必要呼び意味論と名前呼び意味論の対応の形式的検証

水野 雅之

情報科学研究科 情報基礎科学専攻
住井・松田研究室

指導教員：住井 英二郎 教授

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

    $(\lambda x.\ x)\ (\lambda x.\ x)\ ((\lambda x.\ x)\ (\lambda x.\ x))$

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

$$
\begin{aligned}
&(\lambda x.\ x x)\ ((\lambda x.\ x)\ (\lambda x.\ x)) \\
&\xrightarrow{\text{need}} \mathbf{let}\ x = (\lambda x.\ x)\ (\lambda x.\ x)\ \mathbf{in}\ x\ x \\
&\xrightarrow{\text{need}}\!\!\!\!\to \mathbf{let}\ x = \lambda x.\ x\ \mathbf{in}\ x\ x
\end{aligned}
$$

- 名前呼びと振る舞いが一致してほしい

本研究の貢献

- 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 [Curry&Feys 1958] を利用

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. 貢献 1：主定理の簡潔化された証明
3. 貢献 2：必要呼びラムダ計算の形式的検証に適した定式化
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. 貢献 1：主定理の簡潔化された証明
3. 貢献 2：必要呼びラムダ計算の形式的検証に適した定式化
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**
- stuck **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

標準化定理 [Curry&Feys 1958]

定義 (標準簡約列)

$i > j$ であるような全ての i と j について，簡約位置 Δ_i が Δ_j よりも外側かつ左側にある場合，簡約列 $M_1 \xrightarrow[\Delta_1]{\beta} M_2 \xrightarrow[\Delta_2]{\beta} \cdots \xrightarrow[\Delta_{n-1}]{\beta} M_n$ を標準簡約列という

定理 (標準化定理)

$M \overset{\beta}{\twoheadrightarrow} N$ ならば，M から N への標準簡約列が存在する

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，$\xrightarrow{\text{name}} \circ \overset{\beta}{\twoheadrightarrow}$ 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	$::=$	$x \mid V \mid M\ N \mid \mathbf{let}\ x = M\ \mathbf{in}\ N$
Values	V	$::=$	$\lambda x.\ M$
Answers	A	$::=$	$V \mid \mathbf{let}\ x = M\ \mathbf{in}\ A$
Evaluation contexts	E, E'	$::=$	$[] \mid E\ M \mid \mathbf{let}\ x = M\ \mathbf{in}\ E$
			$\mid \mathbf{let}\ x = E\ \mathbf{in}\ E'[x]$

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

$\xrightarrow{\mathrm{I}}$ **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. **貢献 1：主定理の簡潔化された証明**
3. 貢献 2：必要呼びラムダ計算の形式的検証に適した定式化
4. まとめ

主定理：名前呼びと必要呼びの対応

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[1997] の証明

- Informal に定義されたグラフに基づく

Maraist ら [1998] の証明

- 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)

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

- 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 $M^\pitchfork \xrightarrow{\mathrm{name}} \circ \xrightarrow{\beta}\!\!\!\!\twoheadrightarrow N^\pitchfork$

再掲：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

再掲：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，$\xrightarrow{\text{name}} \circ \xrightarrow{\beta}\!\!\!\!\rightarrow$ による M の簡約は必ず止まる

健全性の証明.

$$
\begin{tikzcd}
M \arrow[rr, twoheadrightarrow, "\mathrm{need}"] \arrow[d, dotted, "\pitchfork"'] & & A \arrow[d, dotted, "\pitchfork"'] \\
M^{\pitchfork} \arrow[r, twoheadrightarrow, "\mathrm{name}"] \arrow[rr, twoheadrightarrow, bend right, "\beta"'] & V & A^{\pitchfork}
\end{tikzcd}
$$

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 $M^{\pitchfork} \xrightarrow{\mathrm{name}}\!\!\!\!\twoheadrightarrow V$ 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
  - $M \xrightarrow{\mathrm{VCA}} N$ 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
  - $M \xrightarrow{\mathrm{I}} N$ ならば $M^{\pitchfork} \xrightarrow{\mathrm{name}} \circ \xrightarrow{\beta}\!\!\!\!\rightarrow N^{\pitchfork}$
  - λ 抽象の中も簡約されうる

$\xrightarrow{\text{VCA}}$ の正規化性の証明.

[Maraist+ 1998] の重み付けを変形したものにより証明

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

$M \xrightarrow{\text{VCA}} N$ ならば $\| M \|_s > \| N \|_s$ □

完全性の証明 (1/2).

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

$\boxed{\text{まず } \xrightarrow{\text{need}} \text{ によって } M \text{ が正規化されることを示す}}$

$$M \xrightarrow{\text{need}} - - - - \xrightarrow{\text{need}} N$$

完全性の証明 (1/2).

