Yuki Honda
Koji Nakazawa
Ken-etsu Fujita

Confluence Proofs of Lambda-Mu-Calculi by Z Theorem

Abstract. This paper applies Dehornoy et al.'s Z theorem and its variant, called the compositional Z theorem, to prove confluence of Parigot's $\lambda\mu$-calculi extended by the simplification rules. First, it is proved that Baba et al.'s modified complete developments for the call-by-name and the call-by-value variants of the $\lambda\mu$-calculus with the renaming rule, which is one of the simplification rules, satisfy the Z property. It gives new confluence proofs for them by the Z theorem. Secondly, it is shown that the compositional Z theorem can be applied to prove confluence of the call-by-name and the call-by-value $\lambda\mu$-calculi with both simplification rules, the renaming and the $\mu\eta$-rules, whereas it is hard to apply the ordinary parallel reduction technique or the original Z theorem by one-pass definition of mappings for these variants.

Keywords: lambda calculus, lambda-mu calculus, confluence, permutative conversion.

1. Introduction

The $\lambda\mu$-calculus is proposed by Parigot [8] as an extension of the λ-calculus corresponding to a classical variant of natural deduction. The $\lambda\mu$-calculus can be also seen as an abstract computational model of programming languages with control operators such as exception handling and first-class continuations [7].

The confluence is one of the most important properties of reduction systems. In general, it is hard to prove confluence of variants of the untyped λ-calculi, since it satisfies neither the strong normalization nor the diamond property. To prove confluence of the λ-calculi, some elaborated techniques such as the parallel reduction and the complete development are required. The Z theorem [2] is another tool for confluence, and it says that if there exists a mapping satisfying the Z property, then we can conclude confluence of the system.

The Z theorem is useful to prove confluence of extensions of λ-calculi. For the λ-calculus with the β-reduction, the complete development satisfies the Z property, and it gives a simple proof of confluence of the β-reduction. This

Presented by ; *Received*

technique can be extended to the $\lambda\mu$-calculus (without any simplification rule) [6].

Baba et al. [1] considered variants of $\lambda\mu$-calculi extended by the renaming rule, which was introduced by Parigot [8] as one of the simplification rules. They showed that the standard parallel reduction does not work for such extensions, and modified the parallel reduction and the complete development to prove confluence of the $\lambda\mu$-calculi with the renaming rule. However it seems hard to extend their parallel reduction to cover another simplification rule in [8], called the $\mu\eta$-rule. This rule is also required when we consider computational aspect of the $\lambda\mu$-calculus [7]. Nakazawa [5] proved confluence of the call-by-value $\lambda\mu$-calculus with the polymorphic types and the both simplification rules, that is, renaming and $\mu\eta$. However, confluence of the variants of $\lambda\mu$-calculus with both simplification rules is not well studied.

This paper shows that we can apply the Z theorem to prove confluence of the $\lambda\mu$-calculi with the simplification rules. First, it is proved that the modified variants of the complete developments in [1] for the call-by-name and the call-by-value variants of the $\lambda\mu$-calculus with renaming rule satisfy the Z property. Secondly, it is shown that a variant of the compositional Z theorem, called the compositional Z [6], can be applied to prove confluence of the call-by-name and the call-by-value $\lambda\mu$-calculi with both simplification rules, the renaming and the $\mu\eta$-rules, whereas it is hard to apply the ordinary parallel reduction technique or the original Z theorem by one-pass definition of mappings for these variants.

2. Confluence and Z theorem

In this section, we introduce the Z theorem [2] and the compositional Z theorem [6], which are general techniques to prove confluence.

Definition 2.1 (Confluence)**.** Let $(A, \to)$ be an abstract rewriting system, that is, A is a set and $\to$ is a binary relation on A, and $\twoheadrightarrow$ be the reflexive transitive closure of $\to$. $(A, \to)$ is said to be confluent if and only if, for any $x, y_1, y_2 \in A$, $y_1 \twoheadleftarrow x \twoheadrightarrow y_2$ implies that there is some $z \in A$ such that $y_1 \twoheadrightarrow z \twoheadleftarrow y_2$.

Definition 2.2 ((Weak) Z property [2])**.** Let A be a set, $\to$ and $\to_{\mathsf{x}}$ be two binary relations, and $\twoheadrightarrow$ and $\twoheadrightarrow_{\mathsf{x}}$ be reflexive transitive closures of $\to$ and $\to_{\mathsf{x}}$, respectively.

1. A mapping $f : A \to A$ satisfies the weak Z property for $\to$ by $\to_{\mathsf{x}}$ if $a \to b$ implies $b \twoheadrightarrow_{\mathsf{x}} f(a) \twoheadrightarrow_{\mathsf{x}} f(b)$ for any $a, b \in A$.

2. A mapping $f : A \to A$ satisfies the Z property for $\to$ if it satisfies the weak Z property by $\to$ itself.

It becomes clear why we call it the Z property when we draw the condition as Figure 1.

$$\begin{tikzcd}
a \arrow[r] & b \arrow[dl, dashed, twoheadrightarrow] \\
f(a) \arrow[r, dashed, twoheadrightarrow] & f(b)
\end{tikzcd}$$

Figure 1. Z property

Theorem 2.3 (Z theorem [2]). *If there exists a mapping satisfying the Z property for an abstract rewriting system, then it is confluent.*

One example of a mapping satisfying the Z property is the complete development for the β-reduction of the λ-calculus.

Definition 2.4 (λ-calculus). The λ-terms are defined by

$$M ::= x \mid \lambda x.M \mid MM.$$

The bound and the free variables in M are defined as usual, and we identify α-equivalent terms. $FV(M)$ is defined as the set of the free variables in M. The β-reduction on the λ-terms is defined as the compatible closure of the following β-rule

$$(\lambda x.M)N \to M[x := N] \qquad (\beta),$$

where $M[x := N]$ is the capture-avoiding substitution which is defined as usual.

A subterm of a term which matches a left-hand side of the β-rule is called a redex.

The complete development is the mapping reducing all visible redexes at once.

Definition 2.5 (Complete development for β-reduction). The complete development M^* for the β-reduction is defined as follows.

1. $x^* = x$
2. $(\lambda x.M)^* = \lambda x.M^*$
3. $((\lambda x.M_1)M_2)^* = M_1^*[x := M_2^*]$
4. $(M_1 M_2)^* = M_1^* M_2^*$ (otherwise)

Theorem 2.6 ([2])**.** *The mapping M^* satisfies the Z property.*

The Z theorem is widely applicable to prove confluence for rewriting systems including (first-order) term rewriting systems and some variants of λ-calculus. However, Nakazawa and Fujita [6] showed that it is hard to apply it to λ-calculi extended by permutation-like reduction rules. In order to prove confluence of such calculi, they proposed the compositional Z theorem, by which we can consider the mapping for the Z theorem by dividing the reduction relation into two or more groups.

