Duality between Call-by-value Reductions and Call-by-name Reductions

Daisuke Kimura

Wadler proposed the dual calculus, which corresponds to classical sequent calculus LK, and studied the relationship between the $\lambda\mu$-calculus and the dual calculus as equational systems to explain the duality between call-by-value and call-by-name in a purely syntactical way. Wadler left an open question whether one can obtain similar results by replacing the equations with reductions. This paper gives one answer to his question. We first refine the $\lambda\mu$-calculus as reduction systems by reformulating sum types and omitting problematic reduction rules that are not simulated by reductions of the dual calculus. Secondly, we give translations between the call-by-name $\lambda\mu$-calculus and the call-by-name dual calculus, and show that they preserve the call-by-name reductions. We also show that the compositions of these translations become identity maps up to the call-by-name reductions. We also give translations for the call-by-value systems, and show that they satisfy properties similar to the call-by-name translations. Thirdly, we introduce translations between the call-by-value $\lambda\mu$-calculus and the call-by-name one by composing the above translations with duality on the dual calculus. We finally obtain results corresponding to Wadler’s, but our results are based on reductions.

1. Introduction

The Curry-Howard correspondence for classical logic

In the last twenty years, a lot of work has been done to extend the Curry-Howard correspondence to classical logic. Felleisen\(^3\) introduced the $C$ operator to model call/cc, Griffin\(^5\) observed that the type of call/cc corresponds to Peirce’s Law and extended the Curry-Howard correspondence to classical logic. In this line, the $\lambda\mu$-calculus introduced by Parigot\(^7\) is well known. This calculus corresponds to classical natural deduction and has a simple structure, sufficient expressive power, and nice computational properties such as confluence and strong normalization. Later, a call-by-value (CBV) variant of the $\lambda\mu$-calculus was proposed by Ong and Stewart\(^6\).

Duality

The call-by-name and call-by-value strategies have been well studied as evaluation strategies of programming languages. Filinski\(^4\) suggested that duality between call-by-name and call-by-value is clarified by two notions of programs and continuations. Selinger\(^9\) gave categorical semantics of the call-by-name and call-by-value $\lambda\mu$-calculi and explained Filinski’s duality in terms of categorical duality.

Wadler’s dual calculus and open question

Wadler\(^10,11\) proposed the dual calculus, which corresponds to Gentzen’s classical sequent calculus LK. LK is an appropriate formulation of classical logic that clearly expresses the duality that exists inside classical logic. The main feature of the dual calculus is that it has both terms and continuations as primitives. The computational meaning of the duality of classical logic is expressed in the dual calculus by the duality of terms and continuations. In the dual calculus, call-by-name and call-by-value strategies become dual strategies.

Wadler\(^11\) introduced the translation from the $\lambda\mu$-calculus into the dual calculus, and its inverse translation from the dual calculus into the $\lambda\mu$-calculus. He showed that these translations form an equational correspondence, as defined by Sabry and Felleisen\(^8\). Moreover, he gave the translation from the $\lambda\mu$-calculus into itself by composing the above translations with duality on the dual calculus. This translation satisfies the following properties.

- It takes call-by-value equalities into call-by-value equalities, and vice versa.
- It is an involution up to call-by-value/call-by-name equality.

In other words, he explained Filinski’s duality in a purely syntactical way. However, the $\lambda\mu$-calculus and the dual calculus adopted in his
paper were equational systems, and his results are based on equalities. This is because some rules of the λµ-calculus are not simulated by reductions of the dual calculus, and it is also problematic to introduce some rules, such as (η)-rules, as reductions. But when we discuss whether duality between call-by-value and call-
by-name also holds as a computational procedure, we should consider reductions. In fact, Wadler noted an open question in his paper whether one can replace the equations of his paper with reductions, and extend the properties with equations to properties with reductions.

Our purpose, problems, and solutions
Our purpose in this paper is to answer his question. We encounter problems when we try to obtain refined results by replacing the equations of his paper with reductions. These problems are grouped in the following three cases.

Problem 1 (ζ)-rules
To simulate (ζ)-rule under Wadler’s translation (−)*, we need (βR)-reductions of the dual calculus in both directions. We give a typical example of this problem.

\begin{align*}
((\mu \alpha. [\gamma] \lambda x. [\alpha] y) z)^* &
\equiv (([(x.(y \bullet \alpha)) \not \bullet \gamma].\alpha \bullet (z@\beta)))^* \\
=^{n_{(\beta R)}} ([(x.(y \bullet (z@\beta))) \not \bullet \gamma].\beta) \\
=^{n_{(\beta R)}} ([(x.(y \bullet (z@\delta))).\delta \not \bullet \beta] \not \bullet \gamma).\beta \\
\equiv (\mu \beta. [\gamma] \lambda x. [\beta] (y z))^* 
\end{align*}

Problem 2 (ηv)-rule: \( M =_n \mu(\alpha, \beta).[\alpha, \beta] M \)
To simulate (ηv)-rule under Wadler’s translation (−)*, we need both of (ηv)-reduction and (ηv)-expansion of the dual calculus.

\begin{align*}
((\mu(\alpha, \beta).[\alpha, \beta] M)^* &
\equiv (((((\alpha) \bullet \mu(\beta).[\alpha, \beta]) \not \bullet \gamma).\beta \inr \gamma)^* \\
=^{n_{(\eta v)}} (((\alpha) \bullet \mu(\beta).[\alpha, \beta]) \not \bullet \gamma)^* \\
=^{n_{(\beta R)}} (((\alpha) \bullet \mu(\beta).[\alpha, \beta]) \not \bullet \gamma)^* \beta \\
=^{n_{(\alpha)}} (((\alpha) \bullet \mu(\beta).[\alpha, \beta]) \not \bullet \gamma)^* \alpha \\
=^{n_{(\eta v)}} (((\alpha) \bullet \mu(\beta).[\alpha, \beta]) \not \bullet \gamma)^* \alpha \\
=^{n_{(\beta R)}} (((\alpha) \bullet \mu(\beta).[\alpha, \beta]) \not \bullet \gamma)^* \alpha \\
\equiv (\lambda x. M x)^* 
\end{align*}

Problem 3 (ηp)-rule: \( M =_n \lambda x. M x \)
To simulate (ηp)-rule under Wadler’s translation (−)*, we need both of (ηp)-reduction and (ηp)-expansion of the dual calculus.

\begin{align*}
(\lambda x. M x)^* &
\equiv (\lambda x. [(\alpha) \not \bullet \gamma].\gamma)^* \\
=^{n_{(\eta p)}} (\lambda x. [(\alpha) \not \bullet \gamma].\alpha)^* \\
=^{n_{(\beta p)}} (\lambda x. [(\alpha) \not \bullet \gamma].\gamma)^* \\
=^{n_{(\eta p)}} (\lambda x. [(\alpha) \not \bullet \gamma].\alpha)^* \\
\equiv (\lambda x. M x)^* 
\end{align*}

However, we need (ηp)-rule to simulate \( x(\mu(\alpha.S)) =_n S^{x[-]}[\alpha][-] \), which is defined in the call-by-value λµ-calculus as a part of (ζ)-rule. For example,

\begin{align*}
(x(\mu(\alpha[S]) λz.[\alpha] y))^* &
\equiv x \bullet \not \bullet (\lambda z. (y \bullet \alpha) \not \bullet \beta.\alpha) \\
=^{n_{(\eta p)}} (y'.(x \bullet \not (y')) \not \bullet \beta.\alpha) 
\end{align*}
For \((\eta_D)\)-rule of the \(\lambda\mu\)-calculus, we also encounter a problem similar to this one.

Problem 1 is due to the so-called administrative redexes, and can be solved by modifying Wadler’s translations. The idea of this modification is similar to the modified CPS translation introduced by de Groote\(^1, 2\). However, we need different modifications for call-by-value and call-by-name calculi.

Problem 2 is caused by the difference in how sums are formulated in the dual calculus, and can be solved by modifying Wadler’s (Theorem 40), but our results are similar property for the call-by-value translations (Proposition 35, 36). In Section 6, we introduce translations between the call-by-value and call-by-name \(\lambda\mu\)-calculi by composing the above translations with duality on the dual calculus. We finally obtain results corresponding to Wadler’s (Theorem 40), but our results are based on reductions.

2. The \(\lambda\mu\)-calculus

In this paper, we consider the two variants of the \(\lambda\mu\)-calculus, call-by-value and call-by-name, as reduction systems.

The types of the \(\lambda\mu\)-calculus in this paper follow Wadler, i.e., let \(A\) and \(B\) range over types, then a type is atomic X, a conjunction \(A\land B\), a disjunction \(A\lor B\), a negation \(\neg A\), or an implication \(A\rightarrow B\).

Definition (Types of the \(\lambda\mu\)-calculus)

\(A, B ::= X \mid A\&B \mid A\lor B \mid A\rightarrow B \mid \neg A\)

Two disjoint countable sets of variables for the \(\lambda\mu\)-calculus are given, one is called variables (denoted by \(x, y, z, \ldots\)) and the other is called covariables (denoted by \(\alpha, \beta, \gamma, \ldots\)). We distinguish two notions of terms (denoted by \(M, N, \ldots\)) and statements (denoted by \(S, T, \ldots\)) as the expression of the \(\lambda\mu\)-calculus following Wadler. A term is a variable \(x\), a \(\lambda\)-abstraction \(\lambda x.M\) or \(\lambda x.S\), an implication application \(OM\) (where \(O: A\rightarrow B\)), a projection \(fst(M)\) or \(snd(M)\), a pairing \(\langle M, N\rangle\), a \(\mu\)-abstraction \(\mu x.S\), or a term for sums. A statement is a covariable application \([\alpha]M\), a negation appli-
cation \(OM\) (where \(O : \neg A\)), or a statement for sums. Any free occurrence of \(x\) in \(M\) and \(S\) is bound in the terms \(\lambda x.M\) and \(\lambda x.S\) respectively. Any free occurrence of \(\alpha\) in \(S\) is bound in the term \(\mu \alpha.S\).

For sums, we give another formulation without following Selinger\(^9\). A term for sums is a left injection \(\text{inl}(M)\), a right injection \(\text{inr}(M)\), and case \(\delta(O,x,M,y,N)\), and a statement for sums is a case \(\delta(O,x,S,y,T)\). Any free occurrences of \(x\) in \(M\) and \(y\) in \(N\) are bound in the term \(\delta(O,x,M,y,N)\). Similarly, any free occurrences of \(x\) in \(S\) and \(y\) in \(T\) are bound in the term \(\delta(O,x,S,y,T)\).