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

$\boxed{\text{まず} \xrightarrow{\text{need}} \text{によって } M \text{ が正規化されることを示す}}$

$\xrightarrow{\text{need}}\!\!\!\!\rightarrow \; = (\xrightarrow{\text{I}} \cup \xrightarrow{\text{VCA}})^* = (\xrightarrow{\text{VCA}}\!\!\!\!\rightarrow \circ \xrightarrow{\text{I}})^*$ が成り立つ

$$M \xrightarrow{\text{VCA}}\!\!\!\!\rightarrow \xrightarrow{\text{I}} - - - \xrightarrow{\text{VCA}}\!\!\!\!\rightarrow \xrightarrow{\text{I}} N$$

完全性の証明 (1/2).

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

$\boxed{\text{まず } \xrightarrow{\text{need}} \text{ によって } M \text{ が正規化されることを示す}}$

準正規化定理により，$\xrightarrow{\text{name}} \circ \xrightarrow{\beta}\!\!\!\!\twoheadrightarrow$ による $M^{\pitchfork}$ の簡約は停止する（=導出に関する帰納法を使える）

$$
\begin{tikzcd}
M \arrow[r, two heads, "\mathrm{VCA}"] \arrow[d, dotted, "\pitchfork"'] & {} \arrow[r, "\mathrm{I}"] & {} \arrow[r, dashed, no head] & {} \arrow[r, two heads, "\mathrm{VCA}"] & {} \arrow[r, "\mathrm{I}"] & N \\
M^{\pitchfork} \arrow[r, "\mathrm{name}"] & {} \arrow[r, two heads, "\beta"] & {} \arrow[r, dashed, no head] & {} \arrow[r, "\mathrm{name}"] & {} \arrow[r, two heads, "\beta"] & {}
\end{tikzcd}
$$

完全性の証明 (1/2).

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

$\boxed{\text{まず} \xrightarrow{\text{need}} \text{によって } M \text{ が正規化されることを示す}}$

$\xrightarrow{\mathrm{I}}$ による簡約は $\xrightarrow{\text{name}} \circ \xrightarrow{\beta}\!\!\!\!\twoheadrightarrow$ による簡約に対応.

$$
\begin{tikzcd}
M \arrow[rr, "\mathrm{I}"] \arrow[d, dotted, "\pitchfork"'] & & \cdot \arrow[r, dashed, no head] \arrow[d, dotted, "\pitchfork"] & \cdot \arrow[rr, "\mathrm{I}"] \arrow[d, dotted, "\pitchfork"'] & & N \arrow[d, dotted, "\pitchfork"] \\
M^{\pitchfork} \arrow[r, "\text{name}"] & \cdot \arrow[r, two heads, "\beta"] & \cdot \arrow[r, dashed, no head] & \cdot \arrow[r, "\text{name}"] & \cdot \arrow[r, two heads, "\beta"] & \cdot
\end{tikzcd}
$$

完全性の証明 (1/2).

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

$\boxed{\text{まず } \xrightarrow{\text{need}} \text{ によって } M \text{ が正規化されることを示す}}$

$\xrightarrow{\text{VCA}}$ は正規化性を満たす

$$
\begin{tikzcd}
M \arrow[r, two heads, "\mathrm{VCA}"] \arrow[d, dotted, "\pitchfork"'] & {} \arrow[r, "\mathrm{I}"] & {} \arrow[r, dashed, dash] & {} \arrow[r, two heads, "\mathrm{VCA}"] & {} \arrow[r, "\mathrm{I}"] & N \\
M^{\pitchfork} \arrow[r, "\mathrm{name}"] & {} \arrow[r, two heads, "\beta"] & {} \arrow[r, dashed, dash] & {} \arrow[r, "\mathrm{name}"] & {} \arrow[r, two heads, "\beta"] & {}
\end{tikzcd}
$$

完全性の証明 (1/2).

$M^{\pitchfork} \xrightarrow{\mathrm{name}}\!\!\!\!\rightarrow V$ と仮定する.

$\boxed{\text{まず } \xrightarrow{\mathrm{need}} \text{ によって } M \text{ が正規化されることを示す}}$

$\xrightarrow{\mathrm{VCA}}$ は administrative な簡約

$$
\begin{tikzcd}
M \arrow[r, two heads, "\mathrm{VCA}"] \arrow[d, dotted, "\pitchfork"'] & \cdot \arrow[r, "\mathrm{I}"] \arrow[dl, dotted, bend right, "\pitchfork"] & \cdot \arrow[r, dash, dashed] \arrow[d, dotted, "\pitchfork"] & \cdot \arrow[r, two heads, "\mathrm{VCA}"] \arrow[d, dotted, "\pitchfork"'] & \cdot \arrow[r, "\mathrm{I}"] & N \arrow[dl, dotted, bend right, "\pitchfork"'] \arrow[d, dotted, "\pitchfork"] \\
M^{\pitchfork} \arrow[r, "\mathrm{name}"] & \cdot \arrow[r, two heads, "\beta"] & \cdot \arrow[r, dash, dashed] & \cdot \arrow[r, "\mathrm{name}"] & \cdot \arrow[r, two heads, "\beta"] & \cdot
\end{tikzcd}
$$

完全性の証明 (2/2).

$\boxed{\text{次に，} M \text{ の正規形 } N \text{ が answer であることを示す}}$

$$\begin{tikzcd} M \arrow[dr, twoheadrightarrow, "\text{need}"] & \\ & N \end{tikzcd}$$

□

完全性の証明 (2/2).