Theorem 2.7 (Compositional Z [6])**.** *Let $(A, \to)$ be an abstract rewriting system, and $\to = \to_1 \cup \to_2$. If there exist mappings $f_1, f_2 : A \to A$ such that*

1. *f_1 satisfies the Z property for $\to_1$,*
2. *$a \to_1 b$ implies $f_2(a) \twoheadrightarrow f_2(b)$,*
3. *$a \twoheadrightarrow f_2(a)$ holds for any $a \in Im(f_1)$,*
4. *$f_2 \circ f_1$ satisfies the weak Z property for $\to_2$ by $\to$,*

then $f_2 \circ f_1$ satisfies the Z property for $(A, \to)$, and hence $(A, \to)$ is confluent.

Figure 2 depicts the conditions of the compositional Z theorem.

In [6], they showed that Theorem 2.7 can be used to prove confluence of the λ- and $\lambda\mu$-calculi extended with permutative conversion rules, whereas it seems hard to give a mapping satisfying the Z property by a one-pass definition. By the compositional Z theorem, we can consider the mappings by dividing the reduction into two types, the permutative conversion and the β-reduction.

3. Confluence proof of $\lambda\mu$-calculi with renaming by Z

3.1. Call-by-name $\lambda\mu$-calculus

The call-by-name $\lambda\mu$-calculus is defined as follows.

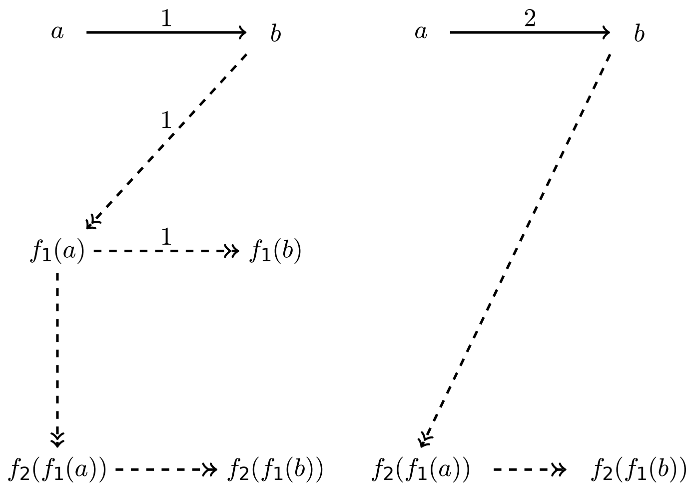

Figure 2. Compositional Z

Definition 3.1 (Call-by-name $\lambda\mu$-calculus)**.** We assume that we have another sort of variables called μ-variables and denoted by α, β,.... The $\lambda\mu$-terms are defined as

$$M ::= x \mid \lambda x.M \mid MM \mid \mu\alpha.M \mid [\alpha]M.$$

We consider that variable occurrences of α in $\mu\alpha.M$ are bound. The reduction rules are

$$\begin{aligned} &(\lambda x.M)N \to M[x := N] && (\beta) \\ &(\mu\alpha.M)N \to \mu\alpha.M[[\alpha]w := [\alpha](wN)] && (\mathrm{Str}) \\ &[\beta](\mu\alpha.M) \to M[\alpha := \beta] && (\mathrm{Ren}), \end{aligned}$$

where $M[[\alpha]w := [\beta](wN)]$ is called structural substitution and defined as follows.

1. $x[[\alpha]w := [\beta](wN)] = x$
2. $(\lambda x.M)[[\alpha]w := [\beta](wN)] = \lambda x.M[[\alpha]w := [\beta](wN)] \qquad (x \notin FV(N))$
3. $(M_1M_2)[[\alpha]w := [\beta](wN)] = M_1[[\alpha]w := [\beta](wN)]M_2[[\alpha]w := [\beta](wN)]$
4. $(\mu\gamma.M)[[\alpha]w := [\beta](wN)] = \mu\gamma.M[[\alpha]w := [\beta](wN)]$ $\qquad (\gamma \notin FV(N) \cup \{\alpha, \beta\})$
5. $([\alpha]M)[[\alpha]w := [\beta](wN)] = [\beta](M[[\alpha]w := [\beta](wN)]N)$

6. $([\gamma]M)[[\alpha]w := [\beta](wN)] = [\gamma]M[[\alpha]w := [\beta](wN)] \qquad (\gamma \neq \alpha)$

The reduction $\to$ of the call-by-name $\lambda\mu$-calculus is defined as the compatible closure of the reduction rules.

We call a term of the form $[\alpha]((\mu\beta.M)N_1 \cdots N_n)$ an extended renaming redex.

If we remove the rule (Ren), we can apply the standard parallel-reduction technique or the original Z theorem to prove confluence. However, as Baba et al. pointed out in [1], for the call-by-name $\lambda\mu$-calculus with the (Ren) rule, the problem arises in the case where a (Ren) redex and a (Str) redex overlap. Consider the term $M = (\mu\alpha.[\alpha](\mu\beta.[\alpha]x))y$. We have $M \to (\mu\alpha.[\alpha]x)y = N_1$ by (Ren) and $M \to \mu\alpha.[\alpha]((\mu\beta.[\alpha](xy))y) = N_2$ by (Str). If we consider the standard complete development, we have $M^* = N_1^* = \mu\alpha.[\alpha](xy)$ whereas $N_2^* = \mu\alpha.[\alpha](\mu\beta.[\alpha]xy)$ since the redex of (Ren) does not exist in N_2, and then we have $M^* \not\to^* N_2^*$. Hence the mapping $(\cdot)^*$ does not satisfy the Z property.

To solve this problem, Baba et al. [1] modified the parallel reduction and the complete development, where every extended renaming redex is reduced at once by consecutive (Str) steps followed by a (Ren) step.

Definition 3.2 ([1])**.** The modified complete development M^* of the call-by-name $\lambda\mu$ calculus is defined as follows.