**Definition (Terms and Statements of \(\lambda\mu\))**

\[
\begin{align*}
M, N, O &::= x | \lambda x.M | \lambda x.S | MN | \mu \alpha.S \\
& | \text{fst}(M) | \text{snd}(N) | (M, N) \\
& | \text{inl}(M) | \text{inr}(N) | \delta(O,x,M,y,N) \\
S, T &::= [\alpha]M | MN | \delta(O,x,S,y,T)
\end{align*}
\]

We consider the term modulo \(\alpha\)-conversion of variables and covariables. The sets of free variables of \(M\) and \(S\) (denoted by \(\text{FV}(M)\) and \(\text{FV}(S)\)), and the sets of free covariables of \(M\) and \(S\) (denoted by \(\text{FCV}(M)\) and \(\text{FCV}(S)\)) are defined as usual. A **typing judgment** of the \(\lambda\mu\)-calculus takes the form \(\Gamma \vdash_{\lambda\mu} \Delta \mid M : A\) or \(\Gamma \vdash_{\lambda\mu} \Delta \mid S : A\), where \(\Gamma\) denotes a \(\lambda\)-context, i.e., \(x_1 : A_1, \ldots, x_n : A_n\), and \(\Delta\) denotes a \(\mu\)-context, i.e., \(\alpha_1 : B_1, \ldots, \alpha_m : B_m\). We note that \(\vdash_{\lambda\mu}\) is sometimes written as \(\vdash\). The **typing rules** for the \(\lambda\mu\)-calculus are defined in Fig. 1.

We use two kinds of substitution for the \(\lambda\mu\)-calculus. The first \(M[\overline{N}/x]\) and \(S[\overline{N}/x]\) are the usual substitutions of a term \(N\) for all free occurrences of the variable \(x\) in \(M\) and \(S\). The second \(M[\overline{S}/T] = \overline{S}/\overline{T}\) and \(S[\overline{S}/T] = \overline{S}/\overline{T}\) are substitutions of a statement context \(T\{\}\) (i.e., a statement with a single hole accepting a term) for a covariable \(\alpha\). This second substitution is defined by induction on \(M\) and \(S\) using the following clause.

\[
([\alpha]M)[\overline{T}/[\alpha]()] \equiv T\{M[\overline{T}/[\alpha]()]\}
\]

The other clauses are defined homomorphically like

\[
(MN)[\overline{T}/[\alpha]()] \equiv M[\overline{T}/[\alpha]()]N[\overline{T}/[\alpha]()] - E
\]

\[
\Gamma \vdash_{\lambda\mu} \Delta \mid M : \neg A \quad \Gamma \vdash_{\lambda\mu} \Delta \mid N : A
\]

\[
\begin{align*}
\Gamma, x : A &\vdash_{\lambda\mu} \Delta \mid x : A & \text{Ax} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid M : B & \text{I} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \lambda x.M : A \Rightarrow B & \text{E} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \text{fst}(M) : A & \text{&E}_1 \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \text{snd}(M) : B & \text{&E}_2 \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \text{inl}(M) : A \lor B & \text{I} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \text{inr}(M) : A \lor B & \text{I} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid (M, N) : A & \text{I} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \delta(O,x,M,y,N) : C & \text{E} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \delta(O,x,S,y,T) \vdash_{\lambda\mu} \Delta & \text{E} \\
\Gamma, x : A &\vdash_{\lambda\mu} \Delta \mid S : \neg A & \text{-I} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \alpha : A & \text{Act} \\
\Gamma &\vdash_{\lambda\mu} \Delta \mid \mu \alpha.S : A & \text{Pass}
\end{align*}
\]

**Fig. 1** Typing rules of the \(\lambda\mu\)-calculus.

2.1 The Call-by-name \(\lambda\mu\)-calculus

We need a notion of call-by-name evaluation and statement contexts to introduce the call-by-name \(\lambda\mu\)-calculus, which is equivalent as an equational theory to the one given by Wadler.
A call-by-name evaluation term context (denoted by $E_n, E_n', \ldots$) is a term context with a hole, and a call-by-name evaluation statement context (denoted by $D_n, D_n', \ldots$) is a statement context with a hole. We write $\{\}$ for a hole, and the results of filling a term $M$ in an evaluation context $E_n$ and a statement context $D_n$ are written $E_n[M]$ and $D_n[M]$, respectively.

**Definition (CBN Term and Statement Contexts)**

$E_n, E_n', E_n'' ::= \{\} \mid E_n M \mid \text{fst}(E_n) \mid \text{snd}(E_n)$

$D_n, D_n' ::= [\alpha]E_n \mid E_n M \mid \delta(E_n, x, E_n' \{x\}, y, E_n'' \{y\})$

In the following, we say a term $M$ is a **simple form with respect to $x$** if there is a call-by-name evaluation term context $E$ such that $M \equiv E\{x\}$ and $x$ is not free in $E$, and a statement $S$ is a **simple form with respect to $x$** if there is a call-by-name evaluation statement context $D$ such that $S \equiv D\{x\}$ and $x$ is not free in $D$.

The one-step call-by-name reduction relation for the $\lambda\mu$-calculus, denoted by $\rightarrow_n^*$, is defined as the compatible closure of the rules in Fig. 2. We write $\rightarrow_n$ for the reflexive transitive closure of $\rightarrow_n$. Similarly, we write $\rightarrow_n^+$ and $\equiv_n$ for the transitive closure and the reflexive symmetric transitive closure of $\rightarrow_n$ respectively.

In the following, when expressions $X$ and $Y$ are in a relation $\mathcal{R}$ of system $S$, we write $S \vdash X \mathcal{R} Y$. For example, we write $\lambda \mu M \rightarrow_n N$ if a term $M$ of the $\lambda\mu$-calculus reduces a term $N$ by the one-step call-by-name reduction of the $\lambda\mu$-calculus.

$(\beta)$-rules reduce a deconstructor applied to a constructor. Note that $(\beta_c)$-rule has an unusual and a restricted form using the call-by-name evaluation and statement contexts. This restriction is needed to obtain sums equivalent to ones formulated in Wadler’s call-by-name system. $(\zeta)$-rules substitute an evaluation context and a statement context for a covalence, and $(\eta)$-rule introduces a $\mu$-abstraction applied to a covalent application. $(\pi)$-rules correspond to the permutative conversions, and $(\nu)$-rules expand a case statement $\delta(M, x, S, y, T)$ when $S$ or $T$ is not a simple form with respect to $y$ and introduce new bindings. These rules are also needed to obtain sums equivalent to sums formulated in Wadler’s system.

We write the $\lambda\mu$-calculus given by Wadler by $\lambda\mu_{\text{wad}}$ and write the call-by-name and call-by-value variants of $\lambda\mu_{\text{wad}}$ by $\lambda\mu_{\text{wad}}^\text{CBN}$ and $\lambda\mu_{\text{wad}}^\text{CV}$, respectively. Detailed definitions of these systems can be found in Wadler (2005). We compare our call-by-name system with the $\lambda\mu_{\text{wad}}^\text{CBN}$-calculus. The differences between them are summarized in the following three points:

- Our system is based on reduction relations while $\lambda\mu_{\text{wad}}$ is based on equations,
- formulates sums differently, and
- does not have $(\eta)$-rules related to implications, negations, pairs, and sums while his system does have them.

We give two translations, $\llbracket - \rrbracket$ and $\llbracket - \rrbracket$, between our $\lambda\mu$ and $\lambda\mu_{\text{wad}}$ that interpret sums as follows.

$$\llbracket (\alpha, \beta).S \rrbracket$$

$$\equiv \mu \gamma.\llbracket S \rrbracket[\gamma\text{inl}(\cdot)/\alpha_{\cdot}\gamma, \gamma\text{inr}(\cdot)/\beta_{\cdot}]$$

$$\llbracket (\alpha, \beta).M \rrbracket \equiv \delta(\llbracket M \rrbracket, x, [\alpha]x, y, [\beta]y)$$

The other clauses are defined homomorphi?
cally like $[MN] \equiv [M] \beta [N]$. 
$$\langle \in (M) \rangle \equiv \mu (\alpha, \beta, [\alpha] \in (M))$$
$$\langle \in (N) \rangle \equiv \mu (\alpha, \beta, [\beta] \in (N))$$
$$\langle \delta (O, x.M, y.N) \rangle \equiv \mu \gamma. (\lambda y. [\gamma] \in (N)) \langle \mu \beta. (\lambda x. [\gamma] \in (M)) \rangle$$
$$\langle \delta (O, x.S, y.T) \rangle \equiv \lambda y. \langle \delta (T) \rangle \langle \mu \beta. (\lambda x. \in (S)) \rangle \langle \mu \alpha. [\alpha, \beta] \in (O) \rangle$$

The other clauses are defined homomorphically like $\langle MN \rangle \equiv \langle M \rangle \langle N \rangle$.

We define the call-by-name system $\lambda \mu_n^{\mathrm{wad}}$ as the one generated by the rules in Fig. 2 and the following $\eta$-rules,

$$(\eta) \quad M \longrightarrow_n \lambda x. M x$$

(where $M : A \supset B$)

$$(\eta) \quad M \longrightarrow_n \in (\lambda x. \in (M))$$

(where $M : A \& B$)

$$(\eta) \quad M \longrightarrow_n \delta (M, x.\in (x), y.\in (y))$$

(where $M : A \lor B$)

$$(\eta) \quad M \longrightarrow_n \lambda x. M x$$

(where $M : \neg A$)

$\lambda \mu_n^{\mathrm{wad}}$ is equivalent to Wadler’s call-by-name system $\lambda \mu_n^{\mathrm{wad}}$ as an equational system. To show this, we need some preparations. Let $E_n$ and $D_n$ be the call-by-name evaluation contexts of the $\lambda \mu_n^{\mathrm{wad}}$-calculus. Then, we define the call-by-name contexts $E_n$ and $D_n$ of the $\lambda \mu_n^{\mathrm{wad}}$-calculus as follows.

$$\{ - \} \equiv \{ - \}$$
$$\in (E_n) \equiv \in (\in (E_n))$$
$$\delta (E_n, x.E_n \{ x \}, y.E_n \{ y \})$$

$$\equiv \mu \gamma. [\gamma] \in (E_n) \langle \mu \beta. [\beta] \in (\mu \alpha. [\alpha, \beta] \in (E_n)) \rangle$$

$$\delta (E_n, x.D_n \{ x \}, y.D_n \{ y \})$$

$$\equiv \{ - \}$$

For the call-by-name contexts $E_n$ and $D_n$ defined above, the following lemma holds.

**Lemma 1**

1. $\lambda \mu_n^{\mathrm{wad}} \vdash \langle E_n \{ M \} \rangle = n \in (E_n \{ M \})$ and $\lambda \mu_n^{\mathrm{wad}} \vdash \langle D_n \{ M \} \rangle = n \in (D_n \{ M \})$ hold for any term $M$ of our $\lambda \mu$-calculus.

