TPP 2017

Formal Verification of the Correspondence between Call-by-Need and Call-by-Name

Masayuki Mizuno
Joint work with Eijiro Sumii

Tohoku University

December 7, 2017

動機: 非正格言語の仕様と実装のギャップ

- 名前呼び: 非正格言語の (ハイレベルな) 仕様
- 必要呼び: 非正格言語の実装

これらの対応を検証したい

λ 計算の基本: full-β 簡約

- **簡約基は複数存在しうる**
  - $\underline{(\lambda xy.\ y)\ ((\lambda x.\ xx)\ (\lambda x.\ xx))}$
  - $(\lambda xy.\ y)\ \underline{((\lambda x.\ xx)\ (\lambda x.\ xx))}$
- **下手に簡約基を選ぶと無限ループに**
  - $\underline{(\lambda xy.\ y)\ ((\lambda x.\ xx)\ (\lambda x.\ xx))}$
    $\xrightarrow{\beta} \lambda y.\ y$
  - $(\lambda xy.\ y)\ \underline{((\lambda x.\ xx)\ (\lambda x.\ xx))}$
    $\xrightarrow{\beta} (\lambda xy.\ y)\ ((\lambda x.\ xx)\ (\lambda x.\ xx))$

非正格な言語の理論的背景: 名前呼び

定義 (名前呼び)

関数呼び出し以前に引数を簡約せず，λ 抽象の中も簡約しない評価戦略

- full-β 簡約で λ 抽象に評価できる項は名前呼びでも λ 抽象まで評価できる

$$\underline{(\lambda xy.\ x)\ ((\lambda x.\ xx)\ (\lambda x.\ xx))} \xrightarrow{\text{name}}_{*} \lambda y.\ y$$

名前呼びの問題点

- 関数引数の簡約基が複製される
  - 余計な簡約が必要になることも

- $(\lambda x.\ x x)\ \underline{((\lambda x.\ x)\ (\lambda x.\ x))}$
  $\xrightarrow{\beta} (\lambda x.\ x x)\ (\lambda x.\ x)$
- $\underline{(\lambda x.\ x x)\ ((\lambda x.\ x)\ (\lambda x.\ x))}$
  $\xrightarrow{\text{name}} (\lambda x.\ x)\ (\lambda x.\ x)\ ((\lambda x.\ x)\ (\lambda x.\ x))$

名前呼びの改良: 必要呼び

- 一度関数引数を評価したら値を覚えて使い回す
  - 余計な簡約が不要に

$$(\lambda x.\ x x)\ ((\lambda x.\ x)\ (\lambda x.\ x))$$
$$\xrightarrow{\text{need}} \mathbf{let}\ x = (\lambda x.\ x)\ (\lambda x.\ x)\ \mathbf{in}\ x\ x$$
$$\xrightarrow{\text{need}}_{*} \mathbf{let}\ x = \lambda x.\ x\ \mathbf{in}\ x\ x$$

- 名前呼びと振る舞いが一致してほしい

本研究の概要

- 必要呼び λ 計算 [Ariola+ 1995] を検証に適するよう変形し Coq で定式化
  - De Bruijn インデックス
  - 評価文脈の排除
- 名前呼びとの評価の対応に既存研究より簡単な証明を考案，Coq で定式化
  - 標準化定理

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. 主定理の証明のアウトライン
3. Coq による検証のポイント
4. まとめ

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. 主定理の証明のアウトライン
3. Coq による検証のポイント
4. まとめ

名前呼び λ 計算

Terms	L, M, N	$::=$	$x \mid V \mid M\ N$
Values (WHNF)	V	$::=$	$\lambda x.M$
Evaluation contexts	E_{n}	$::=$	$[] \mid E_{\mathrm{n}}\ M$

$$(\beta) \qquad (\lambda x.M)N \ \rightarrow\ M[x \mapsto N]$$

- **簡約は決定的**
- **行き詰まり状態は全て $E_{\mathrm{n}}[x]$ の形**

補題 (名前呼び λ 計算の決定性)

- $\xrightarrow{\text{name}}$ は部分関数
- $E_{\mathrm{n}}[x] = E'_{\mathrm{n}}[y]$ ならば $x = y$
- 任意の M について，以下のうちただ一つが成り立つ
  1. M は値
  2. $M = E_{\mathrm{n}}[x]$ となる E_{n} と x が存在する
  3. M は $\xrightarrow{\text{name}}$ で簡約できる

名前呼びの性質: 標準化定理の 2 つの系

定理 (正規化定理)

M は full-β 簡約で λ 抽象に評価できるならば，M は名前呼びの簡約でも λ 抽象に評価できる

定理 (準正規化定理)