1. $x^* = x$
2. $(\lambda x.M)^* = \lambda x.M^*$
3. $((\lambda x.M_1)M_2)^* = M_1^*[x := M_2^*]$
4. $((\mu\alpha.M_1)M_2)^* = \mu\alpha.M_1^*[[\alpha]w := [\alpha](wM_2^*)]$
5. $(M_1M_2)^* = M_1^*M_2^* \qquad$ (otherwise)
6. $(\mu\alpha.M)^* = \mu\alpha.M^*$
7. $([\beta](\mu\alpha.M))^* = M^*[\alpha := \beta]$
8. $([\alpha]((\mu\beta.M)N_1 \ldots N_n))^* = M^*[[\beta]w := [\alpha](wN_1^* \ldots N_n^*)]$
9. $([\alpha]M)^* = [\alpha]M^* \qquad$ (otherwise)

In [1], they showed that the modified variant of the parallel reduction satisfies the diamond property by using the modified variant of the complete development defined above.

3.2. Confluence proof of call-by-name $\lambda\mu$-calculus by Z theorem

We prove that Baba et al.'s mapping M^* satisfies the Z property. This gives a new proof of confluence of the call-by-name $\lambda\mu$-calculus with the renaming rule. We do not need a parallel reduction in this proof.

Lemma 3.3. *$M \twoheadrightarrow M^*$.*

Proof. It is proved by straightforward induction on the structure of M. □

Lemma 3.4. *(1) $M^*[x := N^*] \twoheadrightarrow (M[x := N])^*$.*
(2) $M^*[[\alpha]w := [\alpha](wN^*)] \twoheadrightarrow (M[[\alpha]w := [\alpha](wN)])^*$.

Proof. They are proved by induction on the structure of M. We show only non-trivial cases, and the other cases are proved in a straightforward way.

(1) (Case $M = xM_1$ and $N = \lambda y.L$) It should be noted that, in this case, a β-redex $(\lambda y.L)M_1$ is created by the substitution $M[x := N]$.

$$\begin{aligned}
M^*[x := N^*] &= (xM_1^*)[x := \lambda y.L^*]) \\
&= (\lambda y.L^*)(M_1^*[x := \lambda y.L^*]) \\
&\to L^*[y := M_1^*[x := \lambda y.L^*]] \\
&\twoheadrightarrow L^*[y := (M_1[x := \lambda y.L])^*] && \text{(I.H.)} \\
&= ((\lambda y.L)(M_1[x := \lambda y.L]))^* \\
&= (M[x := N])^*
\end{aligned}$$

(2) (Case $M = [\alpha](\lambda x.M_1)$) In this case, a β-redex $(\lambda x.M_1)N$ is created by the structural substitution $M[[\alpha]w := [\alpha](wN)]$.

$$\begin{aligned}
M^*[[\alpha]w := [\alpha](wN^*)] &= [\alpha]((\lambda x.M_1^*[[\alpha]w := [\alpha](wN^*)])N^*) \\
&\to [\alpha]((M_1^*[[\alpha]w := [\alpha](wN^*)])[x := N^*]) \\
&\twoheadrightarrow [\alpha]((M_1[[\alpha]w := [\alpha](wN)])^*[x := N^*]) && \text{(I.H.)} \\
&= [\alpha]((\lambda x.M_1[[\alpha]w := [\alpha](wN)])N)^* \\
&= (M[[\alpha]w := [\alpha](wN)])^*
\end{aligned}$$

(Case $M = [\alpha]((\mu\beta.M_1)N_1 \dots N_n)$) In this case, an extended renaming redex is expanded (or created) by the structural substitution $M[[\alpha]w := [\alpha](wN)]$. In the following, θ and θ^* denote the structural substitutions

$M[[\alpha]w := [\alpha](wN)]$ and $M[[\alpha]w := [\alpha](wN^*)]$, respectively.

$$\begin{aligned}
M^*\theta^* &= (M_1^*[[\beta]w := [\alpha](wN_1^* \dots N_n^*)])\theta^* \\
&= (M_1^*\theta^*)[[\beta]w := [\alpha](wN_1^*\theta^* \dots N_n^*\theta^* N^*)] \\
&\twoheadrightarrow (M_1\theta)^*[[\beta]w := [\alpha](w(N_1\theta)^* \dots (N_n\theta)^* N^*)] && \text{(I.H.)} \\
&= ([\alpha]((\mu\beta.M_1\theta)N_1\theta \dots N_n\theta N))^* \\
&= (M\theta)^*
\end{aligned}$$

□

Proposition 3.5. *$M \to N$ implies $N \twoheadrightarrow M^*$.*

Proof. It is proved by induction on the definition of $M \to N$.

(Case β) $M = (\lambda x.M_1)N_1 \to M_1[x := N_1] = N$.

$$\begin{aligned}
N &= M_1[x := N_1] \\
&\twoheadrightarrow M_1^*[x := N_1^*] && \text{(by Lemma 3.3)} \\
&= M^*
\end{aligned}$$

(Case Str) $M = (\mu\alpha.M_1)N_1 \to \mu\alpha.M_1[[\alpha]w := [\alpha](wN_1)] = N$.

$$\begin{aligned}
N &= \mu\alpha.M_1[[\alpha]w := [\alpha](wN_1)] \\
&\twoheadrightarrow \mu\alpha.M_1^*[[\alpha]w := [\alpha](wN_1^*)] && \text{(by Lemma 3.3)} \\
&= M^*
\end{aligned}$$

(Case Ren) $M = [\beta](\mu\alpha.M_1) \to M_1[\alpha := \beta] = N$.

$$\begin{aligned}
N &= M_1[\alpha := \beta] \\
&\twoheadrightarrow M_1^*[\alpha := \beta] && \text{(by Lemma 3.3)} \\
&= M^*
\end{aligned}$$

(Case $M = M_1N_1 \to M_2N_1 = N$, where $M_1 \to M_2$)

$$\begin{aligned}
N &= M_2N_1 \\
&\twoheadrightarrow (M_1^*)N_1 && \text{(I.H.)} \\
&\twoheadrightarrow (M_1^*)N_1^* && \text{(by Lemma 3.3)} \\
&= (xN_1^*)[x := M_1^*] \\
&= (xN_1)^*[x := M_1^*] \\
&\twoheadrightarrow ((xN_1)[x := M_1])^* && \text{(by Lemma 3.4 (1))} \\
&= (M_1N_1)^* \\
&= M^*
\end{aligned}$$

The other cases are similarly proved. □

Proposition 3.6. *$M \to N$ implies $M^* \twoheadrightarrow N^*$.*

Proof. It is proved by induction on the definition of $M \to N$.
(Case β) $M = (\lambda x.M_1)N_1 \to M_1[x := N_1] = N$.

$$\begin{aligned} M^* &= M_1^*[x := N_1^*] \\ &\twoheadrightarrow (M_1[x := N_1])^* && \text{(by Lemma 3.4 (1))} \\ &= N^* \end{aligned}$$