2. $\lambda \mu_n^{\mathrm{wad}} \vdash \langle M \rangle \in (D_n \{ \{ - \} \} / [\alpha] \{ - \}) = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

3. $\lambda \mu_n^{\mathrm{wad}} \vdash \langle S \rangle \in (D_n \{ \{ - \} \} / [\alpha] \{ - \}) = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$\lambda \mu_n^{\mathrm{wad}} \vdash [\alpha] \{ - \}$ and $\lambda \mu_n^{\mathrm{wad}} \vdash D_n \{ [\alpha] \{ - \} \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$\lambda \mu_n^{\mathrm{wad}} \vdash E_n \{ \alpha \beta. \mu \alpha. S \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$\lambda \mu_n^{\mathrm{wad}} \vdash D_n \{ \alpha \beta. \mu \alpha. S \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$\lambda \mu_n^{\mathrm{wad}} \vdash [\alpha] \{ - \}$ and $\lambda \mu_n^{\mathrm{wad}} \vdash D_n \{ [\alpha] \{ - \} \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$\lambda \mu_n^{\mathrm{wad}} \vdash E_n \{ [\alpha] \{ - \} \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$\lambda \mu_n^{\mathrm{wad}} \vdash D_n \{ [\alpha] \{ - \} \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$

$$\lambda \mu_n^{\mathrm{wad}} \vdash \langle E_n \rangle \equiv \lambda \mu_n^{\mathrm{wad}} \{ - \}$$

**Proof.** See Appendix.

**Proposition 2**

1. If $\lambda \mu_n^{\mathrm{wad}} \vdash M \equiv_n N$, then $\lambda \mu_n^{\mathrm{wad}} \vdash \langle M \rangle = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$ and $\lambda \mu_n^{\mathrm{wad}} \vdash \langle S \rangle = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$.

2. If $\lambda \mu_n^{\mathrm{wad}} \vdash M \equiv_n N$, then $\lambda \mu_n^{\mathrm{wad}} \vdash [\alpha] \{ - \}$ and $\lambda \mu_n^{\mathrm{wad}} \vdash D_n \{ [\alpha] \{ - \} \} = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$.

3. If $\lambda \mu_n^{\mathrm{wad}} \vdash \langle [\alpha] \{ - \} \rangle = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$ and $\lambda \mu_n^{\mathrm{wad}} \vdash \langle D_n \{ [\alpha] \{ - \} \} \rangle = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$.

4. If $\lambda \mu_n^{\mathrm{wad}} \vdash \langle [\alpha] \{ - \} \rangle = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$ and $\lambda \mu_n^{\mathrm{wad}} \vdash \langle D_n \{ [\alpha] \{ - \} \} \rangle = n \in (\lambda \mu_n^{\mathrm{wad}} \{ - \})$.

**Proof.** See Appendix.

**2.2 The Call-by-value $\lambda \mu$-calculus**

For the call-by-value calculus, we need a notion of values. A *values* (denoted by $V, W, \ldots$) is a variable, a $\lambda$-abstraction, a pair of values, or an injection of a value.

**Definition (Values of the Call-by-value $\lambda \mu$)**

$V, W ::= x \mid \lambda x. M \mid (V, W) \mid \in (V) \mid \in (W) \mid \lambda x. S$

We also use a notion of the call-by-value evaluation term and statement contexts to introduce the call-by-value calculus. However, in this case, it is useful to give the evaluation contexts as singular contexts. The call-by-value evaluation singular contexts (denoted by $E_v, E'_v, \ldots$) are grouped into the elimination contexts (denoted by $E_e, E'_e, \ldots$), which are obtained from an elimination rule. The introduction contexts (denoted by $E_i, E'_i, \ldots$) which are constructed by an introduction rule, and the contexts which have a hole as the argument of a lambda abstraction, i.e., $\langle \lambda x. M \rangle \{ - \}$. The call-by-value evaluation singular statement contexts (denoted by $D_v, D'_v, \ldots$) are grouped into the elimination contexts (denoted by $D_e, D'_e, \ldots$) and the contexts which have a hole as the argument of a lambda abstraction, i.e., $\langle \lambda x. S \rangle \{ - \}$.

**Definition (CBV Evaluation Singular Contexts)**

$E_v ::= (\lambda x. M) \{ - \} \mid E_e \mid E_i$

$E_e ::= \{ - \} M \mid \in (\{ - \}) \mid \in (\{ - \})$

$E_i ::= \in (\{ - \}) \mid \in (\{ - \}) \mid \in (\{ - \})$

$D_v ::= (\lambda x. S) \{ - \} \mid D_e \mid D_i$

$D_i ::= \{ - \} \{ - \} \mid \delta (\{ - \}, x, S, y, T)$

The one-step call-by-value reduction relation for the $\lambda \mu$-calculi, denoted by $\rightarrow_n$, is defined as the compatible closure of the rules in
\[(\beta)\] \((\lambda x.M)V \rightarrow_v M[V/x] \]
\[(\beta_c)\] \(\text{fst}(V,W) \rightarrow_v V \)
\(\text{snd}(V,W) \rightarrow_v W \)
\[(\beta_v)\] \(\delta(\text{inl}(V),x,M,y,N) \rightarrow_v M[V/x] \)
\(\delta(\text{inr}(V),x,M,y,N) \rightarrow_v N[V/x] \)
\(\delta(\text{inl}(V),x,y,T) \rightarrow_v S[V/x] \)
\(\delta(\text{inr}(V),x,y,T) \rightarrow_v T[V/x] \)
\[(\beta_{-})\] \((\lambda x.S)V \rightarrow_v S[V/x] \)
\[(\zeta)\] \(E_{\lambda\mu}(\mu\alpha.S) \rightarrow_v v\mu\beta.S[[\beta]E_{\lambda\mu}(\cdot)[/\alpha][-1]] \)
\(D_v(\mu\alpha.S) \rightarrow_v vS[[\beta]D_v(\cdot)[/\alpha][-1]] \)
\[(\text{comp})\] \(E_{\lambda\mu}(\langle\langle\lambda x.M\rangle\rangle) \rightarrow_v \langle\langle\lambda x.E_{\lambda\mu}\{M\}\rangle\rangle \)
\(D_v(\langle\langle\lambda x.M\rangle\rangle) \rightarrow_v \langle\langle\lambda x.D_v\{M\}\rangle\rangle \)
\[(\pi)\] \(E_{\lambda\mu}(\delta(O,x,M,y.N)) \rightarrow_v \delta(O,x,E_{\lambda\mu}\{M\},y,E_{\lambda\mu}\{N\}) \)
\(D_v(\delta(O,x,M,y.N)) \rightarrow_v \delta(O,x,D_v\{M\},y,D_v\{N\}) \)
\[(\eta_{\mu})\] \(M \rightarrow_v \mu\alpha.[\alpha]M \) (where \(\alpha \notin FCV(M)\))
\[(\text{name})\] \(E_{ie}(O) \rightarrow_v \langle\langle\lambda x.E_{ie}\{x\}\rangle\rangle O \)
\(D_v(\text{name}) \rightarrow_v \langle\langle\lambda x.D_v\{x\}\rangle\rangle O \) (where \(O\) is not a value)
\(\langle\langle\lambda x.V\rangle\rangle \rightarrow_v \lambda x.V \) (where \(O\) is not a value)

**Fig. 3** Reduction rules of the call-by-value \(\lambda\mu\)-calculus.

**Fig. 3.** We write \(\rightarrow^*\) and \(\rightarrow^+_v\) for the reflexive transitive closure and the transitive closure of \(\rightarrow_v\), respectively. \((\beta)\)-rules reduce the deconstructor applied to a constructor with call-by-value restrictions, \((\zeta)\)-rules substitute a call-by-value evaluation context and a statement context for a covalue, \((\eta_{\mu})\)-rule introduces a \(\mu\)-abstraction applied to a covariable application, and \((\pi)\)-rules correspond to the permutation conversions. The \((\text{name})\)-rules push the next term to be evaluated out as an argument of the function. These rules correspond to the \((\text{name})\)-rule of the \(\lambda\mu_{wad}^\text{calculus.\}

The \((\text{comp})\)-rules are associativity rules, which correspond to the \((\text{comp})\)-rule of the \(\lambda\mu_{wad}^\text{-calculus.\}

We now compare our call-by-value system with Wadler’s call-by-value system. The differences between them are summarized in the following four points:

- Our system is based on reduction relations while his system is based on equations,
- formulated sums differently,
- defines values differently: a projection from a value is not a value in our system, and
- does not have \((\eta)\)-rules related to implications, negations, pairs, and sums while his system does have them.

We introduce the call-by-value calculus \(\lambda\mu_{wad}^\text{calculus} as the system generated by the rules in Fig. 3 and the following \((\eta)\)-rules.

\[(\eta_{\rightarrow})\] \(V \rightarrow_n \lambda x.V\chi \) (where \(E_{\lambda\mu}\) is \(E_c\) or \((\lambda x.M)\{\cdot\}\))
\[(\eta_{\rightarrow})\] \(V \rightarrow_n \{\text{fst}(V),\text{snd}(V)\} \) (\(V : A \vee B\))
\[(\eta_{\rightarrow})\] \(M \rightarrow_n \delta(M,x.\text{inl}(x),y.\text{inr}(y)) \) (\(M : A \wedge B\))
\[(\eta_{\rightarrow})\] \(V \rightarrow_n \lambda x.V\chi \) (where \(O\) is not a value, \(E_{ie}\) is \(E_i\) or \(E_c\))
\[(\eta_{\rightarrow})\] \(V \rightarrow_n \lambda x.V\chi \) (where \(O\) is not a value)
\[(\eta_{\rightarrow})\] \(V \rightarrow_n \lambda x.V\chi \) (where \(O\) is not a value)

**Proposition 3**

1. If \(\lambda\mu_{wad}^\text{calculus} \vdash M = v N\), then \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle M\rangle\rangle = v \langle\langle N\rangle\rangle\), and if \(\lambda\mu_{wad}^\text{calculus} \vdash S = v T\), then \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle S\rangle\rangle = v \langle\langle T\rangle\rangle\).
2. If \(\lambda\mu_{wad}^\text{calculus} \vdash M = v N\), then \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle M\rangle\rangle = v \langle\langle N\rangle\rangle\), and if \(\lambda\mu_{wad}^\text{calculus} \vdash S = v T\), then \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle S\rangle\rangle = v \langle\langle T\rangle\rangle\).
3. If \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle M\rangle\rangle = v M\) and \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle S\rangle\rangle = v S\).
4. If \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle M\rangle\rangle = v M\) and \(\lambda\mu_{wad}^\text{calculus} \vdash \langle\langle S\rangle\rangle = v S\).

**Proof.** See Appendix.

We mention some basic properties of the \(\lambda\mu\)-calculus at the end of this section.

**Lemma 4 (Substitution Lemma for \(\lambda\mu\))**

Let \(M\) and \(N\) be terms, and \(S\) and \(T\) be statements of the \(\lambda\mu\)-calculus.

1. Suppose \(\Gamma \vdash_{\lambda\mu} \Delta \vdash M : A\). If \(\Gamma, x : A \vdash_{\lambda\mu} \)
Δ \mid N : B$, then $Γ \vdashλμΔ \mid N^{[M/x]} : B$, and if $Γ, x : A \mid S \vdash λμΔ$, then $Γ \mid S^{[M/x]} \vdashλμΔ$.

(2) Suppose $Γ, x : A \mid T(x) \vdashλμΔ$. If $Γ \vdashλμΔ$, $Δ, α : A \mid N : B$, then $Γ \vdashλμΔ \mid N^{[T(−)/[α(−)]]} : B$, and if $Γ, S \vdash λμΔ, α : A$, then $Γ \mid S^{[T(−)/[α(−)]]} \vdashλμΔ$

where $T(−)$ is a context statement, that is, a statement with a single hole.

