Z for call-by-value

Koji Nakazawa[1], Ken-etsu Fujita[2], and Yuta Imagawa[1]

[1] Graduate School of Informatics, Nagoya University, Nagoya, JAPAN
knak@i.nagoya-u.ac.jp, imagawa.yuuta@d.mbox.nagoya-u.ac.jp
[2] Graduate School of Science and Technology, Gunma University, Gunma, JAPAN
fujita@cs.gunma-u.ac.jp

Abstract

This work gives a new proof of the confluence of Carraro and Guerrieri's call-by-value lambda calculus λ_v^σ with permutation rules by the extended Z theorem, the compositional Z theorem.

1 Introduction

In general, it is hard to prove the confluence of higher-order rewriting, including (extensions of) the lambda calculus. There is a long history to give simple (or elegant) proofs for the confluence of higher-order rewriting systems and the lambda calculi: Church and Rosser's proof with the notion of the residuals of redexes, Tait and Martin-Löf's proofs with the parallel reduction, and Takahashi's proof with the complete development. Dehornoy and van Oostrom's Z theorem in [5] is one of the newest techniques which is widely applicable to confluence proofs in general setting. The Z theorem says that confluence of abstract rewriting system follows from existence of a mapping satisfying the Z property: any one-step reduction $a \to b$ implies $b \to^* f(a) \to^* f(b)$. This theorem has been applied to some variants of the lambda calculus in [5, 6, 2, 8].

The Z theorem is further generalized by Nakazawa and Fujita as the compositional Z theorem in [7], which enables us to use the Z theorem with dividing rewriting systems into two or more subsystems. The compositional Z gives a simple proof of confluence of the lambda calculus with permutation rules for direct sums. Ando's original confluence proof in [3] for that system is much complicated, and we cannot naïvely apply the original Z theorem because it seems hard to directly define a mapping satisfying the Z property for both beta and permutation reductions.

Carraro and Guerrieri's lambda calculus λ_v^σ in [4] is a call-by-value variant of the lambda calculus, in which they adopt permutation rules to avoid that reductions unexpectedly get stuck. In [1], this calculus is also called the shuffling calculus λ_{shuf}, and further discussed as one of call-by-value variants of the lambda calculus for open terms. As for the permutation rules for direct sums, the permutation rules of λ_v^σ make the confluence proof much harder, and we cannot straightforwardly adapt the ordinary proof techniques with the parallel reduction or the complete development. Carraro and Guerrieri proved the confluence of λ_v^σ by the commutativity of the β_v reduction and the permutation rules.

In this work, we show that the compositional Z theorem can be applied to λ_v^σ. As in [7], we cannot naïvely adapted the original Z theorem for λ_v^σ, and the problem can be avoided by the compositional Z. Although the outline of our proof follows [7], the main difference is that the mapping we consider here may leave redexes of permutation reductions, because the calculus has two kinds (directions) of permutation rules. Hence, we cannot apply the simplified variant of the compositional Z (Corollary 2.4 in [7]).

2 λ_v^σ

In this section, the call-by-value lambda calculus λ_v^σ is introduced.

The values and terms of λ_v^σ are given as follows.

$$V ::= x \mid \lambda x.M \qquad \text{(values)}$$
$$M ::= V \mid MM \qquad \text{(terms)}$$

The reduction rules of λ_v^σ are given as follows.

$$(\lambda x.M)V \to_{\beta_v} [V/x]M$$
$$(\lambda x.M)NL \to_{\sigma_1} (\lambda x.ML)N \qquad (x \in FV(L))$$
$$V((\lambda x.M)N) \to_{\sigma_3} (\lambda x.VM)N \qquad (x \in FV(V))$$

Here, V denotes values, M, N, and L denote terms, and $[V/x]M$ is the usual capture-avoiding substitution. $FV(M)$ denotes the set of free variables in M. We consider the compatible closure of the reduction rules.