(Case Str) $M = (\mu\alpha.M_1)N_1 \to \mu\alpha.M_1[[\alpha]w := [\alpha](wN_1)] = N$.

$$\begin{aligned} M^* &= \mu\alpha.M_1^*[[\alpha]w := [\alpha](wN_1^*)] \\ &\twoheadrightarrow \mu\alpha.(M_1[[\alpha]w := [\alpha](wN_1)])^* && \text{(by Lemma 3.4 (2))} \\ &= N^* \end{aligned}$$

(Case Ren) $M = [\beta](\mu\alpha.M_1) \to M_1[\alpha := \beta] = N$.

$$\begin{aligned} M^* &= M_1^*[\alpha := \beta] \\ &= N^* \end{aligned}$$

The other cases are proved by induction hypotheses. □

Theorem 3.7. *The mapping M^* for the call-by-name $\lambda\mu$-calculus satisfies the Z property.*

Proof. This holds by Proposition 3.5 and Proposition 3.6. □

By the Z theorem, the confluence of the call-by-name $\lambda\mu$-calculus with renaming rule follows from Theorem 3.7

3.3. Call-by-value $\lambda\mu$-calculus

The following call-by-value $\lambda\mu$-calculus is introduced in [3, 4].

Definition 3.8 (Call-by-value $\lambda\mu$-calculus)**.** The terms are the same as Definition 3.1. The values are defined as

$$V ::= x \mid \lambda x.M \mid [\alpha]M.$$

The reduction rules are the following.

$$\begin{aligned} &(\lambda x.M)V \to M[x := V] && (\beta) \\ &(\mu\alpha.M)N \to \mu\alpha.M[[\alpha]w := [\alpha](wN)] && (\mathrm{Str}) \\ &V(\mu\alpha.M) \to \mu\alpha.M[[\alpha]w := [\alpha](Vw)] && (\mathrm{Sym}) \\ &[\beta](\mu\alpha.V) \to V[\alpha := \beta] && (\mathrm{Ren}) \end{aligned}$$

The reduction of the call-by-value $\lambda\mu$-calculus $\to$ is defined as the compatible closure of the reduction rules.

The call-by-value contexts are defined as

$$\mathcal{E} ::= [\ ] \mid \mathcal{E}M \mid V\mathcal{E}.$$

Every call-by-value context $\mathcal{E}$ can be expressed as $\mathcal{E} = \mathcal{E}_n[\ldots\mathcal{E}_1[\ ]\ldots]$ where each $\mathcal{E}_i$ is either $V[\ ]$ or $[\ ]M$.

As with the call-by-name case, (Ren) causes a problem, and Baba et al. [1] also gave the modified complete development.

Definition 3.9 ([1])**.** The modified complete development M^* of the call-by-value $\lambda\mu$-calculus is defined as follows.

1. $x^* = x$
2. $(\lambda x.M)^* = \lambda x.M^*$
3. $((\lambda x.M_1)V_2)^* = M_1^*[x := V_2^*]$
4. $((\mu\alpha.M_1)M_2)^* = \mu\alpha.M_1^*[[\alpha]w := [\alpha](wM_2^*)]$
5. $(V_1(\mu\alpha.M_2))^* = \mu\alpha.M_2^*[[\alpha]w := [\alpha](V_1^*w)]$
6. $(M_1M_2)^* = M_1^*M_2^*$ (otherwise)
7. $(\mu\alpha.M)^* = \mu\alpha.M^*$
8. $([\alpha](\mathcal{E}[\mu\beta.V]))^* = V^*[[\beta]w := [\alpha](\mathcal{E}^*[w])]$
9. $([\alpha]M)^* = [\alpha]M^*$ (otherwise),

where $\mathcal{E}^*[\ ]$ is defined as

$$[\ ]^* = [\ ] \qquad (\mathcal{E}M)^* = \mathcal{E}^*M^* \qquad (V\mathcal{E})^* = V^*\mathcal{E}^*.$$

This mapping satisfies the Z theorem. It is proved in a similar way to the call-by-name case.

Lemma 3.10. *(1)* $M \twoheadrightarrow M^*$.
(2) $M^*[x := V^*] \twoheadrightarrow (M[x := V])^*$.
(3) $M^*[[\alpha]w := [\alpha](wN^*)] \twoheadrightarrow (M[[\alpha]w := [\alpha](wN)])^*$.
(4) $M^*[[\alpha]w := [\alpha](V^*w)] \twoheadrightarrow (M[[\alpha]w := [\alpha](Vw)])^*$.

Proof. They are proved by induction on the structure of M. □

Proposition 3.11. *(1)* $M \to N$ *implies* $N \twoheadrightarrow M^*$.
(2) $M \to N$ *implies* $M^* \twoheadrightarrow N^*$.

Proof. The proof is almost the same as those of Proposition 3.5 and 3.6. $\square$

Theorem 3.12. *The mapping M^* for the call-by-value $\lambda\mu$-calculus satisfies the Z property.*

Proof. This holds by Proposition 3.11. $\square$

By the Z theorem, the confluence of the call-by-value $\lambda\mu$-calculus with renaming rule follows from Theorem 3.12.

4. $\lambda\mu$-calculi with $\mu\eta$-reduction

In this section, we recall the $\lambda\mu$-calculi extended by the $\mu\eta$-rule.

4.1. Call-by-name $\lambda\mu$-calculus with $\mu\eta$-rule

First, we consider the call-by-name variant.

Note that the call-by-name $\lambda\mu$-calculus extended by the extensionality rule

$$\lambda x.Mx \to M \qquad (x \notin FV(M)) \qquad (\eta)$$

is not confluent as a counterexample is given by Saurin [9]: For $M = \lambda x.(\mu\alpha.\lambda y.\mu\beta.[\alpha]([\alpha]y))x$, we have $M \to M_1 = \mu\alpha.\lambda y.\mu\beta.[\alpha]([\alpha]y)$ by η and $M \to M_2 = \lambda x.\mu\alpha.\lambda y.\mu\beta.[\alpha]([\alpha]yx)x$ while M_1 and M_2 are distinct normal forms. Therefore, we do not consider η for the call-by-name variant.

Definition 4.1 ($\mu\eta$-rule)**.**

$$\mu\alpha.[\alpha]M \to M \qquad (\alpha \notin FV(M)) \qquad (\mu\eta)$$

The reduction $\to$ is defined as the compatible closure of the rules in Definition 3.1 and $(\mu\eta)$.