Proof. (1) is shown by a straightforward induction on $N$ and $S$. (2) can be shown by an induction on $M$ and $S$ using (1). The key case of (2) is $S \equiv [α]M$. Suppose $Γ \mid [α]M \vdashλμΔ, α : A$ is derived. Since the last rule to obtain this sequent is (pass), we obtain $Γ \vdashλμΔ, α : A \mid M : A$. Hence we have $Γ \vdashλμΔ \mid M^{[T(−)/[α(−)]]} : A$ by the induction hypothesis, and then $Γ \mid T(M^{[T(−)/[α(−)]]}{−}) \vdashλμΔ$ by (1). This means $Γ \vdashλμΔ$. □

Proposition 5 (Subject Reduction for $λμ$)

[Subject Reduction for $λμ$] Let $M$ and $N$ be terms, and $S$ and $T$ be statements of the $λμ$-calculus.

(1) If $Γ \vdashλμΔ \mid M : A$ and $λμM \vdashλμΔ, N$, then $Γ \vdashλμΔ \mid M : A$. If $Γ \mid S \vdashλμΔ$ and $λμS \vdashλμΔ, T$, then $Γ \mid T \vdashλμΔ$.

(2) If $Γ \vdashλμΔ \mid M : A$ and $λμM \vdashλμΔ, N$, then $Γ \vdashλμΔ \mid N : A$. If $Γ \mid S \vdashλμΔ$ and $λμS \vdashλμΔ, T$, then $Γ \mid T \vdashλμΔ$.

Proof. Using the substitution lemma, (1) and (2) are shown by an induction on $\longrightarrow_n$ and $\longrightarrow_v$ respectively. □

The call-by-name and call-by-value $λμ$-calculi given in this paper are confluent. This is proved as a corollary of the results in Section 4 and 5 (see Proposition 37).

3. The Dual Calculus

The dual calculus was proposed by Wadler\textsuperscript{10,11} as a term calculus that corresponds to the classical sequent calculus LK. Wadler\textsuperscript{10} first gave the dual calculus as a reduction system, and introduced it as an equation system in his later paper\textsuperscript{11}. Since we want to consider the system based on reduction relations, we will give the reduction system of the dual calculus referring to the system in his first paper.

Types, variables, and covariables of the dual calculus are the same as those of the $λμ$-calculus.

Definition (Types of the Dual Calculus)

$A, B ::= X \mid A\&B \mid A \lor B \mid A \to B \mid \neg A$

where $X$ is an atomic type.

The expressions of the dual calculus consist of terms (denoted by $M, N, \ldots$), coterms (denoted by $K, L, \ldots$), and statements (denoted by $S, T, \ldots$). A term is either a variable $x$, a pair $⟨M, N⟩$, a left injection $⟨M⟩\text{inl}$ or a right injection $⟨N⟩\text{inr}$, a complement of a coterm $⟨K⟩\text{not}$, a function abstraction $λx.M$, with $x$ bound in $M$, or a covariable abstraction $S.α$ with $α$ bound in $S$. A coterm is either a covariable $α$, a case $[K, L]$; a projection from the left of a product $\text{fst}[K]$ or a projection from the right of a product $\text{snd}[L]$, a complement of a term $\text{not}(M)$, a function application $M@K$, or a variable abstraction $x.S$ with $x$ bound in $S$. A statement is a cut of a term against a coterm $M \bullet K$.

Definition (Expressions of the Dual Calculus)

$M, N ::= x \mid ⟨M, N⟩ \mid ⟨M⟩\text{inl} \mid ⟨M⟩\text{inr} \mid [K]\text{not} \mid λx.M \mid S.α$ (terms)

$K, L ::= α \mid [K, L] \mid \text{fst}[K] \mid \text{snd}[L] \mid \text{not}(M) \mid M@K \mid x.S$ (coterms)

$S, T ::= M \bullet K$ (statements)

The set of free variables and covariables occurring in $M, K, S$ are denoted by $\text{FV}(M)$, $\text{FV}(K)$, and $\text{FV}(S)$ respectively. We identify the two expressions in the $α$-equivalence relation and will use $≡$ for the syntactic identity on the expressions. The expressions $M^{[N/x]}$, $K^{[N/x]}$, and $S^{[N/x]}$ denote the expressions obtained by substituting $N$ for every free occurrence of the variable $x$ in $M, K, S$. The expressions $M^{[ℓ/α]}$, $K^{[ℓ/α]}$ and $S^{[ℓ/α]}$ are similarly defined.

A context of the dual calculus (denoted by $Γ, Σ, \ldots$) is a finite set of term variables annotated with types (denoted by $x_1 : A_1, \ldots, x_m : A_m$), in which each variable occurs once at the most. Similarly, a cocontext of the dual calculus (denoted by $Δ, Λ, \ldots$) is defined as a finite set of covariables with types (denoted by $α_1 : B_1, \ldots, α_m : B_m$). A typing judgment of the dual calculus takes either the form $Γ \vdashdcΔ \mid M : A$, the form $K : A \mid Γ \vdashdcΔ$, or the form $Γ \mid S \vdashdcΔ$. We note that $\vdashdc$ is sometimes written as $\mid$. The typing rules of the dual calculus are shown in Fig. 4. These rules are the same as those in Wadler’s later paper\textsuperscript{11}.

A value of the dual calculus, denoted by $V, W, \ldots$, is a variable $x$, a pair of values $⟨V, W⟩$, an
injection of a value $\langle V \rangle \text{inl}$ or $\langle W \rangle \text{inr}$, a complement of a coterm $\lnot(M)$, or a function $\lambda x. M$.

A covalue of the dual calculus is denoted by $P$, $Q$ . . . . A covalue is a covariable $\alpha$, a case over a pair of covalues $[K, L]$, a projection of a covalue $\text{fst}[P]$ or $\text{snd}[Q]$, a complement of a term $\lnot(M)$, or a function application over a covalue $M @ Q$.

**Definition (Values and Covalues)**

$V, W ::= x | \langle V, W \rangle | \langle V \rangle \text{inl} | \langle W \rangle \text{inr} | [K] \lnot | \lambda x. M$

$P, Q ::= \alpha | [P, Q] | \text{fst}[P] | \text{snd}[Q] | \lnot(M) | M @ Q$

These definitions of the values and covalues are same as those in Wadler (2003) but different from those in Wadler (2005). Note that if we adopt the definitions in Wadler (2005), then terms containing beta redexes at the top level may also be values. For example, a term $\langle (x, y) \bullet \text{fst}[\alpha] \rangle \alpha$ includes a beta redex at the top level even though it is a value according to the definition in Wadler (2005).

A term context for the dual calculus (denoted by $E$) is a term that contains a hole that accepts a term, and a coterm context (denoted by $F$) is a coterm that contains a hole that accepts a coterm. The hole is written $\{-\}$, and the result of filling the hole in the term context $E$ with a term $M$ is written $E\{M\}$. Similarly, the result of filling the hole in the coterm context $F$ with a coterm $K$ is written $F\{K\}$.

**Definition (Term Contexts and Coterm Contexts)**

$E ::= \langle \{-\}, N \rangle | \langle V, \{\{-\}\} \rangle | \langle \{\{-\}\} \rangle \text{inl} | \langle \{\{-\}\} \rangle \text{inr}$

$F ::= [K, \{-\}] | \langle \{-\}, P \rangle | \text{fst}[\{-\}] | \text{snd}[\{-\}] | M @ \{-\}$

Note that the context of the form of $M @ \{-\}$ is defined as a coterm context in this paper, though it was not defined in Wadler (2003). This means the reduction rule

$\alpha \bullet (M @ K) \rightarrow^n (N \bullet (M @ \alpha)) \alpha \bullet K$

is permitted as (name)-rule in our call-by-name calculus. This seems to have added a new rule to Wadler’s original system. However, this rule is not an essentially new rule, because this rule is justified when implication is defined in terms of conjunction, disjunction and negation (see Proposition 7).

The call-by-name reduction relation $\rightarrow^n$ and the call-by-value reduction relation $\rightarrow^v$ of the dual calculus are defined to be the compatible closure of rules in Fig. 5. In the sequel, we use $\rightarrow^*, \rightarrow^{n+}$, and $\rightarrow^n$ as the reflexive transitive closure, the transitive closure, and the symmetric reflexive transitive closure of $\rightarrow^n$ respectively. $\rightarrow^v$, $\rightarrow^{v+}$, and $\rightarrow^v$ are defined similarly.

Some of our reduction rules are slightly different from those in Wadler (2003), but the differences are not essential. ($\beta$)-rules given
here are justified in Proposition 6. \((\text{name})\)-rules correspond to \((\gamma)\)-rules of the dual calculus in Wadler (2003), though these rules are not included in his system. Indeed, we can easily show \((\text{name})\)-rules from \((\gamma)\)-rules using \((\beta_L)\) and \((\beta_R)\)-rules in both the call-by-name and call-by-value systems. Conversely, we can obtain \((\gamma)\)-rules from \((\text{name})\)-rules using \((\eta_L)\) and \((\eta_R)\)-rules.

When a term \(M\) of the dual calculus reduces a term \(N\) by the one-step call-by-name reduction, we write \(DC \vdash M \rightarrow^n N\). We also write \(DC \vdash K \rightarrow^n L, DC \vdash S \rightarrow^n T\) etc. for call-by-value calculus, we also define these notations similarly.

As in Wadler’s original dual calculus, implication can be defined in terms of the other connectives, i.e., the following propositions hold.

**Proposition 6**
Under call-by-value, an implication can be defined by

\[
A \supset B \equiv \neg (A \land \neg B)
\]

\[
\lambda x.M \equiv [z.(z \cdot \text{fst}[x.(z \cdot \text{snd}[\neg(M)])])\not \cdot \text{not}(\langle N, [K] \not\text{ot} \rangle)
\]

The translation of a function abstraction is a value, and the typing and reduction rules for implication can be derived from the typing rules for the other connectives.