M は full-β 簡約で λ 抽象に評価できるならば，$\xrightarrow{\text{name}} \circ \xrightarrow{\beta}_{*}$ による M の簡約は必ず止まる

- 必要呼びとの対応の証明に用いる

必要呼び λ 計算 [Ariola+ 1995]

Terms	M, N	$::=$	$x \mid V \mid M\ N \mid \textbf{let}\ x = M\ \textbf{in}\ N$
Values	V	$::=$	$\lambda x.\ M$
Answers	A	$::=$	$V \mid \textbf{let}\ x = M\ \textbf{in}\ A$
Evaluation contexts	E, E'	$::=$	$[] \mid E\ M \mid \textbf{let}\ x = M\ \textbf{in}\ E$
			$\mid \textbf{let}\ x = E\ \textbf{in}\ E'[x]$

$$
\begin{array}{lrcl}
(I) & (\lambda x.M)N & \to & \textbf{let}\ x = N\ \textbf{in}\ M \\
(V) & \textbf{let}\ x = V\ \textbf{in}\ E[x] & \to & \textbf{let}\ x = V\ \textbf{in}\ E[V] \\
(C) & (\textbf{let}\ x = M\ \textbf{in}\ A)\ N & \to & \textbf{let}\ x = M\ \textbf{in}\ A\ N \\
(A) & \textbf{let}\ y = (\textbf{let}\ x = M\ \textbf{in}\ A)\ \textbf{in}\ E[y] & \to & \\
 & \textbf{let}\ x = M\ \textbf{in}\ \textbf{let}\ y = A\ \textbf{in}\ E[y] & &
\end{array}
$$

$\xrightarrow{\mathrm{I}}$ **簡約規則 (I) のみ用いる簡約**

$\xrightarrow{\mathrm{VCA}}$ **簡約規則 (V)，(C) と (A) のみ用いる簡約**

補題 (必要呼び λ 計算の決定性)

- $\xrightarrow{\mathrm{I}}$ は部分関数
- $\xrightarrow{\mathrm{VCA}}$ は部分関数
- $E[x] = E'[y]$ ならば $x = y$
- 任意の M について，以下のうちただ一つが成り立つ
  1. M は answer
  2. $M = E[x]$ となる E と x が存在する
  3. M は $\xrightarrow{\mathrm{I}}$ で簡約できる
  4. M は $\xrightarrow{\mathrm{VCA}}$ で簡約できる

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. **主定理の証明のアウトライン**
3. Coq による検証のポイント
4. まとめ

本研究の主定理: 名前呼びと必要呼びの評価の対応

定理 (必要呼び評価の健全性)

M が必要呼びの簡約によって answer に評価できるならば，M に対応する項 N は名前呼びの簡約によって λ 抽象に評価される

定理 (必要呼び評価の完全性)

M が名前呼びの簡約によって λ 抽象に評価できるならば，M に対応する項 N は必要呼びの簡約によって answer に評価される

先行研究の証明方法

- `[Ariola+ 1995]` は Ariola らの研究と Maraist らの研究を合わせたもの

Ariola と Felleisen の証明 [Ariola+ 1997]

- Informal に定義されたグラフに基づく

Maraist らの証明 [Maraist+ 1998]

- マーク付きの簡約と簡約位置を陽に扱うのが煩雑
- 本研究で一部参考

定義 (項同士の対応関係)

$$
\begin{aligned}
x^{\pitchfork} &= x \\
(\lambda x.M)^{\pitchfork} &= \lambda x.M^{\pitchfork} \\
(M\ N)^{\pitchfork} &= M^{\pitchfork}\ N^{\pitchfork} \\
(\mathbf{let}\ x = M\ \mathbf{in}\ N)^{\pitchfork} &= N^{\pitchfork}[x \mapsto M^{\pitchfork}]
\end{aligned}
$$

- let を全て展開する関数
- マークを除くと [Maraist+ 1998] と一致

補題 (1 ステップの対応)

- $A^{\pitchfork}$ は値
- 任意の E と x について，$E[x]^{\pitchfork} = E_{\mathrm{n}}[x]$ となる E_{n} が存在
- $M \xrightarrow{\mathrm{VCA}} N$ ならば $M^{\pitchfork} = N^{\pitchfork}$
- $M \xrightarrow{\mathrm{I}} N$ ならば $M^{\pitchfork} \xrightarrow{\mathrm{name}} \circ \xrightarrow{\beta}_{*} N^{\pitchfork}$

再掲: 名前呼びと必要呼びの評価の対応

定理 (必要呼び評価の健全性)

$M \xrightarrow{\text{need}}_* A$ ならば，$M^{\pitchfork} \xrightarrow{\text{name}}_* V$ となるような値 V が存在する

定理 (必要呼び評価の完全性)

$M^{\pitchfork} \xrightarrow{\text{name}}_* V$ ならば，$M \xrightarrow{\text{need}}_* A$ となるような answer A が存在する

- 評価結果 A と V の間には $V \xrightarrow{\beta}_* A^{\pitchfork}$ が成り立つ (本発表では詳細は省きます)

必要呼び評価の健全性の証明

Proof.