This calculus is introduced for operational characterization of solvability in call-by-value lambda calculi in particular for open terms. In Plotkin's original call-by-value lambda calculus λ_v in [9], the term $M \equiv (\lambda yx.xx)(zz)(\lambda x.xx)$ is stuck since zz is not a value, whereas semantically it is equivalent to $(\lambda x.xx)(\lambda x.xx)$ and hence unsolvable. In λ_v^σ, M is reduced to $(\lambda y.(\lambda x.xx)(\lambda x.xx))(zz)$ by the σ-rules, and we can discover the redex $(\lambda x.xx)(\lambda x.xx)$.

3 Compositional Z

We summarize Dehornoy and van Oostrom's Z theorem [5], and then extend it for compositional functions, called the *compositional Z* [7]. It gives a sufficient condition for that a compositional function satisfies the Z property, and enables us to consider a reduction system by dividing into two parts to prove confluence.

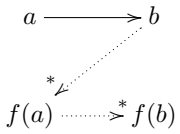
Definition 3.1 ((Weak) Z property)**.** Let $(A, \to)$ be an abstract rewriting system, and $\to^*$ be the reflexive transitive closure of $\to$. Let $\to_{\mathsf{x}}$ be another relation on A, and $\to_{\mathsf{x}}^*$ be its reflexive transitive closure.

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

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

When f satisfies the (weak) Z property, we also say that f *is (weakly) Z.*

It becomes clear why we call it the Z property when we draw the condition as the following diagram.



Theorem 3.2 (Z theorem [5])**.** *If there exists a mapping satisfying the Z property for an abstract rewriting system, then it is confluent.*

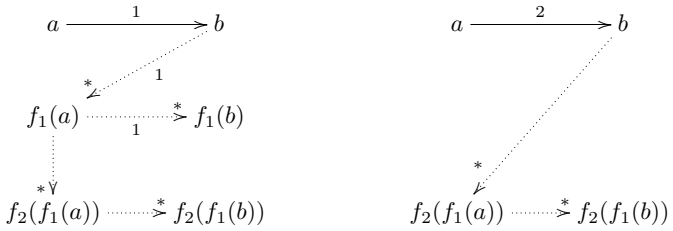


Figure 1: Proof of Theorem 3.3

In fact, we can often prove that the usual complete developments have the Z property.

The compositional Z is the following, which is easily proved from Theorem 3.2 with the diagrams in Figure 1.

Theorem 3.3 (Compositional Z [7]). *Let* $(A, \to)$ *be an abstract rewriting system, and* $\to$ *be* $\to_1 \cup \to_2$. *If there exist mappings* $f_1, f_2 : A \to A$ *such that*

(a) f_1 *is Z for* $\to_1$

(b) $a \to_1 b$ *implies* $f_2(a) \to^* f_2(b)$

(c) $a \to^* f_2(a)$ *holds for any* $a \in \mathrm{Im}(f_1)$

(d) $f_2 \circ f_1$ *is weakly Z for* $\to_2$ *by* $\to$,

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

One of the simplest applications of the compositional Z is for the $\beta\eta$-reduction on the untyped lambda calculus (although it can be directly proved by the Z theorem as in [6]). In [7], the compositional Z is used for the lambda calculus with permutation rules for direct sums such as

$$(\mathsf{case}\ M\ \mathsf{with}\ \mathsf{inl}\ x_1 \Rightarrow N_1 \mid\ \mathsf{inr}\ x_2 \Rightarrow N_2)L \to \mathsf{case}\ M\ \mathsf{with}\ \mathsf{inl}\ x_1 \to N_1 L \mid\ \mathsf{inr}\ x_2 \to N_2 L,$$

which is denoted as $(M[x_1.N_1, x_2, N_2])L \to M[x_1.N_1 L, x_2, N_2 L]$ in [7]. Such permutation rules make confluence proofs much harder because of the critical pair: $M \equiv P[x_1.N_1, x_2.N_2][y_1.L_1, y_2.L_2]K$ is reduced to both

$$M_1 \equiv P[x_1.N_1[y_1.L_1, y_2.L_2], x_2.N_2[y_1.L_1, y_2.L_2]]K$$
$$M_2 \equiv P[x_1.N_1, x_2.N_2][y_1.L_1 K, y_2.L_2 K].$$