**Proof.** The call-by-value \((\beta_\gamma)\)-rule is validated as follows.

(a) If \(N\) is a value \(V\), then

\[
(\lambda x.M) \cdot (V@K)
\]

\[
\equiv z.(z \cdot \text{fst}[x.(z \cdot \text{snd}[\neg(\not(M)])])\not \cdot \text{not}(\langle V, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_\gamma)} (V, [K] \not\text{ot} \cdot z.(z \cdot \text{fst}[x.(z \cdot \text{snd}[\neg(\not(M)])])
\]

\[
\rightarrow_v^{(\beta_L)} (V, [K] \not\text{ot} \cdot \text{fst}[x.(\langle V, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_R)} (\mu x.M) \equiv V \cdot x.(\langle K \not\text{ot} \cdot \text{not}(\not(M)])
\]

\[
\rightarrow_v^{(\beta_\gamma)} (V, [K] \not\text{ot} \cdot \text{fst}[x.(\langle V, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_L)} (V, [K] \not\text{ot} \cdot \text{fst}[x.(\langle V, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_R)} (\mu x.M) \equiv V \cdot x.(\langle K \not\text{ot} \cdot \text{not}(\not(M)])
\]

\[
\rightarrow_v^{(\beta_\gamma)} (V, [K] \not\text{ot} \cdot \text{fst}[x.(\langle V, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_L)} (V, [K] \not\text{ot} \cdot \text{fst}[x.(\langle V, [K] \not\text{ot} \rangle)
\]

**Fig. 5** Reduction rules of the call-by-value and call-by-name dual calculi.

(b) If \(N\) is not a value, we need \((\text{name})\)-rule:

\[
(\lambda x.M) \cdot (N@K)
\]

\[
\equiv z.(z \cdot \text{fst}[x.(z \cdot \text{snd}[\neg(\not(M)])])\not \cdot \text{not}(\langle N, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_\gamma)} (N, [K] \not\text{ot} \cdot z.(z \cdot \text{fst}[x.(z \cdot \text{snd}[\neg(\not(M)])])
\]

\[
\rightarrow_v^{(\beta_L)} (N, [K] \not\text{ot} \cdot \text{fst}[x.(\langle N, [K] \not\text{ot} \rangle)
\]

\[
\rightarrow_v^{(\beta_R)} (N, [K] \not\text{ot} \cdot \text{fst}[x.(\langle N, [K] \not\text{ot} \rangle)
\]

**Proposition 7**
Under call-by-name, an implication can be defined by

\[
A \supset B \equiv \neg A \lor B
\]

\[
\lambda x.M \equiv ([x.((\mu x.M) \not\text{ot} \cdot \text{not}(\not(M)]) \not\text{ot} \cdot \text{not}(\not(N), P])\not\text{ot} \cdot \text{not}(\not(N), P)
\]

The translation of a function application with covalue is a covalue, and the typing and reduction rules for implication can be derived from the typing rules for the other connectives.

**Proof.** The call-by-name \((\beta_\gamma)\) and \((\text{name})\)-rules are validated as follows.

\[
(\lambda x.M) \cdot (N@P)
\]

\[
\equiv (\langle x.((\mu x.M) \not\text{ot} \cdot \text{not}(\not(M)]) \not\text{ot} \cdot \text{not}(\not(N), P])\not\text{ot} \cdot \text{not}(\not(N), P)
\]

\[
\rightarrow_v^{(\beta_\gamma)} (\langle x.((\mu x.M) \not\text{ot} \cdot \text{not}(\not(M)]) \not\text{ot} \cdot \text{not}(\not(N), P])\not\text{ot} \cdot \text{not}(\not(N), P)
\]
\[ \text{DC} \vdash \text{not } \cdot \text{not } \langle N \rangle \]
\[ \text{n}_0^\beta [x.(M \bullet P)] \text{not } \cdot \text{not } \langle N \rangle \]
\[ \text{n}_0^\beta (N \bullet x.(M \bullet P)) \]
\[ \text{n}_0^\beta (M^\alpha / x) \bullet P \]
\[ N \bullet (M \otimes K) \equiv N \bullet \text{not } \langle M \rangle, K \]
\[ \text{n}_0^\beta (\text{name}) \langle N \bullet \text{not } \langle M \rangle, \alpha \rangle \cdot \alpha \bullet K \]
\[ \equiv (N \bullet (M \otimes \alpha)) \cdot \alpha \bullet K \]

We now mention some basic properties of the dual calculus.

**Lemma 8 (Substitution Lemma for DC)**
Let \( M \) and \( N \) be terms, \( K \) and \( L \) be coterm, and \( S \) and \( T \) be statements of the dual calculus.

1. Suppose \( \Gamma \vdash \Delta \mid M : A \). If \( \Gamma, x : A \vdash \Delta \mid N : B \), then \( \Gamma \vdash \Delta \mid N^M/ x : B \). If \( L : B \mid \Gamma, x : A \vdash \Delta \), then \( L^M/x : B \mid \Gamma \vdash \Delta \).

2. Suppose \( K \mid \Gamma \vdash \Delta \), \( \alpha : A \mid \Gamma \vdash \Delta \). If \( \Gamma, x : A \vdash \Delta \mid N^M/x : B \), then \( L : B \mid \Gamma \vdash \Delta \mid N^K/\alpha : B \). If \( L : B \mid \Gamma \vdash \Delta, \alpha : A \), then \( L^K/\alpha : B \mid \Gamma \vdash \Delta \).

**Proof.** (1) and (2) are shown by a straightforward induction on \( M, K, S \), and \( S \).

**Proposition 9 (Subject Reduction for DC)**
Let \( M \) and \( N \) be terms, \( K \) and \( L \) be coterm, and \( S \) and \( T \) be statements of the dual calculus.

1. If \( \Gamma \vdash \Delta \mid M : A \) and \( \text{DC} \vdash M \rightarrow_n L \), then \( \Gamma \vdash \Delta \mid N : A \). If \( K : A \mid \Gamma \vdash \Delta \) and \( \text{DC} \vdash K \rightarrow_l L \), then \( L : A \mid \Gamma \vdash \Delta \).

2. If \( \Gamma \vdash \Delta \mid M : A \) and \( \text{DC} \vdash M \rightarrow_v N \), then \( \Gamma \vdash \Delta \mid N : A \). If \( K : A \mid \Gamma \vdash \Delta \) and \( \text{DC} \vdash K \rightarrow_v L \), then \( L : A \mid \Gamma \vdash \Delta \).

**Proof.** Using the substitution lemma, (1) and (2) are shown by an induction on \( \rightarrow_n \) and \( \rightarrow_v \), respectively.

As Wadler mentioned in his paper, the reductions of his dual calculus are subject. Moreover, if \( \eta_L \), \( \eta_R \), and \( \gamma \)-rules are omitted, then the remaining reductions are strongly normalizing for typed terms. Our systems enjoy similar properties. However, since \( \eta_L \) and \( \eta_R \)-rules are expansions, the full reductions are not strongly normalizing. Moreover, the full reductions of our systems, like Wadler’s original systems, include looping terms even if for typed terms. For example, \( (x, y) \bullet \alpha \) is a typable statement, and this statement loops in the call-by-value calculus.

\[ (x, y) \bullet \alpha \rightarrow_v (\text{name}) (x \bullet \beta, \beta, y) \bullet \alpha \]
\[ \rightarrow_v^\gamma (x \bullet \beta, \beta, z, (z, y) \bullet \alpha) \]

We can give a similar example for the call-by-name calculus.

We now consider the two versions of the dual calculus; one given in Wadler (2003) and the other given in Wadler (2005). For the latter version, we write \( \text{DC}^m \) as the call-by-name system and \( \text{DC}^n \) as the call-by-value system.

The differences between the two versions of the dual calculus are summarized in the following three points:

- The first version is based on reduction relations while the second one is based on equations,
- the first version does not have \( \eta \)-rules related to implications, negations, pairs and sums while the second one does contain them, and
- the second version contains terms of the forms \( V \bullet \text{fst}[\alpha] \cdot \alpha \) and \( V \bullet \text{snd}[\beta] \cdot \beta \) as values and coterm of the forms \( x.(\text{inl} \bullet P) \) and \( y.(\text{inr} \bullet Q) \) as covalues.

4. Translations from the \( \lambda \mu \)-calculus into the Dual Calculus

In this section, we give the translations from the \( \lambda \mu \)-calculus into the dual calculus. We consequently introduce two different translations for the call-by-name and call-by-value calculi, and show that these translations preserve typing and reductions.

4.1 The Naive Translation

In this subsection, we give the naive translation from the \( \lambda \mu \)-calculus into the dual calculus. This translation preserves equalities, but does not preserves reductions.

**Definition (The Naive Translation \((-)^\circ \))**
The naive translation \((-)^\circ \) from the \( \lambda \mu \)-calculus into the dual calculus is defined as follows. This translation maps a term \( M \) and a statement \( S \) of our \( \lambda \mu \)-calculus to a term \( M^\circ \) and a statement \( S^\circ \) of the dual calculus respectively.

\[ (x)^\circ \equiv x \]
\[ ((M, N))^\circ \equiv (M^\circ, N^\circ) \]
\[ (\delta(O, x.M, y.N))^\circ \equiv (O^\circ \cdot [x.(M^\circ \bullet \alpha), y.(N^\circ \bullet \alpha)]) \cdot \alpha \]
\[ (\delta(O, x.S, y.T))^\circ \equiv O^\circ \cdot [x.S^\circ, y.T^\circ] \]
\[ (\text{fst}(O))^\circ \equiv (O^\circ \cdot \text{fst}[\alpha]) \cdot \alpha \]
\[ (\text{inl}(O))^\circ \equiv (O^\circ \cdot \text{inl}) \]
This claim can be shown directly by an in-forward induction on \(\lambda\mu\). (where \(\alpha\) is a fresh covariable)

\[
\begin{align*}
(M)^\ddagger &\equiv (M : \lambda n \alpha)\alpha \\
([\alpha]M)^\ddagger &\equiv M : \lambda n \alpha \\
(OM)^\ddagger &\equiv O : \lambda n \text{not}(M^\ddagger) \\
((\lambda x.S)\alpha)^\ddagger &\equiv N^\ddagger \bullet x.S^\ddagger \\
(\delta(O,x,S,y,T)^\ddagger) &\equiv O^\ddagger \bullet [x.S^\ddagger,y.T^\ddagger] \\
\end{align*}
\]

(2) If \(\lambda\mu\) \(\vdash M \equiv_n N\), then \(\lambda\mu\) \(\vdash M^\ddagger \equiv_n N^\ddagger\).\(\alpha\)

(3) \(\lambda\mu\) \(\vdash S \equiv_n T\), then \(\lambda\mu\) \(\vdash S^\ddagger \equiv_n T^\ddagger\).

\[
\begin{align*}
(snd(O)^\ddagger) &\equiv (O^\ddagger \bullet snd[\alpha])\alpha \\
(inr(O)^\ddagger) &\equiv (O^\ddagger)\text{inr} \\
(OM)^\ddagger &\equiv O^\ddagger \bullet n(\alpha^\ddagger) \\
(\lambda x.S)^\ddagger &\equiv [x.S^\ddagger]\text{not} \\
(\mu\alpha.S)^\ddagger &\equiv S^\ddagger\alpha \\
([\alpha]M)^\ddagger &\equiv M^\ddagger \bullet \alpha \\
(OM)^\ddagger &\equiv (O^\ddagger \bullet (M^\ddagger \bullet \alpha))\alpha \\
(\lambda x.M)^\ddagger &\equiv (\lambda x.M^\ddagger) \bullet \alpha \\
\end{align*}
\]

This naive translation is defined by changing the part of sums of the original translation \((-)^\ddagger\) given in Wadler (2005). The naive translation is consistent with Wadler’s translation in the sense of the following lemma.