For this extended calculus, when we try to extend the modified complete development M^* in Definition 3.2 to obtain a mapping satisfying the Z property, a problem arises in the case where a $(\mu\eta)$ redex and an extended renaming redex overlap. Naively, there are three ways to define the mapping: (1) reducing $(\mu\eta)$ redex preferentially, (2) reducing extended renaming redex preferentially, and (3) reducing outer redexes preferentially. However, none of the options satisfy Z property.

(1) (Reducing $(\mu\eta)$ redex preferentially) Consider the β-reduction step

$$M = [\alpha](\mu\beta.(\lambda x.x)([\beta](\mu\gamma.y))) \to [\alpha](\mu\beta.[\beta](\mu\gamma.y)) = N.$$

M has no overlap, so we have

$$M^* = ((\lambda x.x)([\beta](\mu\gamma.y)))^*[\beta := \alpha] = ([\beta](\mu\gamma.y))^*[\beta := \alpha] = y.$$

On the other hand, the redexes of $(\mu\eta)$ and (Ren) overlap in N. If we give priority to $(\mu\eta)$, we have

$$N^* = [\alpha](\mu\gamma.y).$$

Hence, $M^* \to N^*$ does not hold.

(2) (Reducing extended renaming redex preferentially) Consider the β-reduction step

$$M = \mu\alpha.[\alpha]((\lambda x.x)((\mu\beta.[\beta]x)y)) \to \mu\alpha.[\alpha]((\mu\beta.[\beta]x)y) = N.$$

M has no extended renaming redex, so we have

$$M^* = ((\lambda x.x)((\mu\beta.[\beta]x)y))^* = ((\mu\beta.[\beta]x)y)^* = xy.$$

On the other hand, N contains an extended renaming redex between two $(\mu\eta)$ redexes. If we give priority to (Ren), we have

$$N^* = \mu\alpha.([\beta]x)^*[[\beta]w := [\alpha](wy)] = \mu\alpha.[\alpha](xy).$$

Hence, $M^* \to N^*$ does not hold.

(3) (Reducing outer redexes preferentially) Consider the β-reduction step

$$\begin{aligned} M &= \mu\alpha.((\lambda x.x)([\alpha]((\mu\beta.[\beta](\mu\gamma.[\gamma]([\gamma]x)))y))) \\ &\to \mu\alpha.[\alpha]((\mu\beta.[\beta](\mu\gamma.[\gamma]([\gamma]x)))y) = N. \end{aligned}$$

M and N have consecutive overlaps of $(\mu\eta)$ and extended renaming redexes. The outermost redex except for the β-redex in M is the extended renaming redex $[\alpha]((\mu\beta\cdots)y)$. If we give priority to outer redexes, we have

$$\begin{aligned} M^* &= \mu\alpha.(([\beta](\mu\gamma.[\gamma]([\gamma]x)))^*[[\beta]w := [\alpha](wy)]) \\ &= \mu\alpha.(([\beta]([\beta]x))[[\beta]w := [\alpha](wy)]) \\ &= \mu\alpha.([\alpha]([\alpha]xy)y). \end{aligned}$$

On the other hand, we have

$$\begin{aligned} N^* &= ((\mu\beta.[\beta](\mu\gamma.[\gamma]([\gamma]x)))y)^* \\ &= (\mu\gamma.[\gamma]([\gamma]x))^* y \\ &= (\mu\gamma.[\gamma]([\gamma]x))y, \end{aligned}$$

since the outermost redexes in each line is the $(\mu\eta)$ redexes $\mu\alpha.[\alpha]\cdots$ and $\mu\beta.[\beta]\cdots$. Hence, $M^* \to N^*$ does not hold.

This problem is solved by the compositional Z (Theorem 2.7) as follows. We define $\to_1$ as the compatible closure of $(\mu\eta)$ rule, and $\to_2$ as the compatible closure of the rules (β), (Str), and (Ren) in Definition 3.1. Clearly, $\to\ =\ \to_1 \cup \to_2$ holds.

We define M^{*_1} and M^{*_2} as follows. M^{*_1} is the standard complete development for the $\mu\eta$-reduction. M^{*_2} is almost the same as Definition 3.2 except all consecutive Str-redexes are also reduced.

Definition 4.2. The mapping M^{*_1} and M^{*_2} are defined as follows.

Definition of M^{*_1}:

1. $x^{*_1} = x$
2. $(\lambda x.M)^{*_1} = \lambda x.M^{*_1}$
3. $(M_1M_2)^{*_1} = M_1^{*_1}M_2^{*_1}$
4. $(\mu\alpha.[\alpha]M)^{*_1} = M^{*_1} \qquad (\alpha \notin FV(M))$
5. $(\mu\alpha.M)^{*_1} = \mu\alpha.M^{*_1} \qquad$ (otherwise)
6. $([\alpha]M)^{*_1} = [\alpha]M^{*_1}$

Definition of M^{*_2}:

1. $x^{*_2} = x$
2. $(\lambda x.M)^{*_2} = \lambda x.M^{*_2}$
3. $((\lambda x.M_1)M_2)^{*_2} = M_1^{*_2}[x := M_2^{*_2}]$
4. $((\mu\alpha.M_1)N_1 \dots N_n)^{*_2} = \mu\alpha.M_1^{*_2}[[\alpha]w := [\alpha](wN_1^{*_2} \dots N_n^{*_2})]$
5. $(M_1M_2)^{*_2} = M_1^{*_2}M_2^{*_2} \qquad$ (otherwise)
6. $(\mu\alpha.M)^{*_2} = \mu\alpha.M^{*_2}$
7. $([\alpha](\mu\beta.M))^{*_2} = M^{*_2}[\beta := \alpha]$
8. $([\alpha](\mu\beta.M)N_1 \dots N_n)^{*_2} = M^{*_2}[[\beta]w := [\alpha](wN_1^{*_2} \dots N_n^{*_2})]$
9. $([\alpha]M)^{*_2} = [\alpha]M^{*_2} \qquad$ (otherwise)

In the following, we prove that $*_1$ and $*_2$ satisfy the conditions of Theorem 2.7, and hence the composition $((\cdot)^{*_1})^{*_2}$ satisfies the Z property for the call-by-name $\lambda\mu$-calculus with the renaming and the $\mu\eta$ rules.

Lemma 4.3. *(1)* $M \twoheadrightarrow_1 M^{*_1}$.

(2) $M \twoheadrightarrow_2 M^{*_2}$.