These can be reduced to a common term $M_3 \equiv P[x_1.N_1[y_1.L_1 K, y_2.L_2 K], x_2.N_2[y_1.L_1 K, y_2.L_2 K]]$, but in $M_1 \to^* M_3$ we have to reduce the redex which does not occur in M_1. Due to this fact, we can apply neither the ordinary parallel-reduction technique nor the original Z theorem.

4 Confluence of λ_v^σ by compositional Z

In this section, we see that λ_v^σ contains similar critical pairs to those in the lambda calculus with direct sums, and that the compositional Z solves the problem.

Let's consider the term $M \equiv (\lambda x.N)((\lambda y.K)L)P$, which is reduced to both

$$M_1 \equiv (\lambda x.NP)((\lambda y.K)L) \qquad (\text{by } \sigma_1)$$
$$M_2 \equiv (\lambda y.(\lambda x.N)K)LP \qquad (\text{by } \sigma_3).$$

These terms are reduced to a common term $M_3 \equiv (\lambda y.(\lambda x.NP)K)L$. $M_1 \to M_3$ is one-step σ_3 whereas $M_2 \to^* M_3$ consists of two σ_1 steps:

$$M_2 \equiv (\lambda y.(\lambda x.N)K)LP \to_{\sigma_1} (\lambda y.(\lambda x.N)KP)L \to_{\sigma_1} (\lambda y.(\lambda x.NP)K)L,$$

where the σ_1-redex of the second step does not occur in M_2. This means that we cannot naïvely extend the ordinary parallel reduction, by which M_2 is not reduced to M_3 in one step, and that, in the mapping satisfying the Z property, we have to reduce successive σ-reductions at once. For example, the above example shows that M must be mapped to M_3 or its reduct. We consider the auxiliary mapping $M@N$ to reduce successive σ-reductions such as

$$(\lambda y.(\lambda x.N)K)L@P \equiv (\lambda y.(\lambda x.N@P)K)L.$$

Here, the major difference from [7] is that there are two kinds of permutation rules σ_1 and σ_3 with opposite direction. We consider the following two auxiliary mappings $@_1$ and $@_3$:

$$\begin{array}{rlrl} (\lambda x.M)N@_1P &\equiv (\lambda x.M@_1P)N & V@_3(\lambda x.M)N &\equiv (\lambda x.V@_3M)N \\ M@_1P &\equiv MP \quad \text{(otherwise)} & V@_3M &\equiv VM \quad \text{(otherwise)} \end{array}$$

As for [7], the following naïve mapping does not satisfy the Z property.

$$\begin{array}{rl} x^* &\equiv x \\ (\lambda x.M)^* &\equiv \lambda x.M^* \\ ((\lambda x.M)V)^* &\equiv [V^*/x]M^* \\ (MN)^* &\equiv M^*@_1N^* \quad (M \text{ not a value}) \\ (VN)^* &\equiv V^*@_3N^* \quad \text{(otherwise)} \end{array}$$

For $P \equiv (\lambda x.xy)(\lambda z.z)v$ and $Q \equiv (\lambda x.xyv)(\lambda z.z)$, we have $P \to_{\sigma_1} Q$, but we also have $P^* \equiv (\lambda z.z)y@_1v \equiv (\lambda z.zv)y$ and $Q^* \equiv (\lambda z.z)yv$, and hence $P^* \not\to^* Q^*$.