次に，M の正規形 N が answer であることを示す

正規形 N は answer または stuck 状態 $E[x]$ である

$$\begin{tikzcd} M \arrow[dr, "\text{need}"] & \\ & A \vee E[x] \end{tikzcd}$$

□

完全性の証明 (2/2).

$\boxed{\text{次に，} M \text{ の正規形 } N \text{ が answer であることを示す}}$

正規形 N が stuck 状態 $E[x]$ と仮定し，矛盾を導く．

$$\begin{tikzcd} M \arrow[dr, "\mathrm{need}"] & \\ & E[x] \end{tikzcd}$$

□

完全性の証明 (2/2).

$\boxed{\text{次に，} M \text{ の正規形 } N \text{ が answer であることを示す}}$

1 ステップの対応により $M^{\pitchfork} \overset{\beta}{\twoheadrightarrow} (E[x])^{\pitchfork}$

$$
\begin{tikzcd}
M^{\pitchfork} \arrow[dr, "\beta"'] & M \arrow[l, dotted, "\pitchfork"'] \arrow[dr, "\mathrm{need}"] & \\
& (E[x])^{\pitchfork} & E[x] \arrow[l, dotted, "\pitchfork"]
\end{tikzcd}
$$

$\square$

完全性の証明 (2/2).

$\boxed{\text{次に，} M \text{ の正規形 } N \text{ が answer であることを示す}}$

1 ステップの対応により $E[x]^{\pitchfork} = E_{\mathrm{n}}[x]$

$$
\begin{tikzcd}
M^{\pitchfork} \arrow[dr, "\beta"] & M \arrow[l, dotted, "\pitchfork"'] \arrow[dr, "\mathrm{need}"] & \\
& E_{\mathrm{n}}[x] & E[x] \arrow[l, dotted, "\pitchfork"]
\end{tikzcd}
$$

□

完全性の証明 (2/2).

$\boxed{\text{次に，} M \text{ の正規形 } N \text{ が answer であることを示す}}$

仮定より $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$

$$
\begin{tikzcd}
 & M^{\pitchfork} \arrow[dl, ->>, "\text{name}"'] \arrow[dr, ->>, "\beta"] & M \arrow[l, dotted, "\pitchfork"'] \arrow[dr, ->>, "\text{need}"] & \\
V & & E_{\text{n}}[x] & E[x] \arrow[l, dotted, "\pitchfork"]
\end{tikzcd}
$$

$\square$

完全性の証明 (2/2).

次に，M の正規形 N が answer であることを示す

$\xrightarrow{\beta}$ **の合流性により，** $V \overset{\beta}{\twoheadrightarrow} L$ **かつ** $E_{\mathrm{n}}[x] \overset{\beta}{\twoheadrightarrow} L$

となるような L **が存在する（**$\xrightarrow{\text{name}} \subset \xrightarrow{\beta}$**）**

$$
\begin{tikzcd}
 & M^{\pitchfork} \arrow[dl, "\text{name}"'] \arrow[dr, two heads, "\beta"] & M \arrow[l, dotted, "\pitchfork"'] \arrow[dr, two heads, "\text{need}"] & \\
V \arrow[dr, dashed, two heads, "\beta"'] & & E_{\mathrm{n}}[x] \arrow[dl, dashed, two heads, "\beta"] & E[x] \arrow[l, dotted, "\pitchfork"] \\
 & L & &
\end{tikzcd}
$$

$\square$

完全性の証明 (2/2).

$\boxed{\text{次に，} M \text{ の正規形 } N \text{ が answer であることを示す}}$

名前呼びにおける値であることや stuck 状態であることは，full-β 簡約で保存される

$$
\begin{tikzcd}
 & M^{\pitchfork} \arrow[dl, "\mathrm{name}"'] \arrow[dr, "\beta"] & M \arrow[l, dotted, "\pitchfork"'] \arrow[dr, "\mathrm{need}"] & \\
V \arrow[dr, dashed, "\beta"'] & & E_{\mathrm{n}}[x] \arrow[dl, dashed, "\beta"] & E[x] \arrow[l, dotted, "\pitchfork"] \\
 & L = V' = E'_{\mathrm{n}}[x] & &
\end{tikzcd}
$$

□

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. 貢献 1：主定理の簡潔化された証明
3. 貢献 2：必要呼びラムダ計算の形式的検証に適した定式化
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

$$\Downarrow$$

（一般的には）高階パターンマッチングが必要

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
  - $\xrightarrow{\beta}$, $\xrightarrow{\text{name}}$, $\xrightarrow{\text{I}}$ と $\xrightarrow{\text{VCA}}$
- stuck 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) $\mathbf{let}\ x = V\ \mathbf{in}\ E[x] \to \mathbf{let}\ x = V\ \mathbf{in}\ E[V]$
  を代入に簡略化

(元の定式化についても証明)

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.
```

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. 貢献 1：主定理の簡潔化された証明
3. 貢献 2：必要呼びラムダ計算の形式的検証に適した定式化
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
- 本研究の成果は FLOPS 2018 に投稿中
- TPP 2017 でも口頭発表