Proof. They are proved by straightforward induction on M. □

Lemma 4.4. *(1)* $M^{*_1}[x := N^{*_1}] = (M[x := N])^{*_1}$

(2) $M^{*_1}[[\alpha]w := [\alpha](wN^{*_1})] = (M[[\alpha]w := [\alpha](wN)])^{*_1}$

(3) $M^{*_2}[x := N^{*_2}] \twoheadrightarrow_2 (M[x := N])^{*_2}$

(4) $M^{*_2}[[\alpha]w := [\alpha](wN^{*_2})] \twoheadrightarrow_2 (M[[\alpha]w := [\alpha](wN)])^{*_2}$

Proof. (1) It is proved by induction on M. We give only a non-trivial case, and the other cases are similarly proved.
(Case $M = \mu\alpha.[\alpha]M_1$ and $\alpha \notin FV(M_1)$) By renaming bound variables, we can assume that $\alpha \notin FV(N)$.

$$\begin{aligned}
M^{*_1}[x := N^{*_1}] &= M_2^{*_1}[x := N^{*_1}] \\
&= (M_1[x := N])^{*_1} \qquad \text{(I.H.)} \\
&= (\mu\alpha.[\alpha](M_2[x := N]))^{*_1} \qquad (\text{by } \alpha \notin FV(M_2[x := N])) \\
&= (M[x := N])^{*_1}
\end{aligned}$$

(2) It is proved by straightforward induction on M.

(3) It is proved by induction on M. The proof is almost the same as that of Lemma 3.4 (1).

(4) It is proved by induction on M. The proof is almost the same as that of Lemma 3.4 (2).

□

Proposition 4.5. *(1)* $M \to_1 N$ *implies* $N \twoheadrightarrow_1 M^{*_1}$

(2) $M \to_2 N$ *implies* $N \twoheadrightarrow (M^{*_1})^{*_2}$

Proof. (1) It is proved by straightforward induction on $M \to_1 N$ by Lemma 4.3 (1).

(2) It is proved by induction on $M \to_2 N$ by Lemma 4.3 (2).
(Case $M = \mu\alpha.[\alpha]M_1 \to_2 \mu\alpha.M_2 = N$, $\alpha \notin FV(M_1)$, and $[\alpha]M_1 \to_2 M_2$) In this case, we have $(M^{*_1})^{*_2} = (M_1^{*_1})^{*_2}$.

(Subcase $M_1 = \mu\beta.M_3$, $\alpha \notin FV(M_3)$, and $M_2 = M_3[\beta := \alpha]$)

$$\begin{aligned} N &= \mu\alpha.M_3[\beta := \alpha] \\ &= \mu\beta.M_3 && \text{(by } \alpha \notin FV(M_3)\text{)} \\ &\twoheadrightarrow ((\mu\beta.M_3)^{*_1})^{*_2} && \text{(by Lemma 4.3)} \\ &= (M^{*_1})^{*_2} \end{aligned}$$

(Otherwise) In this case, we have $M_2 = [\alpha]M_3$ and $M_1 \to_2 M_3$ for some M_3, and then

$$\begin{aligned} N &= \mu\alpha.[\alpha]M_3 \\ &= M_3 && \text{(by } \alpha \notin FV(M_1)\text{)} \\ &\twoheadrightarrow (M_1^{*_1})^{*_2} && \text{(I.H.)} \\ &= (M^{*_1})^{*_2} \end{aligned}$$

The other cases are similarly proved.

□

Proposition 4.6. *(1) $M \to_1 N$ implies $M^{*_1} \twoheadrightarrow_1 N^{*_1}$*

(2) $M \to_1 N$ implies $M^{*_2} \twoheadrightarrow N^{*_2}$

(3) $M \to_2 N$ implies $M^{*_1} \twoheadrightarrow N^{*_1}$

(4) $M \to_2 N$ implies $M^{*_2} \twoheadrightarrow_2 N^{*_2}$

Proof. (1) It is proved by induction on $M \to_1 N$. Note that we cannot prove $M^{*_1} = N^{*_1}$, since we have the following case.

$$M = \mu\alpha.\mu\beta.[\beta]([\alpha]M_2) \to_1 \mu\alpha.[\alpha]M_2 = N,$$

where $\alpha, \beta \notin FV(M_2)$. In this case, we have $M^{*_1} = \mu\alpha.[\alpha]M_2^{*_1}$ and $N^{*_1} = M_2^{*_1}$.

(2) It is proved by induction on $M \to_1 N$.
(Case $M = \mu\alpha.[\alpha]((\mu\beta.M_1)\overrightarrow{N}) \to_1 (\mu\beta.M_1)\overrightarrow{N} = N$, where $\alpha \notin FV(M_1, \overrightarrow{N})$)
By definition of M^{*_2}, $\alpha \notin FV(M_1^{*_2}, \overrightarrow{N}^{*_2})$.

$$\begin{aligned} M^{*_2} &= \mu\alpha.M_1^{*_2}[[\beta]w := [\alpha](w\overrightarrow{N}^{*_2})] \\ &= \mu\beta.M_1^{*_2}[[\beta]w := [\beta](w\overrightarrow{N}^{*_2})] && \text{(by } \alpha \notin FV(M_1^{*_2}, \overrightarrow{N}^{*_2})\text{)} \\ &= N^{*_2} \end{aligned}$$

(Case $M = (\mu\alpha.[\alpha](\mu\beta.M_1)\overrightarrow{L})\overrightarrow{N}M_2 \to_1 (\mu\beta.M_1)\overrightarrow{L}\overrightarrow{N}M_2 = N$, where $\alpha \notin FV(M_1, \overrightarrow{L})$)
By definition of M^{*_2}, $\alpha \notin FV(M_1^{*_2}, \overrightarrow{L}^{*_2})$. In the following, θ denotes the structural substitutions $[[\alpha]w := [\alpha](w\overrightarrow{N}^{*_2}M_2^{*_2})]$.

$$\begin{aligned}
M^{*_2} &= \mu\alpha.M_1^{*_2}[[\beta]w := [\alpha](w\overrightarrow{L}^{*_2})]\theta \\
&= \mu\alpha.M_1^{*_2}[[\beta]w := [\alpha](w\overrightarrow{L}^{*_2}\overrightarrow{N}^{*_2}M_2^{*_2})] \quad (\text{by } \alpha \notin FV(M_1^{*_2}, \overrightarrow{L}^{*_2})) \\
&= \mu\beta.M_1^{*_2}[[\beta]w := [\beta](w\overrightarrow{L}^{*_2}\overrightarrow{N}^{*_2}M_2^{*_2})] \quad (\text{by } \alpha \notin FV(M_1^{*_2}, \overrightarrow{L}^{*_2})) \\
&= ((\mu\beta.M_1)\overrightarrow{L}\overrightarrow{N}M_2)^{*_2} \\
&= N^{*_2}
\end{aligned}$$