**Lemma 10**

Let \(M\) be a term, and \(S\) be a statement of our \(\lambda\mu\)-calculus, then

1. \(\lambda\mu\) \(\vdash DC_{\eta^=n} \vdash \langle M \rangle^* =^n M^\ddagger\) and \(DC_{\eta^=n} \vdash \langle S \rangle^* =^n S^\ddagger\).
2. \(\lambda\mu\) \(\vdash DC_{\eta^=v} \vdash \langle M \rangle^* =^v M^\ddagger\) and \(DC_{\eta^=v} \vdash \langle S \rangle^* =^v S^\ddagger\).

**Proof.** See Appendix.

The naive translation preserves typing rules and equalities.

**Proposition 11**

1. If \(\Gamma \vdash M : A\), then \(\Gamma \vdash M^\ddagger : A\).
2. If \(\lambda\mu\) \(\vdash M \equiv_n N\), then \(\lambda\mu\) \(\vdash M^\ddagger \equiv_n N^\ddagger\).

3. \(\lambda\mu\) \(\vdash S \equiv_n T\), then \(\lambda\mu\) \(\vdash S^\ddagger \equiv_n T^\ddagger\).

**Proof.** (1) We can prove this claim by a straightforward induction on \(\lambda\mu\).

(2) This claim can be shown directly by an induction on \(\equiv_n\). Even if we do not adopt this approach, we can show this claim as a corollary of Theorem 16 using Lemma 12.

(3) As with (2), we can show this claim directly, or as a corollary of Theorem 21 using Lemma 17.

In general, this naive translation does not preserve reductions as well as Wadler’s translation. This is because of the so-called administrative redexes. A typical example is \((\zeta)\)-reduction: \((\mu\alpha.\beta\lambda x.\alpha x)y^\rightarrow_n\mu\gamma.\beta\lambda x.\gamma(xy)

\[
(\mu\alpha.\beta\lambda x.\alpha x y)^\ddagger
\]

\(\equiv((\mu\alpha.\beta\lambda x.\alpha x y)^\ddagger)\).\(\gamma\)

\(\equiv\mu\gamma.\beta\lambda x.\gamma(xy)^\ddagger\).

To solve this problem, we modify the naive translation. The idea of modification is similar to the modified CPS translation by de Groote\(^{1,2}\). In the following two subsections, we give different translations for the call-by-name and call-by-value calculi. This is because the administrative redexes of these two calculi are slightly different.

### 4.2 The Translation from CBN \(\lambda\mu\)-calculus into CBN Dual Calculus

The call-by-name translation consists of the following two translations.

1. \((-)^\ddagger\) maps any term \(M\) and statement \(S\) of the \(\lambda\mu\)-calculus to a term \(M^\ddagger\) and statement \(S^\ddagger\) of the dual calculus, respectively.
2. \((\lambda n K)^\ddagger\) is a translation given by coterm \(K\) and maps any term \(M\) of the \(\lambda\mu\)-calculus to a statement \(M : \lambda n K\) of the dual calculus.

The definition of the call-by-name translation \((-)^\ddagger\) is given in **Fig. 6**. In this figure, \(S.\alpha\bullet K\) means \(S^K[\alpha]\) if \(K\) is a covalue, otherwise it means \(S.\alpha\bullet K\).

This translation is consistent with the naive
translation, that is, the following lemma holds.

**Lemma 12**
Let $M$ and $S$ be a term and a statement of the $\lambda\mu$-calculus, then
1. $DC \vdash M^{\otimes} \bullet P \rightarrow^{n^*} M :_n P$ for any covalue $P$.
2. $DC \vdash M^{\otimes} \rightarrow^{n^*} M^\sharp$;
3. $DC \vdash S^{\otimes} \rightarrow^{n^*} S^\sharp$.

**Proof.** See Appendix.

This translation preserves the typing derivation, that is, the following proposition holds.

**Proposition 13**
(1) If $\Gamma \vdash \lambda\mu \Delta | M : A$, then $\Gamma \vdash_{dc} \Delta | M^\sharp : A$.
(2) If $\Gamma \vdash S : \Delta$, then $\Gamma \vdash S^\sharp : dc \Delta$.

**Proof.** This proposition can be shown by the subject reduction property for the dual calculus using Proposition 11 and Lemma 12. □

**Lemma 14**
(1) Let $M$ and $N$ be terms, and $S$ be a statement of the $\lambda\mu$-calculus, then
$DC \vdash (M :_n P)^{[N/x]} \rightarrow^{n^*} M^{[N/x]} :_n P^{[N^\sharp/x]}$; and $DC \vdash S^{[N^\sharp/x]} \rightarrow^{n^*} S^{[N/x]}$ hold for any covalue $P$.

(2) Let $P$ and $Q$ be covalues, and $M$ be a term of the dual calculus, then
$DC \vdash M \bullet [x,(x \bullet P),y.(y \bullet Q)] \rightarrow^{n^*} M \bullet [P,Q]$.

(3) If $K$ is not a covalue, then $DC \vdash O :_n K \rightarrow^{n^*} O^\sharp \bullet K$ for any term $O$ of the $\lambda\mu$-calculus.

(4) $DC \vdash (M :_n N^\sharp @ P) \rightarrow^{n^*} (MN :_n P)$ and $DC \vdash (M :_n not(N^\sharp)) \rightarrow^{n^*} (MN :_n P)$ for any terms $M$ and $N$ of the $\lambda\mu$-calculus, and covalue $P$.

**Proof.** See Appendix. □

Let $E_n$ and $D_n$ be a call-by-name evaluation term context and statement context of the $\lambda\mu$-calculus, and $P$ be a covalue of the dual calculus, then we define covalues $\Phi(E_n,P)$ and $\Phi(D_n)$ as follows:

- $\Phi(\{\ldots\},P) \equiv P$
- $\Phi(E_n N,P) \equiv \Phi(E_n,N^\sharp @ P)$
- $\Phi(fst(E_n),P) \equiv \Phi(E_n,fst[P])$
- $\Phi(snd(E_n),P) \equiv \Phi(E_n,snd[P])$
- $\Phi(\delta(E_n,x.E'_n\{x\},y.E'_n\{y\}),P) \equiv \Phi(E_n,[\Phi(E'_n,P),\Phi(E'_n,P)])$
- $\Phi([\alpha]E_n) \equiv \Phi(E_n,\alpha)$
- $\Phi(E_n N) \equiv \Phi(E_n,not(N^\sharp))$
- $\Phi(\delta(E_n,x.D'_n\{x\},y.D'_n\{y\}),P) \equiv \Phi(E_n,[\Phi(D'_n),\Phi(D'_n)])$

Then the following properties hold.

**Lemma 15**
(1) If $M$ is not a $\lambda$-abstraction, then $DC \vdash (E_n[M] :_n P) \rightarrow^{n^*} (M :_n \Phi(E_n,P))$ and $DC \vdash (D_n[M])^\sharp \rightarrow^{n^*} (M :_n \Phi(D_n))$ hold.
(2) For any term $M$ of the $\lambda\mu$-calculus, $DC \vdash (M :_n \Phi(E_n,P)) \rightarrow^{n^*} (E_n[M] :_n P)$ and $DC \vdash (M :_n \Phi(D_n)) \rightarrow^{n^*} (D_n[M])^\sharp$ hold.
(3) If $M$ be a term and $S$ be a statement of the $\lambda\mu$-calculus, then $DC \vdash (M :_n P)^{[Φ(D_n)/α]} \rightarrow^{n^*} M^{[D_n(\cdots)[α]} :_n P^{[Φ(D_n)/α]}$, and $DC \vdash S^{[Φ(D_n)/α]} \rightarrow^{n^*} S^{[D_n(\cdots)[α]}$ hold for any covalue $P$.

**Proof.** See Appendix. □

Then, we prove that the call-by-name modified translation $\rightarrow^{n^*}$ preserves reductions.

**Theorem 16 (Soundness of $\rightarrow^{n^*}$)**
(1) If $\lambda\mu | M →_c N$, then $DC | (M :_n P) →^{n^*} (N :_n P)$ for any covalue $P$, especially $DC | M^\sharp →^{n^*} N^\sharp$.
(2) If $\lambda\mu | S →_c T$, then $DC | S^\sharp →^{n^*} T^\sharp$. Moreover, if $→_c n$ is $(\beta_2), (\beta_k), (\beta_v)$ or $(\beta_\nu)$, then $→_c n$ can be replaced by $→^{n^*}$.

**Proof.** See Appendix. □

### 4.3 The Translation from CBV $\lambda\mu$-calculus into CBV Dual Calculus

In this subsection, we introduce the call-by-value translation from the $\lambda\mu$-calculus into the dual calculus by modifying the naive translation $\rightarrow^{\otimes}$. The call-by-name translation also consists of the two translations: $\rightarrow^{\otimes}$ and $\rightarrow^\dagger$. $\rightarrow^\dagger$ maps any term $M$ and statement $S$ of the $\lambda\mu$-calculus to a term $M^\dagger$ and statement $S^\dagger$ of the dual calculus respectively. The infix operator “$\rightarrow^\dagger$” translates a pair of terms $M$ of the $\lambda\mu$-calculus and a coterm $K$ of the dual calculus into a statement $M \rightarrow^\dagger K$ of the dual calculus. The definition of the call-by-value translation $\rightarrow^{\otimes}$ is given in Fig. 7.

When we compare the CBV translation given here with the CBN translation, the definitions of $(μα.S :_v K)$, $((λx.M)N :_v K)$, $δ(O,x.M,y.N) :_v K)$, $((λx.S)N)^\dagger$, and $δ(O,x,S,y.T)^\dagger$ are different. This is because the administrative redexes differ according to the difference of the $ζ$-rules of the call-by-name and call-by-value systems.

Like the call-by-name translation, the call-by-value translation is also consistent with the naive translation in the sense of the following lemma.

**Lemma 17**
Let $M$ and $S$ be a term and a statement of the $\lambda\mu$-calculus, then

- If $M$ is not a $\lambda$-abstraction, then $DC | (E_n[M] :_n P) \rightarrow^{n^*} (M :_n \Phi(E_n,P))$ and $DC | (D_n[M])^\sharp \rightarrow^{n^*} (M :_n \Phi(D_n))$ hold.
- For any term $M$ of the $\lambda\mu$-calculus, $DC | (M :_n \Phi(E_n,P)) \rightarrow^{n^*} (E_n[M] :_n P)$ and $DC | (M :_n \Phi(D_n)) \rightarrow^{n^*} (D_n[M])^\sharp$ hold.
- If $M$ be a term and $S$ be a statement of the $\lambda\mu$-calculus, then $DC | (M :_n P)^{[Φ(D_n)/α]} \rightarrow^{n^*} M^{[D_n(\cdots)[α]} :_n P^{[Φ(D_n)/α]}$, and $DC | S^{[Φ(D_n)/α]} \rightarrow^{n^*} S^{[D_n(\cdots)[α]}$ hold for any covalue $P$.

**Proof.** See Appendix. □
(1) $\text{DC} \vdash M^\circ \bullet K \rightarrow^{\nu^s} M :_{\nu} V$ for any coterm $K$.
(2) $\text{DC} \vdash M^\circ \rightarrow^{\nu^s} M^1$, and
(3) $\text{DC} \vdash S^\circ \rightarrow^{\nu^s} S^1$.