$M \xrightarrow{\mathrm{need}}_* A$ と仮定すると，1ステップの対応により $M^{\pitchfork} \xrightarrow{\beta}_* A^{\pitchfork}$

1ステップの対応により $A^{\pitchfork}$ は値であるから，正規化定理により $M^{\pitchfork} \xrightarrow{\mathrm{name}}_* V$ となるような値 V が存在する □

完全性の証明の困難な点

- Administrative な簡約が止まらないかもしれない
  - $M \xrightarrow{\text{VCA}} N$ ならば $M^{\pitchfork} = N^{\pitchfork}$
- let で共有されている部分がすべて簡約される
  - $M \xrightarrow{\text{I}} N$ ならば $M^{\pitchfork} \xrightarrow{\text{name}} \circ \xrightarrow{\beta}_{*} N^{\pitchfork}$
  - λ 抽象の中も簡約されうる

$\xrightarrow{\text{VCA}}$ の正規化性

Proof.

[Maraist+ 1998] の重み付けを変形したものにより証明

$$
\begin{aligned}
\| \ x \ \|_s &= s(x) \\
\| \ \lambda x.M \ \|_s &= \| \ M \ \|_{s \circ [x \mapsto 1]} \\
\| \ M \ \ N \ \|_s &= 2 \ \| \ M \ \|_s + 2 \ \| \ N \ \|_s \\
\| \ \mathbf{let} \ x = M \ \mathbf{in} \ N \ \|_s &= 2 \ \| \ M \ \|_s + \| \ N \ \|_{s \circ [x \mapsto 1 + \| M \|_s]}
\end{aligned}
$$

□

必要呼びの完全性の証明 (1/3)

Proof.

$M^{\pitchfork} \xrightarrow{\text{name}}_* V$ と仮定する．

まず，$\xrightarrow{\text{need}}$ によって M が正規化されることを示す．
準正規化定理により，$\xrightarrow{\text{name}} \circ \xrightarrow{\beta}_*$ による $M^{\pitchfork}$ の簡約は停止する．よって，$\xrightarrow{\text{I}}$ による簡約は $\xrightarrow{\text{name}} \circ \xrightarrow{\beta}_*$ による簡約に対応している (1 ステップの対応) ため，停止する．
また，$\xrightarrow{\text{VCA}}$ の簡約も停止するため，$\xrightarrow{\text{need}}$ $(= \xrightarrow{\text{I}} \cup \xrightarrow{\text{VCA}})$ による簡約は停止する．

必要呼びの完全性の証明 (2/3)

Proof.

次に，正規形 N が answer であることを示す．
必要呼び λ 計算の決定性により，N は answer または行き詰まり状態 $E[x]$ である．
N は行き詰まり状態 $E[x]$ と仮定すると，1 ステップの対応により $E[x]^{\pitchfork} = E_{\mathrm{n}}[x]$ となるような E_{n} が存在し，加えて $M^{\pitchfork} \xrightarrow{\beta}_{*} N^{\pitchfork}$ が成り立つ．

必要呼びの完全性の証明(3/3)

Proof.

ここで，full-β 簡約の合流性により，$E_{\mathrm{n}}[x] \xrightarrow{\beta}_{*} L$ かつ $A \xrightarrow{\beta}_{*} L$ となるような L が存在する．名前呼び λ 計算における値であることと行き詰まり状態であることは full-β 簡約により保存されるため，L は値かつ行き詰まり状態となってしまう．これは名前呼び λ 計算の決定性に矛盾する． □

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. 主定理の証明のアウトライン
3. Coq による検証のポイント
   - 束縛の表現
   - 評価文脈の除去
4. まとめ

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. 主定理の証明のアウトライン
3. Coq による検証のポイント
   - 束縛の表現
   - 評価文脈の除去
4. まとめ

名前による束縛の表現の問題点

- 項の等価性が Coq の等価性でなく α 等価性となる

$$\lambda x.\lambda y.x \stackrel{\alpha}{=} \lambda a.\lambda b.a$$

- Capture を避ける必要がある

$$\begin{aligned}&[y \mapsto x](\lambda x.y)\\ \stackrel{\alpha}{\neq}\ & \lambda x.x\end{aligned}$$