(Case $M = [\alpha]((\mu\beta.[\beta](\mu\gamma.M_1)\overrightarrow{L})\overrightarrow{N}) \to_1 [\alpha]((\mu\gamma.M_1)\overrightarrow{L}\overrightarrow{N}) = N$, where $\beta \notin FV(M_1, \overrightarrow{L})$)
By definition of M^{*_2}, $\beta \notin FV(M_1^{*_2}, \overrightarrow{L}^{*_2})$. In the following, θ denotes the structural substitutions $[[\beta]w := [\alpha](w\overrightarrow{N}^{*_2})]$.

$$\begin{aligned}
M^{*_2} &= M_1^{*_2}[[\gamma]w := [\beta](w\overrightarrow{L}^{*_2})]\theta \\
&= M_1^{*_2}[[\gamma]w := [\alpha](w\overrightarrow{L}^{*_2}\overrightarrow{N}^{*_2})] \qquad (\text{by } \beta \notin FV(M_1^{*_2}, \overrightarrow{L}^{*_2})) \\
&= ([\alpha]((\mu\gamma.M_1)\overrightarrow{L}\overrightarrow{N}))^{*_2} \\
&= N^{*_2}
\end{aligned}$$

The other cases are similarly proved.

(3) The proof is by induction on the structure of $M \to_2 N$.
(Case β)
$M = (\lambda x.M_1)N_1 \to_2 M_1[x := N_1] = N$

$$\begin{aligned}
M^{*_1} &= (\lambda x.M_1^{*_1})N_1^{*_1} \\
&\to M_1^{*_1}[x := N_1^{*_1}] \\
&\twoheadrightarrow (M_1[x := N_1])^{*_1} \qquad (\text{by Lemma 4.4 (1)}) \\
&= N^{*_1}
\end{aligned}$$

(Case Str)
$M = (\mu\alpha.M_1)N_1 \to_2 \mu\alpha.M_1[[\alpha]w := [\alpha](wN_1)] = N$
(Subcase $M_1 = [\alpha]M_2$, where $\alpha \notin FV(M_2)$)
By Str, $N = \mu\alpha.[\alpha](M_2N_1)$
Clearly, $\alpha \notin FV(N_1)$.

$$\begin{aligned}
M^{*_1} &= M_2^{*_1}N_1^{*_1} \\
&= N^{*_1}
\end{aligned}$$

(Subcase Otherwise)
Because $\mu\alpha.M_1$ is not $\mu\eta$-redex, $\mu\alpha.M_1[[\alpha]w := [\alpha](wN_1)]$ is not $\mu\eta$-redex.

$$\begin{aligned} M^{*_1} &= (\mu\alpha.M_1^{*_1})N_1^{*_1} \\ &\to \mu\alpha.M_1^{*_1}[[\alpha]w := [\alpha](wN_1^{*_1})] \\ &\twoheadrightarrow \mu\alpha.(M_1[[\alpha]w := [\alpha](wN_1)])^{*_1} && \text{(by Lemma 4.4 (2))} \\ &= N^{*_1} \end{aligned}$$

(Case Ren)
$M = [\beta](\mu\alpha.M_1) \to_2 M_1[\alpha := \beta] = N$
(Subcase $M_1 = [\alpha]M_2$, where $\alpha \notin FV(M_2)$)
$N = [\beta]M_2$

$$\begin{aligned} M^{*_1} &= [\beta]M_2^{*_1} \\ &= N^{*_1} \end{aligned}$$

(Subcase Otherwise)

$$\begin{aligned} M^{*_1} &= [\beta](\mu\alpha.M_1^{*_1}) \\ &\to M_1^{*_1}[\alpha := \beta] \\ &= (M_1[\alpha := \beta])^{*_1} \\ &= N^{*_1} \end{aligned}$$

(Case $M = \mu\alpha.M_1 \to_2 \mu\alpha.M_2 = N$, where $M_1 \to_2 M_2$)
(Subcase $M_1 = [\alpha]M_3$, where $\alpha \notin FV(M_3)$)
$M^{*_1} = M_3^{*_1}$
(Subsubcase $M_3 = \mu\beta.M_4$, where $\alpha \notin FV(M_4), M_2 = M_4[\beta := \alpha]$)

$$\begin{aligned} M^{*_1} &= (\mu\beta.M_4)^{*_1} \\ &= (\mu\alpha.M_4[\beta := \alpha])^{*_1} && \text{(by } \alpha \notin FV(M_4)\text{)} \\ &= N^{*_1} \end{aligned}$$

(Subsubcase $M_2 = [\alpha]M_4$, where $M_3 \to_2 M_4$)
By definition of $\to_2$, $\alpha \notin FV(M_4)$

$$\begin{aligned} M^{*_1} &= M_3^{*_1} \\ &\twoheadrightarrow M_4^{*_1} && \text{(I.H.)} \\ &= N^{*_1} && \text{(by } \alpha \notin FV(M_4)\text{)} \end{aligned}$$

(Subcase $M^{*_1} = \mu\alpha.M_1^{*_1}$)

$$\begin{aligned} M^{*_1} &= \mu\alpha.M_1^{*_1} \\ &\twoheadrightarrow_2 \mu\alpha.M_2^{*_1} && \text{(I.H.)} \end{aligned}$$

(Subsubcase $M_2 = [\alpha]M_3$, where $\alpha \notin FV(M_3)$)
By definition of M^{*_1}, $\alpha \notin FV(M_3^{*_1})$

$$\begin{aligned} M^{*_1} &\twoheadrightarrow_2 \mu\alpha.M_2^{*_1} \\ &= \mu\alpha.([\alpha]M_3)^{*_1} \\ &= \mu\alpha.[\alpha](M_3^{*_1}) \\ &\rightarrow M_3^{*_1} && (\text{by } \alpha \notin FV(M_3)) \\ &= N^{*_1} \end{aligned}$$

(Subcase Otherwise)

$$\begin{aligned} M^{*_1} &\twoheadrightarrow_2 \mu\alpha.M_2^{*_1} \\ &= N^{*_1} \end{aligned}$$

The other cases are similarly proved.

(4) It is proved by induction on $M \rightarrow_2 N$. The proof is almost the same as that of Proposition 3.6.

