]\rightarrowtail E[a\{x\leftarrow[&l_{val}=\varsigma\mathcal{A}(1.l_{data}),\\ &l_{arg}=\varsigma 1.l_{arg},\\ &l_{data}=\varsigma\lceil E[1.l_{arg}]\rceil]\}]\end{aligned}$$

このように評価文脈を具現化された「継続オブジェクト」とすることにより，具現化したあと，フィールド l_{data} を変更することにより継続オブジェクトの内部を変更することが可能になる。

3 おわりに

3.1 変更可能なファーストクラス継続の利用例

本稿では，紙面の都合により，具体的な例について記載できなかったが，具体例としては，文献 [3] で紹介されているような「電卓プログラム」においての例を得ている。電卓プログラムでは，継続は未執行のキー操作列に対応する。そして，本研究での継続の書き換えにより，キー操作の内容を変更することが可能になっている。

3.2 ファーストクラス継続の変更方法

本稿では，紙面の都合により，具現化された継続に対する変更は，フィールド名へのアップデート操作を介して行っている。この仕組みだけでは，ファーストクラス継続に対する変更は限定的なものであるし，使用しにくい

3.3 具現化された継続を反映する際のエラー

本体系で与えた，アボートに関する簡約規則 $E[\mathcal{A}(\lceil a\rceil)]\rightarrowtail a$ は，a を $H[b]$ とすると

$$E[\mathcal{A}(\lceil H[b]\rceil)]\rightarrowtail H[b]$$

と表すことができる。この規則は，$\mathcal{A}$ の引数である a がある評価文脈 $H[\,]$ が存在して，その具現化した評価文脈 $\lceil H[\,]\rceil$ の穴に a を嵌めこんだものであることを意味している。したがって，$\mathcal{A}$ の引数として許されるオブジェクトには構文的な制限が設けられている。

継続λ計算 [4] では，アボートの簡約において失敗することはないのに対して，$\varsigma\kappa m$ 計算では，具現化された継続オブジェクトの変更が妥当でないと，アボートに失敗してしまうのである。継続オブジェクトの変更の妥当性を静的に決定できるような仕組みを考える必要があり，それは重要な今後の課題であると言える。

参考文献

[1] 小田 崇史 西崎 真也. ファーストクラス継続をもつオブジェクト計算, 2003. (口頭発表).

[2] Shin-ya NISHIZAKI and Ritsuya IKEDA. Object calculus with first-class continuations, 2011. Submitted.

[3] M. Abadi and L. Cardelli. *A Theory of Objects*. Springer-Verlag, 1996.

[4] Matthias Felleisen, Daniel P. Friedman, Eugene Kohlbecker, and Bruce F. Duba. A syntactic theory of sequential control. *Theoretical Computer Science*, 52(3), 1987.

[5] Michael Sperber, R. Kent Dybvig, Matthew Flatt, and Anton van Straaten, editors. *Revised [6] Report on the Algorithmic Language Scheme*. Cambridge University Press, 2010.