De Bruijn インデックスによる束縛の表現

- 内側から数えて何番目の束縛かで表現

$$\lambda x.x\ (\lambda y.x\ y)\ \Rightarrow\ \lambda.0\ (\lambda.1\ 0)$$

- α 等価な式は文面上も同じ

$$\lambda x.\lambda y.x\ \Rightarrow\ \lambda.\lambda.1$$
$$\lambda a.\lambda b.a\ \Rightarrow\ \lambda.\lambda.1$$

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. 主定理の証明のアウトライン
3. Coq による検証のポイント
   - 束縛の表現
   - 評価文脈の除去
4. まとめ

評価文脈で困る例: 必要呼び λ 計算の決定性

```
Lemma answer_or_stuck_or_reducible M :
  answer M
  \/ (exists E x,
    evalctx E /\ M = E.[tvar x] /\ bv E <= x)
  \/ (exists E L N,
    evalctx E /\ M = E.[L] /\ reduceI L N)
  \/ (exists E L N,
    evalctx E /\ M = E.[L] /\ reduceVCA L N).
```

- M についての帰納法で証明を試みる

必要呼び λ 計算の決定性: $M = x$ のケース

```
x : var
============================
answer M
\/ (exists E y,
  evalctx E /\ tvar x = E.[tvar y]
  /\ bv E <= y)
\/ (exists E L N,
  evalctx E /\ tvar x = E.[L] /\ reduceI L N)
\/ (exists E L N,
  evalctx E /\ tvar x = E.[L]
  /\ reduceVCA L N)
```

- E が $[]$ のとき，明らかに成り立つ

必要呼び λ 計算の決定性: $M = x$ のケース

- しかし，自動証明に失敗してしまう

```
Coq < eauto.
x : var
============================
answer M
\/ (exists E y,
  evalctx E /\ tvar x = E.[tvar y]
  /\ bv E <= y)
\/ (exists E L N,
  evalctx E /\ tvar x = E.[L] /\ reduceI L N)
\/ (exists E L N,
  evalctx E /\ tvar x = E.[L]
  /\ reduceVCA L N)
```

なぜ自動証明に失敗するか？

```
exists E y, evalctx E /\
  tvar x = E.[tvar y] /\ bv E <= y
```

を証明するためには，

```
tvar x = E.[tvar y]
```

となるような E を探さなくてはならない

$$\Downarrow$$

高階単一化が必要

解決策: 評価文脈の除去

- 評価文脈のルールを簡約規則に展開
  - $\xrightarrow{\beta}$, $\xrightarrow{\mathrm{name}}$, $\xrightarrow{\mathrm{I}}$ と $\xrightarrow{\mathrm{VCA}}$
- 行き詰まり状態を表す述語の導入
  - $\mathbf{needs}_{\mathrm{n}}(M, x)$ $(\iff \exists E.M = E_{\mathrm{n}}[x])$
    と $\mathbf{needs}(M, x)$ $(\iff \exists E.M = E[x])$
- 評価規則
  (V) $\mathbf{let}\ x = V\ \mathbf{in}\ E[x] \rightarrow \mathbf{let}\ x = V\ \mathbf{in}\ E[V]$
  を代入に変更

自動証明が回る例: 必要呼び λ 計算の決定性

```
Lemma answer_or_stuck_or_reducible M :
  answer M \/
  (exists x, needs M x) \/
  (exists N, reduceI M N) \/
  (exists N, reduceVCA M N).
Proof.
  induction M as
    [|? [Hanswer|[[]|[[]|[]]]]]
    ||? [Hanswer|[[]|[[]|[]]]]
      ? [|[[[]]|[[]|[]]]]]]; eauto 6;
    inversion Hanswer; subst; eauto 6.
Qed.
```

完全性の証明の補足

- $\xrightarrow{\text{name}} \circ \xrightarrow{\beta}_{*}$ による $M^{\Cap}$ の簡約が必ず停止することは accessibility predicate $\texttt{Acc}$ で表現
- 必要呼び簡約の停止性は，$\texttt{Acc}$ についての帰納法の内側で $\xrightarrow{\text{VCA}}$ が停止することによる整礎帰納法を回して証明

アウトライン

1. 対象言語: 名前呼び λ 計算と必要呼び λ 計算
2. 主定理の証明のアウトライン
3. Coq による検証のポイント
4. まとめ

まとめ

- **必要呼び λ 計算** [Ariola+ 1995] **を検証に適するよう変形し** Coq **で定式化**
  - De Bruijn **インデックス**
  - **評価文脈の排除**
- **名前呼びとの評価の対応に既存研究より簡単な証明を考案，**Coq **で定式化**
  - **標準化定理**