□

Proposition 4.7. *$M \rightarrow_2 N$ implies $(M^{*_1})^{*_2} \twoheadrightarrow (N^{*_1})^{*_2}$*

Proof. This holds by Proposition 4.6 (2), (3), and (4). □

Theorem 4.8. *(1) M^{*_1} satisfies the Z property for $(M, \rightarrow_1)$.*
(2) $M \rightarrow_1 N$ implies $M^{*_2} \twoheadrightarrow N^{*_2}$.
(3) $M \twoheadrightarrow M^{*_2}$.
(4) M^{*_1} satisfies the weak Z property for $(M, \rightarrow_2)$ by $\rightarrow$.

Proof. (1) By Proposition 4.5 (1) and Proposition 4.6 (1).
(2) By Proposition 4.6 (2).
(3) By definition of M^{*_2}.
(4) By Proposition 4.5 (2) and Proposition 4.7.

□

Corollary 4.9. *The call-by-name $\lambda\mu$-calculus with the $\mu\eta$-reduction is confluent.*

Proof. By Theorem 2.7 and Theorem 4.8. □

4.2. Call-by-value $\lambda\mu$-calculus with η- and $\mu\eta$-rules

In this section, we recall the call-by-value $\lambda\mu$-calculus [3, 4] extended by the $\mu\eta$-rule and the η_V-rule, which is the call-by-value variant of the η-rule.

Definition 4.10 ($\mu\eta$-rule and η_V-rule)**.**

$$\lambda x.Vx \to V \ (x \notin FV(M)) \qquad (\eta_V)$$
$$\mu\alpha.[\alpha]M \to M \ (\alpha \notin FV(M)) \qquad (\mu\eta)$$

The reduction $\to$ is defined as the compatible closure of the rules in Definition 3.8 and (η_V) and ($\mu\eta$).

Unlike the η rule in the call-by-name variant, the η_V-rule does not break the confluence since V is a value and Vx is not a Str-redex.

We define M^{*_1} and M^{*_2} as follows. M^{*_1} is the standard complete development of $\mu\eta$-reduction and η_V-reduction. M^{*_2} is almost the same as Definition 3.9 except all consecutive Str-redexes and Sym-redexes are also reduced.

Definition 4.11. The mapping M^{*_1} and M^{*_2} are defined as follows.

Definition of M^{*_1}:

1. $x^{*_1} = x$
2. $(\lambda x.Vx)^{*_1} = V^{*_1} \qquad (x \notin FV(M))$
3. $(\lambda x.M)^{*_1} = \lambda x.M^{*_1}$
4. $(M_1M_2)^{*_1} = M_1^{*_1}M_2^{*_1}$
5. $(\mu\alpha.[\alpha]M)^{*_1} = M^{*_1} \qquad (\alpha \notin FV(M))$
6. $(\mu\alpha.M)^{*_1} = \mu\alpha.M^{*_1} \qquad$ (otherwise)
7. $([\alpha]M)^{*_1} = [\alpha]M^{*_1}$

Definition of M^{*_2}:

1. $x^{*_2} = x$
2. $(\lambda x.M)^{*_2} = \lambda x.M^{*_2}$
3. $((\lambda x.M_1)V_2)^{*_2} = M_1^{*_2}[x := V_2^{*_2}]$
4. $(\mathcal{E}[\mu\alpha.M])^{*_2} = \mu\alpha.M^{*_2}[[\alpha]w := [\alpha](\mathcal{E}^*[w])]$
5. $(M_1M_2)^{*_2} = M_1^{*_2}M_2^{*_2} \qquad$ (otherwise)
6. $(\mu\alpha.M)^{*_2} = \mu\alpha.M^{*_2}$

7. $([\alpha](\mathcal{E}[\mu\beta.V]))^{*_2} = V^{*_2}[[\beta]w := [\alpha](\mathcal{E}^{*_2}[w])]$
8. $([\alpha]M)^{*_2} = [\alpha]M^{*_2}$ (otherwise),

where $\mathcal{E}^{*_2}[\,]$ is defined as

$$[\,]^{*_2} = [\,] \qquad (\mathcal{E}M)^{*_2} = \mathcal{E}^{*_2}M^{*_2} \qquad (V\mathcal{E})^{*_2} = V^{*_2}\mathcal{E}^{*_2}.$$

This mapping satisfies the conditions of the compositional Z. It is proved in a similar way to the call-by-name case.

Theorem 4.12. *(1) M^{*_1} satisfies the Z property for $(M, \to_1)$.*

(2) $M \to_1 N$ implies $M^{*_2} \twoheadrightarrow N^{*_2}$.

(3) $M \twoheadrightarrow M^{*_2}$.

(4) M^{*_1} satisfies the weak Z property for $(M, \to_2)$ by $\to$.

Corollary 4.13. *The call-by-value $\lambda\mu$-calculus with the $\mu\eta$-reduction and η_V-reduction is confluent.*

Acknowledgements. This work was partially supported by Grants-in-Aid for Scientific Research KAKENHI 18K11161 (to Yuki Honda and Koji Nakazawa) and 17K05343 and 20K03711 (to Ken-etsu Fujita).

References

[1] Baba, K., Hirokawa, S., and Fujita, K. Parallel reduction in type free $\lambda\mu$-calculus. *Electronic Notes in Theoretical Computer Science*, 42:52–66, January 2001.

[2] Dehornoy, P. and van Oostrom, V. Z, proving confluence by monotonic single-step upperbound functions. In *Logical Models of Reasoning and Computation (LMRC-08)*, 2008.

[3] Fujita, K. Explicitly typed $\lambda\mu$-calculus for polymorphism and call-by-value. In *Typed Lambda Calculi and Applications, 4th International Conference (TLCA 1999)*, volume 1581 of *Lecture Notes in Computer Science*, pages 162–176, 1999.

[4] Fujita, K. Domain-free $\lambda\mu$-calculus. *Theoretical Informatics and Applications*, 34:433–466, 2000.

[5] Nakazawa, K. Confluency and strong normalizability of call-by-value $\lambda\mu$-calculus. *Theoretical Computer Science*, 290:429–463, 2003.

[6] Nakazawa, K. and Fujita, K. Compositional Z: Confluence proofs for permutative conversion. *Studia Logica*, 104:1205–1224, 2016.

[7] Ong, C.-H.L. and Stewart, C.A. A curry-howard foundation for functional computation with control. In *24th Annual ACM Symposium of Principles of Programming Languages (POPL 1997)*, pages 215–227, 1997.

[8] Parigot, M. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR '92)*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

[9] Saurin, A. Typing streams in the $\Lambda\mu$-calculus. *ACM Transactions on Computational Logic*, 11:1–34, 2010.