In order to apply the compositional Z theorem, we divide the reductions of λ_v^σ into β_v and σ, and define the mapping $(\cdot)^{\mathsf{S}}$ and $(\cdot)^{\mathsf{B}}$ as follows.

$$\begin{array}{rlrl} x^{\mathsf{S}} &\equiv x & x^{\mathsf{B}} &\equiv x \\ (\lambda x.M)^{\mathsf{S}} &\equiv \lambda x.M^{\mathsf{S}} & (\lambda x.M)^{\mathsf{B}} &\equiv \lambda x.M^{\mathsf{B}} \\ (MN)^{\mathsf{S}} &\equiv M^{\mathsf{S}}@_1N^{\mathsf{S}} \quad (M \text{ not a value}) & ((\lambda x.M)V)^{\mathsf{B}} &\equiv [V^{\mathsf{B}}/x]M^{\mathsf{B}} \\ (VN)^{\mathsf{S}} &\equiv V^{\mathsf{S}}@_3N^{\mathsf{S}} & (MN)^{\mathsf{B}} &\equiv M^{\mathsf{B}}N^{\mathsf{B}} \quad \text{(otherwise)} \end{array}$$

Theorem 4.1. *The two mappings $(\cdot)^{\mathsf{S}}$ and $(\cdot)^{\mathsf{B}}$ (for σ- and β_v-reductions, respectively) satisfy the conditions for the compositional Z, and hence λ_v^σ is confluent.*

The outline of the proof almost follows [7]. However, in contrast to [7], the mapping $(\cdot)^{\mathsf{S}}$ does not necessarily collapse σ-steps, that is, there are terms M and N such that $M \to_\sigma N$ and $M^{\mathsf{S}} \to^+ N^{\mathsf{S}}$. For example, we have

$$M \equiv (\lambda x.x)(yz)((\lambda v.v)w) \to_{\sigma_1} (\lambda x.x((\lambda v.v)w))(yz) \equiv N,$$

and

$$\begin{array}{l} M^{\mathsf{S}} \equiv ((\lambda x.x)(yz))^{\mathsf{S}}@_1((\lambda v.v)w)^{\mathsf{S}} \equiv (\lambda x.x((\lambda v.v)w))(yz) \\ N^{\mathsf{S}} \equiv (\lambda x.x((\lambda v.v)w))^{\mathsf{S}}@_3(yz)^{\mathsf{S}} \equiv (\lambda x.x@_3((\lambda v.v)w))(yz) \equiv (\lambda x.(\lambda v.xv)w)(yz). \end{array}$$

The σ_3-redex $x((\lambda v.v)w)$ in M^{S} is created in the application of $(\cdot)^{\mathsf{S}}$, and it is not reduced in M^{S}. Hence, we cannot apply the simplified variant of the compositional Z (Corollary 2.4 in [7]) for these mappings.

References

[1] Accattoli, B. and Guerrieri, G. Open call-by-value. In *Asian Symposium on Programming Languages and Systems (APLAS 2016)*, volume 10017 of *Lecture Notes in Computer Science*, pages 206–226, 2016.

[2] Accattoli, B. and Kesner, D. The permutative λ-calculus. In *Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR 2012)*, volume 7180 of *Lecture Notes in Computer Science*, pages 15–22, 2012.

[3] Ando, Y. Church-Rosser property of a simple reduction for full first-order classical natural deduction. *Annals of Pure and Applied Logic*, 119:225–237, 2003.

[4] Carraro, A. and Guerrieri, G. A semantical and operational account of call-by-value solvability. In *Foundations of Software Science and Computation Structures (FoSSaCS 2014)*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–18. Springer, 2014.

[5] 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.

[6] Komori, Y., Matsuda, N., and Yamakawa, F. A simplified proof of the church-rosser theorem. *Studia Logica*, 102(1):175–183, 2013.

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

[8] Nakazawa, K. and Nagai, T. Reduction system for extensional lambda-mu calculus. In *25th International Conference on Rewriting Techniques and Applications joint with the 12th International Conference on Typed Lambda Calculi and Applications (RTA-TLCA 2014)*, volume 8560 of *Lecture Notes in Computer Science*, pages 349–363, 2014.

[9] Plotkin, G. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1:125–159, 1975.