FLOPS 2018

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

Masayuki Mizuno and Eijiro Sumii

Tohoku University

May 9, 2018

# Motivation: a gap between abstraction and implementations of non-strict languages

- Call-by-name [Abramsky 1990 etc.]: (high-level) abstraction of non-strict languages
- Call-by-need [Wadsworth 1971 etc.]: implementations of non-strict languages

Our goal is mechanized verification of their correspondence

## Background 1: full-$\beta$ reduction

- Reduction is non-deterministic
  - $\underline{(\lambda xy.\ y)\ \Omega} \xrightarrow{\beta} \lambda y.\ y$
  - $(\lambda xy.\ y)\ \underline{\Omega} \xrightarrow{\beta} (\lambda xy.\ y)\ \Omega$

$$\Omega = (\lambda x.\ xx)\ (\lambda x.\ xx)$$

# Background 2: call-by-name

**Definition (call-by-name)**

$$E_{\mathrm{n}} ::= [\,] \mid E_{\mathrm{n}}\ M$$

$$E_{\mathrm{n}}[(\lambda x.M)N] \xrightarrow{\text{name}} E_{\mathrm{n}}[M[x \mapsto N]]$$

- If $M \xrightarrow{\beta}\!\!\!\!\rightarrow \lambda x.N$, then $M \xrightarrow{\text{name}}\!\!\!\!\rightarrow \lambda x.N'$

$$(\lambda xy.\ y)\ \Omega \xrightarrow{\text{name}}\!\!\!\!\rightarrow \lambda y.\ y$$

# Problem: Redundant reductions

$$
\begin{aligned}
&(\lambda x.\ xx)\ \underline{(I\ I)} \\
&\xrightarrow{\beta} \underline{(\lambda x.\ xx)\ I} \\
&\xrightarrow{\beta} \underline{I\ I} \\
&\xrightarrow{\beta} I
\end{aligned}
$$

$$
\begin{aligned}
&\underline{(\lambda x.\ xx)\ (I\ I)} \\
&\xrightarrow{\text{name}} \underline{I\ I}\ (I\ I) \\
&\xrightarrow{\text{name}} \underline{I\ (I\ I)} \\
&\xrightarrow{\text{name}} \underline{I\ I} \\
&\xrightarrow{\text{name}} I
\end{aligned}
$$

$$I = \lambda x.\ x$$

# Background 3: call-by-need

- Reuse evaluation

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

- Should correspond with call-by-name

# Our contributions

- Formalization of call-by-need $\lambda$-calculus [Ariola+ 1995] in the Coq proof assistant
- Simplified proof of correspondence with call-by-name, and verification in Coq
    - using standardization theorem [Curry&Feys 1958]

# Outline

1. Call-by-name and call-by-need $\lambda$-calculi
2. Simplified proof of the correspondence
3. Coq formalization
4. Conclusion

# Outline

1. Call-by-name and call-by-need $\lambda$-calculi
2. Simplified proof of the correspondence
3. Coq formalization
4. Conclusion

# Call-by-name $\lambda$-calculus

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

$$\frac{M \to N}{E_\mathrm{n}[M] \xrightarrow{\text{name}} E_\mathrm{n}[N]}$$

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

- Reduction is deterministic
- All stuck states are of the form $E_\mathrm{n}[x]$

## Lemma (determinacy of call-by-name reduction)

- $\xrightarrow{\text{name}}$ is partial function
- If $E_{\mathrm{n}}[x] = E'_{\mathrm{n}}[y]$ then $x = y$
- For any term $M$, exactly one of the following holds:
  1. $M$ is a value
  2. $M = E_{\mathrm{n}}[x]$ for some $E_{\mathrm{n}}$ and $x$
  3. $M \xrightarrow{\text{name}} N$ for some $N$

# Standardization theorem [Curry&Feys 1958]

**Definition (standard reduction sequence)**

A reduction sequence
$M_1 \xrightarrow[\Delta_1]{\beta} M_2 \xrightarrow[\Delta_2]{\beta} \cdots \xrightarrow[\Delta_{n-1}]{\beta} M_n$ is *standard* if
every $\Delta_i$ is outer and lefter than $\Delta_{i+1}$

**Theorem (standardization)**

If $M \overset{\beta}{\twoheadrightarrow} N$, then there is a standard reduction sequence from $M$ to $N$

## Corollaries

**Corollary (termination of $\xrightarrow{\text{name}}$)**

If $M \overset{\beta}{\twoheadrightarrow} V$ then, $M \overset{\text{name}}{\twoheadrightarrow} V'$ for some $N$

**Corollary (termination of $\xrightarrow{\text{name}} \circ \overset{\beta}{\twoheadrightarrow}$)**

If $M \overset{\beta}{\twoheadrightarrow} V$, then $M$ is terminating by $\xrightarrow{\text{name}} \circ \overset{\beta}{\twoheadrightarrow}$