Proof. We prove (1), (2), and (3) by a simultaneous induction on $M$ and $S$. Most parts of this proof are similar to the one in Lemma 12.

For example, Case of $(\lambda x. M)N$:

\[
((\lambda x. M)N)^\circ \bullet K \\
\equiv ((\lambda x. M)^\circ \bullet (N^\circ @\alpha))\alpha \bullet K \\
\rightarrow^{(\betaL)} (\lambda x. M)^\circ \bullet (N^\circ @\alpha) \\
\rightarrow^{(\betaR)} N^\circ \bullet x. (M^\circ \bullet K) \\
I.H. (1) \\
\rightarrow^{\nu^s} N :_{\nu} x. (M :_{\nu} V) \\
\equiv ((\lambda x. M)N) :_{\nu} K \\
\]

This call-by-value translation also preserves the typing derivation.

**Proposition 18**

(1) If $\Gamma \vdash^L_\lambda \Delta \vdash M : A$, then $\Gamma \vdash^L_\nu \Delta \vdash M^1 : A$.
(2) If $\Gamma \vdash^L_\lambda S \vdash \lambda \mu \Delta$, then $\Gamma \vdash^L_\nu S^1 \vdash \lambda \nu \Delta$.

Proof. This proposition can be shown by the subject reduction property for the dual calculus using Proposition 11 and Lemma 17. □

**Definition**

For each value $V$ in the $\lambda \mu$-calculus, we define a value $(V)^\circ$ in the dual calculus as follows.

\[
(x)^\circ \equiv x, \quad ((V, W))^\circ \equiv (V^\circ, W^\circ) \\
\text{inl}(V)^\circ \equiv \langle V^\circ \rangle \text{inl} \quad (\lambda x. M)^\circ \equiv \lambda x. M^1 \\
\text{inr}(W)^\circ \equiv \langle W^\circ \rangle \text{inr} \quad (\lambda x. S)^\circ \equiv [x, S^1] \text{not}
\]

Using this notation, we can show the following lemma.

**Lemma 19**

(1) $\text{DC} \vdash V^\circ \bullet K \rightarrow^{\nu^s} (V :_{\nu} V^\circ \bullet K)$

(2) $\text{DC} \vdash (V :_{\nu} K) \rightarrow^{\nu^s} V^\circ \bullet K$ for any coterm $K$, especially $V^\circ \rightarrow^{\nu^s} V^\circ$.

(3) $\text{DC} \vdash (V :_{\nu} K) \rightarrow^{\nu^s} V^\circ \bullet K$ for any coterm $K$. That is, the statement $(V :_{\nu} V^\circ \bullet K)$ loops in the call-by-value dual calculus.

(3) Let $M$ be a term, $S$ be a statement, and $V$ be a value of the $\lambda \mu$-calculus, then

$\text{DC} \vdash (M :_{\nu} K)[V^\circ / x], (M^1[V^\circ / x] :_{\nu} K)[V^\circ / x]$, and $\text{DC} \vdash S^1[V^\circ / x] \rightarrow^{\nu^s} (S^1[V^\circ / x])^1$ for any coterm $K$.

Proof. (1) is proved by a straightforward induction on $V$. For example, we consider the case of $(V, W)$:

\[
((V, W)^\circ \bullet K) \equiv \langle V^\circ, W^\circ \rangle \bullet K \\
\rightarrow^{\nu^s} \langle V^\circ, W^\circ \rangle \bullet K \equiv \langle V^\circ, W^\circ \rangle^1 \bullet K
\]

(2) is also proved by an induction on $V$. We consider the key cases.

Case of $(V, W)$:

\[
((V, W)^\circ \bullet K) \equiv \langle V^\circ, W^\circ \rangle \bullet K \\
\rightarrow^{\nu^s} \langle V^\circ, W^\circ \rangle \bullet K \equiv \langle V^\circ, W^\circ \rangle^1 \bullet K \\
I.H. \\
\rightarrow^{\nu^s} \langle V^\circ, W^\circ \rangle \bullet K \equiv \langle V^\circ, W^\circ \rangle^1 \bullet K
\]

Cases of $(V)^\circ\text{inl}$ and $(W)^\circ\text{inr}$: this case can be proved in a way similar to the above case using the induction hypothesis and $(\text{name})$-rule.

(3) is proved by a straightforward induction on $M$ and $S$. The key case is:
$$(x \triangleright K)[V^v/x] \equiv (x \bullet K)[V^v/x]$$

$$\equiv V^v \bullet K[V^v/x] \quad \text{(1)}$$

Let $E_v$ be a call-by-value evaluation singular term context, and $K$ be a coterm of the dual calculus. Then we define coterm $\Psi(E_v, K)$ as follows.

$$\Psi(\{\}N, K) \equiv N^\uparrow \circ K$$

$$\Psi((\lambda x. M)\{\}N, K) \equiv x.(M \triangleright_v K)$$

$$\Psi((\{\}N, K) \equiv y.(\langle V^v, y \rangle \bullet K)$$

$$\Psi(\delta(\{\}N), K) \equiv [x.(M \triangleright_v K), y.(N \triangleright_v K)]$$

and for every singular statement context $D_v$, we define coterm $\Psi(D_v)$ as follows.

$$\Psi([\alpha]\{\}N) \equiv \alpha$$

$$\Psi((\lambda x. S)\{\}N) \equiv \lambda x.S^\uparrow$$

$$\Psi(\delta(\{\}N), x.S, y.T) \equiv [x.S^\uparrow, y.T^\uparrow]$$

About this notation, the following properties hold.

**Lemma 20**

(1) Let $M$ not be a $\lambda$-abstraction, then $DC \vdash (E_v \{M \triangleright_v K) \sim^v E(\Psi(E_v, K))$ and $DC \vdash (D_v \{M\})^\uparrow \sim^v (M \triangleright_v E(D_v))$ hold.

(2) Let $E$ be an elimination context or $(\lambda x. M)\{\}N$, and $D_v$ be an evaluation singular statement context. Then $DC \vdash (M \triangleright_v E(K)) \sim^v (E(M) \triangleright_v K) (D_v) \vdash (M \triangleright_v E(D_v)) \sim^v (D_v \{M\})^\uparrow$ hold for any term $M$ of the $\lambda\mu$-calculus.

(3) Let $E_i$ be an introduction context of the $\lambda\mu$-calculus. Then $DC \vdash (M \triangleright_v E_i K) \sim^v ((\lambda x. E_i \{x\}) \triangleright_v K)$ for any term $M$.

(4) Let $E$ be an elimination context or $(\lambda x. M)\{\}N$ of the call-by-value $\lambda\mu$-calculus. Then $DC \vdash (M \triangleright_v E(\Psi(E, \beta)/\alpha)) \sim^v M[\beta_{\Psi(E, \beta)/\alpha}]^\triangleright_v K[\Psi(E, \beta)/\alpha]$ and $DC \vdash S^\uparrow[\Psi(E, \beta)/\alpha] \sim^v (S[\beta_{\Psi(E, \beta)/\alpha}]^\triangleright_v K[\Psi(E, \beta)/\alpha])^\uparrow$ hold for any coterm $K$.

(5) Let $D_v$ be an evaluation singular statement context of the call-by-value $\lambda\mu$-calculus. Then $DC \vdash (M \triangleright_v E(D_v)/\alpha) \sim^v M[D_v(\{\}N)]^\triangleright_v K[\Psi(D_v)/\alpha]$ and $DC \vdash S^\uparrow[\Psi(D_v)/\alpha] \sim^v (S[D_v(\{\}N)]^\triangleright_v K[\Psi(D_v)/\alpha])^\uparrow$ hold for any coterm $K$.

Thus we prove that the call-by-value modified translation $(-)^\uparrow$ preserves reductions.

**Theorem 21 (Soundness of $(-)^\uparrow$)**

(1) If $\lambda\mu \vdash M \rightarrow_v N$, then $DC \vdash (M \triangleright_v K) \rightarrow^v (N \triangleright_v K)$ for any coterm $K$, especially $DC \vdash M^\uparrow \rightarrow^v N^\uparrow$.

(2) If $\lambda\mu \vdash S \rightarrow_v T$, then $S^\uparrow \rightarrow^v T^\uparrow$. Moreover, if $\rightarrow_v$ is $(\beta_\gamma), (\beta_\delta), (\beta_\epsilon)$ or $(\beta_\zeta)$, then $\rightarrow^v$ can be replaced by $\rightarrow^v\uparrow$.

**Proof.** See Appendix.

5. Translation from the Dual Calculus into the $\lambda\mu$-calculus

In this section, we introduce the translations from the dual calculus into the $\lambda\mu$-calculus. As in the previous section, we give two different translations for the call-by-name and call-by-value calculi.

### 5.1 The Naive Translation

In this subsection, we give the naive translation from the dual calculus into the $\lambda\mu$-calculus.

This translation preserves equalities but does not preserve reductions.

**Definition (The Naive Translation $(-)_\oplus$)**

The naive translation from the dual calculus into the $\lambda\mu$-calculus is defined as follows. This translation $(-)_\oplus$ maps a term $M$ and a statement $S$ of the dual calculus to a term $M_\oplus$ and a statement $S_\oplus$ of the $\lambda\mu$-calculus respectively, and maps a coterm $K$ with a term $O$ of the $\lambda\mu$-calculus to a statement $K_\oplus O$ of the $\lambda\mu$-calculus.

$$(x)\oplus \equiv x \quad \alpha_\oplus O \equiv [\alpha]O$$

$$(\lambda x. N)_\oplus \equiv (\lambda x. N)\oplus$$

$$(\lambda x. K) \oplus (x, K) \equiv \delta(O, x.K_\oplus x, y.L_\oplus y)$$

$$(\lambda x. E_\oplus) \equiv \lambda x.E_\oplus$$

$$((\lambda x. E_\oplus)\{x\})_\oplus \equiv (\lambda x. E_\oplus)\{x\}_\oplus$$

$$\lambda x. K_\oplus \equiv \lambda x. K$$

$$\lambda x. E\oplus \equiv \lambda x. E\oplus$$

$$(\lambda x. M)_\oplus \equiv \lambda x. M\oplus$$

$$(\lambda x. E\oplus) \equiv (\lambda x. E\oplus)\oplus$$

$$\lambda x. M_\oplus \equiv \lambda x. M$$

$$\lambda x. E\oplus \equiv \lambda x. E\oplus$$

This naive translation is given by changing the part of sums of the original translation $(-)_\circ$, given by Wadler (2005). The naive translation...
is consistent with Wadler’s translation in the sense of the following lemma.

