論文の内容の要旨

論文題目　Meta-continuation Semantics via Meta-lambda Calculus
(メタラムダ計算を用いたメタ継続意味論)

氏　　名　戸澤　一成

本研究の目的は，限定継続により記述される制御構造が内包する動的性質を特徴付けることである．

継続は計算過程のある時点での状態を意味する概念であり，これを一級オブジェクトとして扱うことにより様々な制御構造を表現できることが知られている．継続の表す情報を細分化し部分毎に扱えるよう一般化された概念が限定継続である．限定継続を扱うことで，通常の継続よりも効率的に制御構造が記述でき，またより多くの計算効果を表現できるようになる．限定継続の表現能力を特徴付けるため，値呼び $\lambda\mu\hat{\mathsf{tp}}$ 計算が提唱された [AHS09]．値呼び $\lambda\mu\hat{\mathsf{tp}}$ 計算は限定継続を一級オブジェクトとして扱うことができる計算体系である．値呼び $\lambda\mu\hat{\mathsf{tp}}$ 計算は一級継続をもつ体系である値呼び $\lambda\mu$ 計算の拡張の形で定義されており，違いは特別な継続変数 $\hat{\mathsf{tp}}$ の存在のみである．この変数 $\hat{\mathsf{tp}}$ は他の変数と異なり動的性質を持つため，動的継続変数とよばれる．

変数 $\hat{\mathsf{tp}}$ の動的性質は以下のように説明される．

$$\begin{aligned}\mathbf{let}\ x := (\lambda y.\mu\delta.[\alpha]y)\ \mathbf{in}\ \mu\alpha.[\alpha]x \to &\quad \mu\beta.[\beta](\lambda y.\mu\delta.[\alpha]y)\\ \mathbf{let}\ x := (\lambda y.\mu\delta.[\hat{\mathsf{tp}}]y)\ \mathbf{in}\ \mu\hat{\mathsf{tp}}.[\hat{\mathsf{tp}}]x \to &\quad \mu\hat{\mathsf{tp}}.[\hat{\mathsf{tp}}](\lambda y.\mu\delta.[\hat{\mathsf{tp}}]y)\end{aligned}$$

ラムダ計算では通常，変数の束縛構造が静的に定まるよう，代入の際に変数名を書き換えることで名前の衝突を防いでいる．$\lambda\mu\hat{\mathsf{tp}}$ 計算においても通常変数はこのルールに従うため変数 α は代入に伴い名前が書き換えられる．一方で，変数 $\hat{\mathsf{tp}}$ については変数名の書き換えが行われない．そのため $\hat{\mathsf{tp}}$ に関する束縛構造は計算に伴い動的に変化する．

値呼び $\lambda\mu\hat{\mathsf{tp}}$ 計算への意味付けにはメタ継続意味論に基づいた CPS 変換が用いられる [DF90]. CPS 変換は計算体系の意味論を定めるための手法であるが，その主要な役割は，評価戦略を与えることである．値呼び $\lambda\mu\hat{\mathsf{tp}}$ 計算の CPS 変換は内部的には２回の値呼び CPS 変換を行っており，CPS2 変換とよばれる．CPS2 変換は，$\lambda\mu\hat{\mathsf{tp}}$ 計算固有の特徴を扱う必要があるため，次の２つの役割を持つ．

1. 評価戦略を与える．
2. 変数継続変数 $\hat{\mathsf{tp}}$ に対する意味論を与える．

CPS2 変換はこの２つの役割を１回の変換で同時に行う．そのため，限定継続の持つ動的性質を個別に考慮する場合，CPS2 変換を用いるだけでは難しい．そのような場合には CPS2 変換を明示的に分解して考えるのが有用であり，先行研究においていくつかの分解が提唱されている．例えば，高階の限定継続に対する意味論を構成するために導入された Kameyama の分解 [Kam07] や，動的継続変数を複数持つ計算体系の意味論を与えるのに用いられた Downen, Ariola の分解 [DA12]，動的性質の存在により複雑化した型システムの構造を解析するため用いられた Ariola, Herbelin, Sabry の分解 [AHS09] がある．

本論文の主題は，限定継続の持つ動的な性質を特徴付けるような CPS2 変換の分解で，それぞれの変換が健全かつ完全であるものを与えることである．限定継続の動的性質を個別に解析するため，２回の変換の中間システムを形式化し，分解で生じたそれぞれの変換が与える意味論を個別に扱いたい．中間システムを適切に与えることによって，役割１，役割２をそれぞれ独立に果たすような２つの変換に対し，健全性，完全性が成り立つようなものを構成することが目標である．

本研究では，分解の中間システムとしてメタラムダ計算が有用であることを提案する．メタラムダ計算は，メタ変数とテキスト代入を形式的に扱う計算体系である．テキスト代入はラムダ計算の通常の代入と比較して次のように説明される．

$$
\begin{aligned}
(\lambda x.\lambda y.x)(xy) &\to \quad \lambda z.xy \\
(\lambda X.\lambda y.X)(xy) &\to \quad \lambda y.xy
\end{aligned}
$$

通常変数 x への代入では変数名の衝突を回避するための名前の書き換えが行われるのに対し，メタ変数 X への代入では項が文字通り代入される．変数名の書き換えの有無に関して，テキスト代入は動的継続変数の代入と類似している．この類似性に注目することで，新たな変換 $\mathcal{C}$ が定義される．変換 $\mathcal{C}$ は，値呼び $\lambda\mu$ 計算の CPS 変換をメタラムダ計算のレベル構造を用いて自然に拡張したものになっている．そのため，変換 $\mathcal{C}$ は役割 1 を担う変換であると考えられる．