- Used for our proof of the correpondence with call-by-need

# Call-by-need $\lambda$-calculus [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 | $A$ | $::=$ | $V \mid \mathbf{let}\ x = M\ \mathbf{in}\ A$ |
| Evalctx | $E, E'$ | $::=$ | $[] \mid E\ M \mid \mathbf{let}\ x = M\ \mathbf{in}\ E$ |
| | | $\mid$ | $\mathbf{let}\ x = E\ \mathbf{in}\ E'[x]$ |

$$(I) \quad (\lambda x.M)N \to \mathbf{let}\ x = N\ \mathbf{in}\ M$$

$$(V) \quad \mathbf{let}\ x = V\ \mathbf{in}\ E[x] \to \mathbf{let}\ x = V\ \mathbf{in}\ E[V]$$

$$(C) \quad (\mathbf{let}\ x = M\ \mathbf{in}\ A)\ N \to \mathbf{let}\ x = M\ \mathbf{in}\ A\ N$$

$$(A) \quad \mathbf{let}\ y = (\mathbf{let}\ x = M\ \mathbf{in}\ A)\ \mathbf{in}\ E[y] \to \mathbf{let}\ x = M\ \mathbf{in}\ \mathbf{let}\ y = A\ \mathbf{in}\ E[y]$$

$\xrightarrow{\mathrm{I}}$ reduction only using (I)

$\xrightarrow{\mathrm{VCA}}$ reduction only using (V), (C) and (A) (administrative)

## Lemma (determinacy of call-by-need reduction)

- $\xrightarrow{\mathrm{I}}$ is a partial function
- $\xrightarrow{\mathrm{VCA}}$ is a partial function
- If $E[x] = E'[y]$ then $x = y$
- For any term $M$, exactly one of the following holds:
  1. $M$ is an answer
  2. $M = E[x]$ for some $E$ and $x$
  3. $M \xrightarrow{\mathrm{I}} N$ for some $N$
  4. $M \xrightarrow{\mathrm{VCA}} N$ for some $N$

# Outline

## Definition (correspondence of terms)

$\pitchfork$: call-by-need terms $\longrightarrow$ call-by-name terms

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

- Expands **let**
- Equivalent to [Maraist+ 1998] (except "marked redexes")

# Main theorem: correspondence of call-by-need with call-by-name

**Theorem (soundness of $\xrightarrow{\text{need}}$)**

If $M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$, then $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$ for some $V$

**Theorem (completeness of $\xrightarrow{\text{need}}$)**

If $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$, then $M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$ for some $A$

(Correspondence between $A$ and $V$ also holds)

# Cf. previous researches

- Ariola and Felleisen [1997]
  - Based on informally defined term graphs and their correspondence
- Maraist et al. [1998]
  - Complicated "marked reduction" and explicit treatment of reduction position

We give a simpler proof!

# Our proof

**Lemma (single-step correspondence)**

- $A^{\pitchfork}$ is a value
- For any $E$ and $x$, there exists $E_{\mathrm{n}}$ such that $E[x]^{\pitchfork} = E_{\mathrm{n}}[x]$
- If $M \xrightarrow{\mathrm{VCA}} N$ then $M^{\pitchfork} = N^{\pitchfork}$
- If $M \xrightarrow{\mathrm{I}} N$ then $M^{\pitchfork} \xrightarrow{\mathrm{name}} \circ \overset{\beta}{\twoheadrightarrow} N^{\pitchfork}$

# Proof of soundness

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

Since $A$ is an answer, $A^{\pitchfork}$ is a value
Hence $M^{\pitchfork} \overset{\mathrm{name}}{\twoheadrightarrow} V$ by the termination of $\xrightarrow{\mathrm{name}}$ □

# Completeness

If $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$ then
$M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$

Difficulties:

- Administrative reductions might not terminate
  - If $M \xrightarrow{\text{VCA}} N$ then $M^{\pitchfork} = N^{\pitchfork}$
- Redexes shared by **let** are reduced at once
  - $M \xrightarrow{\text{I}} N$ implies $M^{\pitchfork} \xrightarrow{\text{name}} \circ \xrightarrow{\beta}\!\!\!\!\rightarrow N^{\pitchfork}$
    - Bodies of $\lambda$-abstraction can be reduced

### Lemma (normalization of $\xrightarrow{\text{VCA}}$)

By a variant of [Maraist+ 1998]'s weighting:

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

$M \xrightarrow{\text{VCA}} N$ implies $\| M \|_s > \| N \|_s$

## Proof (completeness of call-by-need) (1/2).

Assume $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$, show $M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$

First, we show call-by-need reduction of $M$ is normalizing

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

## Proof (completeness of call-by-need) (1/2).

Assume $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$, show $M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$

First, we show call-by-need reduction of $M$ is normalizing

$\xrightarrow{\text{need}}\!\!\!\!\rightarrow = (\xrightarrow{\text{I}} \cup \xrightarrow{\text{VCA}})^* = (\xrightarrow{\text{VCA}}\!\!\!\!\rightarrow \circ \xrightarrow{\text{I}})^*$ holds

$$M \xrightarrow{\text{VCA}}\!\!\!\!\rightarrow \xrightarrow{\text{I}} \cdots \xrightarrow{\text{VCA}}\!\!\!\!\rightarrow \xrightarrow{\text{I}}$$