変換 $\mathcal{C}$ による意味論を個別に扱うためには中間システムに理論を与える必要があるが，これはレベル横断的計算機構を持つメタラムダ計算を用いることで可能となった．変換 $\mathcal{C}$ はメタラムダ計算のメタ層をオブジェクトレベルとみなして扱うため，変換で生じた項を計算するにはメタ変数を含む項の計算が必要である．通常，メタラムダ計算の簡約の順序はレベルに関する制限がかけられているため，このような項の計算は許されていないが，近年，レベル横断的計算機構を持つメタラムダ計算が提案された [Tob15]．レベル横断的計算機構によってメタ変数を持つ項の計算が可能となり，これにより変換 $\mathcal{C}$ を個別に扱うことができる．

第３章では変換 $\mathcal{C}$ のターゲットとなるメタラムダ計算 $\lambda\hat{\mathsf{d}}^*$ を与える．上記の理由から，この体系はレベル横断的計算機構を持つ必要があるため Tobisawa のメタラムダ計算を基に定義される．変換 $\mathcal{C}$ はラムダ計算の値呼び CPS 変換の自然な拡張なので，メタラムダ計算 $\lambda\hat{\mathsf{d}}^*$ は η 簡約をもつラムダ計算を部分体系として含んでいる必要がある．一方で，Tobisawa の体系は変数名を書き換える機構を持たず，また η 簡約が与えられていない．そのため，α 変換と η 簡約を新たに定義し加えた．その上で，メタラムダ計算 $\lambda\hat{\mathsf{d}}^*$ が合流性を満たすことを示した．

第４章では，変換 $\mathcal{C}$ がメタラムダ計算 $\lambda\hat{\mathsf{d}}^*$ に関して健全かつ完全であることを示す．変換 $\mathcal{C}$ による値呼び $\lambda\mu\hat{\mathsf{tp}}$ 計算の意味論はターゲットの評価戦略に依存しないため，この結果は変換 $\mathcal{C}$ がメタラムダ計算をターゲットに持つ CPS 変換であることを意味している．本研究では，第３章で示した合流性を用いることで，メタラムダ計算の $\beta\eta$ 簡約の定める等式理論 $=_{\beta\eta}$ に関して変換 $\mathcal{C}$ が健全かつ完全であることを示した．

第５章では，CPS2 変換の分解におけるもう一つの変換 $\mathcal{D}$ を与える．本研究では変換 $\mathcal{D}$ を定義し，変換 $\mathcal{C}$ と変換 $\mathcal{D}$ の合成が CPS2 変換と一致していることを示した．また，変換 $\mathcal{C}$ で得られる項に関して，メタラムダ計算の等式理論が変換 $\mathcal{D}$ に対し健全かつ完全であることを示した．これは，変換 $\mathcal{D}$ が役割２を担う変換であり，本研究の分解が CPS2 変換の２つの役割を分離しているものであることを意味している．

第６章では，変換 $\mathcal{C}$ が限定継続体系とメタラムダ計算の型システムを対応付けることを見る．限定継続を扱う計算体系は動的束縛に対しても型の整合性を考える必要があるが，そのためには通常の単純型システムでは型情報が不十分である．Danvy, Filinski によって導入された型システムは通常よりも多くの型情報を保持することで，継続の答の型が変化するような計算を表す項に対しても型付けが可能となった [DF89]．この型システムにおける型判定は以下の形である．

$$\Gamma\backslash S_1 \vdash M : T\backslash S_2|\Delta,$$

通常の型システムとの違いは，型 S_1 と型 S_2 をパラメータとして保持していることである．一方，メタラムダ計算の型システムとしては様相論理を用いたものが知られている [NPP08]．様相演算子を用いることで，

項のコンテクストに対して型情報を与えることができる．この型情報によりメタ変数の出現を制限することで型エラーを起こすようなテキスト代入が生じるのを防いでいる．本研究では，第３章で構成したメタラムダ計算 $\lambda\hat{d}^*$ に対し，様相論理を基にした型システムを与え，この体系が簡約に関して型を保存する性質を満たすことを示した．その上で変換 $\mathcal{C}$ を型システムへ拡張し，型付けに関して健全であることを示した．

参考文献

[AHS09] Zena M. Ariola, Hugo Herbelin, and Amr Sabry. A type-theoretic foundation of delimited continuations. *Higher-Order and Symbolic Computation*, 22(3):233–273, 2009.

[DA12] Paul Downen and Zena M. Ariola. A systematic approach to delimited control with multiple prompts. In *Proceedings of Programming Languages and Systems - 21st European Symposium on Programming (ESOP 2012)*, pages 234–253, 2012.

[DF89] Olivier Danvy and Andrzej Filinski. A functional abstraction of typed contexts. Technical Report DIKU-89/12, University of Copenhagen, 1989.

[DF90] Olivier Danvy and Andrzej Filinski. Abstracting control. In *Proceedings of the 1990 ACM Conference on LISP and Functional Programming*, pages 151–160, 1990.

[Kam07] Yukiyoshi Kameyama. Axioms for control operators in the cps hierarchy. *Higher-Order and Symbolic Computation*, 20(4):339–369, 2007.

[NPP08] Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. Contextual modal type theory. *ACM Transactions on Computational Logic*, 9(3):23:1–23:49, 2008.

[Tob15] Kazunori Tobisawa. A meta lambda calculus with cross-level computation. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2015)*, pages 383–393, 2015.