## Proof (completeness of call-by-need) (1/2).

Assume $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$, show $M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$

First, we show call-by-need reduction of $M$ is normalizing

$\xrightarrow{\text{VCA}}$ is an administrative reduction

$\xrightarrow{\text{I}}$ corresponds $\xrightarrow{\text{name}} \circ \xrightarrow{\beta}\!\!\!\!\rightarrow$

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

# Proof (completeness of call-by-need) (1/2).

Assume $M^{\pitchfork} \xrightarrow{\text{name}}\!\!\!\!\rightarrow V$, show $M \xrightarrow{\text{need}}\!\!\!\!\rightarrow A$

First, we show call-by-need reduction of $M$ is normalizing

By $M^{\pitchfork} \xrightarrow{\beta}\!\!\!\!\rightarrow V$, $M^{\pitchfork}$ is terminating by $\xrightarrow{\text{name}} \circ \xrightarrow{\beta}\!\!\!\!\rightarrow$ (= induction on derivation is available)

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

## Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

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

## Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

Normal form $N$ is an answer or stuck state $E[x]$

$$M \xrightarrow{\text{need}} N = A \vee N = E[x]$$

## Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

Assume $N$ is stuck state, show it leads to contradiction

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

## Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

By single-step correspondence:

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

# Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

By single-step correspondence:

$$\begin{tikzcd}
M^{\pitchfork} \arrow[dr, ->>, "\beta"] & M \arrow[l, dotted, "\pitchfork"'] \arrow[dr, ->>, "\text{need}"] & \\
& \color{red}{E_{\mathrm{n}}[x]} & N = E[x] \arrow[l, dotted, "\pitchfork"]
\end{tikzcd}$$

# Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

By assumption:

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

# Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

By confluence of $\xrightarrow{\beta}$:

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

## Proof (completeness of call-by-need) (2/2).

Next, show normal form $N$ of $M$ is an answer

$\xrightarrow{\beta}$ preserves valueness and stuckness in call-by-name

□

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

# Outline

# Coq formalization

Almost straightforward, expect treatment of evaluation contexts

```
Lemma answer_or_stuck_or_reducible M :
  answer M
  ∨ (∃ E x,
    evalctx E ∧ M = E.[tvar x] ∧ bv E ≦ x)
  ∨ (∃ E L N,
    evalctx E ∧ M = E.[L] ∧ reduceI L N)
  ∨ (∃ E L N,
    evalctx E ∧ M = E.[L] ∧ reduceVCA L N).
```

- Try induction on $M$

# Case $M = x$

```
x : var
============================
answer (tvar x)
∨ (∃ E y, evalctx E ∧ tvar x = E.[tvar y]
  ∧ bv E ≦ x)
∨ (∃ E L N, evalctx E ∧ tvar x = E.[L]
  ∧ reduceI L N)
∨ (∃ E L N, evalctx E ∧ tvar x = E.[L]
  ∧ reduceVCA L N).
```

- Trivial from $E = []$

## Case $M = x$

- However, automated reasoning fails

```
Coq < eauto.

x : var
============================
answer (tvar x)
∨ (∃ E y, evalctx E ∧ tvar x = E.[tvar y]
  ∧ bv E ≦ x)
∨ (∃ E L N, evalctx E ∧ tvar x = E.[L]
  ∧ reduceI L N)
∨ (∃ E L N, evalctx E ∧ tvar x = E.[L]
  ∧ reduceVCA L N).
```

# Why fails?

To prove...

```
∃ E y, evalctx E ∧
  tvar x = E.[tvar y] ∧ bv E ≦ x
```

...we must find $E$ such that

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

$$\Downarrow$$

Higher order pattern matching required!

# Solution: eliminate evaluation contexts

- Expand evaluation contexts into reductions
  - $\xrightarrow{\beta}$, $\xrightarrow{\text{name}}$, $\xrightarrow{\text{I}}$ and $\xrightarrow{\text{VCA}}$
- Introduce stuckness predicate
  - $\mathbf{needs}_{\text{n}}(M, x)$ $(\Leftrightarrow \exists E.M = E_{\text{n}}[x])$ and $\mathbf{needs}(M, x)$ $(\Leftrightarrow \exists E.M = E[x])$
- Approximate
  $\mathbf{let}\ x = V\ \mathbf{in}\ E[x] \to \mathbf{let}\ x = V\ \mathbf{in}\ E[V]$
  by substitution
  $\mathbf{let}\ x = V\ \mathbf{in}\ E[x] \to E[x][x \mapsto V]$

  (N.B. correspondence in original semantics is also proved)

# Automation succeeds!

```
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.
```

# Outline

## Conclusion

- Formalized call-by-need $\lambda$-calculus [Ariola$+$ 1995] in the Coq proof assistant
- Gave simplified proof of correspondence with call-by-name, and verified in Coq
  - Using standardization theorem [Curry&Feys 1958]