Lemma 22
Let \( M \) be a term, \( K \) be a coterm, \( S \) be a statement of the dual calculus, and \( O \) be a term of the \( \lambda \mu \text{ word} \)-calculus. Then
\[
\begin{align*}
(1) & \quad \lambda \mu \vdash [M_{\oplus}] =_{n} M_{\oplus}, \quad \lambda \mu \vdash [K_{\ast}\{O\}] =_{n} K_{\ast}\{O\} \quad (\text{if } K_{\ast}\{O\} \neq \text{empty}), \\
(2) & \quad \lambda \mu \vdash [M_{\ominus}] =_{v} M_{\ominus}, \quad \lambda \mu \vdash [K_{\ast}\{O\}] =_{v} K_{\ast}\{O\}.
\end{align*}
\]

Proof. See Appendix. □

The naive translation preserves typing rules and equalities.

Proposition 23
(1) If \( \Gamma \vdash \Delta \vdash M : A \), then \( \Gamma \vdash \lambda \mu \Delta \vdash M_{\oplus} : A \).
(2) If \( K : A \vdash \Gamma \vdash \Lambda \), and \( \Gamma \vdash \lambda \mu \Delta \vdash O : A \), then \( \Gamma \vdash K_{\ast}\{O\} \vdash \lambda \mu \Delta \).
(3) If \( \Gamma \vdash \Delta \vdash S \), then \( \Gamma \vdash S_{\ominus} \vdash \lambda \mu \Delta \).
(4) If \( \Gamma \vdash \Delta \vdash M \), then \( \lambda \mu \vdash M_{\oplus} =_{n} N_{\ominus} \).

Proof. (1) We can prove this claim by a straightforward induction on \( \vdash \).
(2) This claim can be shown directly by an induction on \( =_{n} \). Even if we do not adopt this approach, we can show this claim as a corollary of Theorem 28 using Lemma 24.
(3) As in (2), we can show this claim directly, or as a corollary of Theorem 34 using Lemma 30. □

In general, this naive translation does not preserve reductions as well as Wadler’s translation. \((\eta_{L})\)-rule is a counter-example for the call-by-name system:
\[
\alpha_{\oplus}\{O\} \equiv [\alpha]O \leftarrow_{n} (\lambda x. [\alpha]x)O \equiv (x. (x \bullet \alpha))_{\ominus}\{O\}.
\]

On the other hand, \((\beta_{\eta})\)-rule is a counter-example for the call-by-value system:
\[
\begin{align*}
(\alpha \eta) & \equiv (\alpha \eta)(x \bullet \beta \gamma)_{\oplus} \equiv (\lambda x. [\alpha]x)[\beta \gamma]_{\oplus} \equiv (\alpha \eta)(x \bullet \beta \gamma)_{\ominus} \\
& \equiv \alpha_{\oplus}\{(x \bullet \beta \gamma)_{\oplus}\} \equiv ((x \bullet \beta \gamma)_{\ominus} \alpha)_{\ominus}.
\end{align*}
\]

We also need to modify this naive translation. In the following two subsections, we give different translations for the call-by-name and call-by-value calculi.

5.2 The Translation from CBN Dual Calculus into CBN \( \lambda \mu \)-calculus
To solve the problem for the call-by-name calculus displayed at the end of the previous subsection, we need to modify the translation of the coterm \( x.S \).

Definition \((-)_{\sharp} : \text{CBN DC} \rightarrow \text{CBN } \lambda \mu\)
We introduce the new translation \((-)_{\sharp} \) by modifying the definition of \((-)_{\ast} \) as
\[
(x.S)_{\sharp}\{O\} \equiv S_{\sharp}\{O/x\}.
\]
The other clauses are similar to the naive translation.

The full definition of \((-)_{\sharp} \) is displayed in Fig. 8 (see Appendix).

The following lemma means that the call-by-name translation \((-)_{\sharp} \) is consistent with the naive translation.

Lemma 24
Let \( M \) be a term, \( K \) be a coterm, and \( S \) be a statement of the dual calculus. Then
\[
\begin{align*}
(1) & \quad \lambda \mu \vdash M_{\oplus} \rightarrow_{n} M_{\ominus}, \\
(2) & \quad \lambda \mu \vdash K_{\ast}\{O\} \rightarrow_{v} K_{\ast}\{O\} \quad \text{for any term } O \text{ of the } \lambda \mu \text{-calculus}, \\
(3) & \quad \lambda \mu \vdash S_{\ominus} \rightarrow_{v} S_{\ominus}.
\end{align*}
\]

Proposition 25
(1) If \( \Gamma \vdash \Delta \vdash M : A \), then \( \Gamma \vdash \lambda \mu \Delta \vdash M_{\ominus} : A \).
(2) If \( K : A \vdash \Gamma \vdash \Lambda \), and \( \Gamma \vdash \lambda \mu \Delta \vdash O : A \), then \( \Gamma \vdash K_{\ast}\{O\} \vdash \lambda \mu \Delta \).
(3) If \( \Gamma \vdash \Delta \vdash S \), then \( \Gamma \vdash S_{\ominus} \vdash \lambda \mu \Delta \).

Proof. These claims can be shown by the subject reduction property for the \( \lambda \mu \text{-calculus using Proposition 23 (1) and Lemma 24.} \) □

Lemma 26
(1) If \( \lambda \mu \vdash O \rightarrow_{n} O' \), then \( \lambda \mu \vdash K_{\ast}\{O\} \rightarrow_{v} K_{\ast}\{O'\} \) for any coterm \( K \).
(2) \( M_{\sharp}\{N_{\sharp}/x\} \equiv (M^{[N]/x])_{\sharp}, \quad K_{\ast}\{O\}^{[N_{\sharp}/x]} \equiv (K^{[N]/x})_{\sharp}\{O[N_{\sharp}/x]\} \) and \( S_{\sharp}\{N_{\sharp}/x\} \equiv (S^{[N]/x})_{\sharp} \).
(3) If \( P \) is a covalue, then \( P_{\sharp}\{\} \) is a call-by-name evaluation context of the \( \lambda \mu \text{-calculus.
}\)
(4) \( M_{\sharp}^{[L_{\sharp}/\{\}]}/(\{L\})_{\sharp}, \quad (K_{\ast}\{O\})^{[L_{\sharp}/\{\}]}/(\{L\})_{\sharp}, \quad (K_{\ast}\{O\})^{[L_{\sharp}/\{\}]}/(\{L\})_{\sharp}, \quad (K_{\ast}\{O\})^{[L_{\sharp}/\{\}]}/(\{L\})_{\sharp}, \quad (S^{[L]/\{\}]})_{\sharp} \) for any coterm \( L \).

Proof. (1) The claim is proved by induction on \( K \).
(2) The claim is proved by induction on \( M, K, \) and \( S \).
(3) The claim is proved by induction on \( P \).
(4) The claim is proved by induction on \( M, K, \) and \( S \). We give the key case:
Let \( F \) be a coterm of the dual calculus, and \( O \) be a term of the \( \lambda \mu \)-calculus. Then we define term \( F_\downarrow \{ O \} \) of the \( \lambda \mu \)-calculus as follows:

\[
(\alpha_\downarrow \{ O \})[\downarrow_\alpha \{ - \}] := (\alpha \{ O \})[\downarrow_\alpha \{ - \}]
\equiv L_\downarrow \{ O[\downarrow_\alpha \{ - \}] \}
\]

We give the key cases.

**Case of \((-)\downarrow \{ O \} \equiv O \):**

\[
\text{Let } O \equiv \text{OM}_\downarrow \{ \}
\]

**Case of \([-P]\downarrow \{ O \} \equiv \text{fst}(O)\):**

\[
\text{Let } O \equiv \text{snd}(O)
\]

This notation satisfies the following property.

**Lemma 27**

Let coterm \( L \) not be a covalue, and \( P \) be a covalue. Then

1. \( \lambda \mu \vdash \text{F}\{ L_\downarrow \{ - \} \} \equiv \text{n}_\downarrow \{ F_\downarrow \{ O \} \} \), and
2. \( \lambda \mu \vdash P_\downarrow \{ F_\downarrow \{ O \} \} \equiv \text{n}_\downarrow \{ F_\downarrow \{ P \} \} \)

for any term \( O \) of the \( \lambda \mu \)-calculus.

**Proof.** (1) is proved by a case analysis of \( F \). We give the key cases.

**Case of \([-P]\downarrow \{ - \} :\)

\[
L_\downarrow \{ - \} \equiv \text{fst}(O, x.L_\downarrow \{ x \}, y.P_\downarrow \{ y \})
\]

\[
\equiv \text{snd}(O, x.\alpha.x, y.P_\downarrow \{ y \})
\]

This notation satisfies the following property.

**Lemma 28**

Let coterm \( L \) not be a covalue, and \( P \) be a covalue. Then

1. \( \lambda \mu \vdash \text{F}\{ L_\downarrow \{ - \} \} \equiv \text{n}_\downarrow \{ F_\downarrow \{ O \} \} \), and
2. \( \lambda \mu \vdash P_\downarrow \{ F_\downarrow \{ O \} \} \equiv \text{n}_\downarrow \{ F_\downarrow \{ P \} \} \)

for any term \( O \) of the \( \lambda \mu \)-calculus.

**Proof.** (1) is proved by a case analysis of \( F \). We give the key cases.

**Case of \([-P]\downarrow \{ - \} :\)

\[
L_\downarrow \{ - \} \equiv \text{fst}(O, x.L_\downarrow \{ x \}, y.P_\downarrow \{ y \})
\]

\[
\equiv \text{snd}(O, x.\alpha.x, y.P_\downarrow \{ y \})
\]

This notation satisfies the following property.

**Theorem 28 (Soundness of \((-)_\downarrow \) )**

1. If \( DC \vdash M \equiv_n L \), then \( \lambda \mu \vdash M_\downarrow \equiv_n L_\downarrow \).
   \( \square \)

5.3 The Translation from CBV Dual Calculus into CBV \( \lambda \mu \)-calculus

Now we introduce the modified translation \((-)_\uparrow \) for the call-by-value calculus. The problematic cases of the naive translation were \( (\eta_L) \) and \( (\beta_-) \)-rules. To solve these problems, we introduce a notation: \((\lambda_n.x.S)O\). This means \( S^{O/\alpha}_n \) if \( O \) is a value, otherwise \( (\lambda x.S)O \). Our idea for solving the latter case is to modify the definition of \( M \circ K \) to \( (\lambda_n x.K_\uparrow(x))M_\uparrow \).

**Definition \((\lambda)_\uparrow : \text{CBV DC} \rightarrow \text{CBV } \lambda \mu \)**

We define the translation \((-)_\uparrow \) as follows:

\[
\text{K}_\uparrow \{ O \} \equiv \text{(\lambda}_n x.K_\uparrow(x)) \{ O \}
\]

\[\text{fst}[K_\uparrow \{ O \} \equiv K_\uparrow \{ \text{fst}(O) \}
\]

\[\text{snd}[K_\uparrow \{ O \} \equiv K_\uparrow \{ \text{snd}(O) \}
\]

\[\text{O}_\uparrow \{ O \} \equiv K_\uparrow \{ O \}
\]

\[\text{M}_\uparrow \{ O \} \equiv \text{(\lambda}_n x.S_\uparrow(x)) \{ O \}
\]

\[\text{N}_\uparrow \{ O \} \equiv \text{O}_\uparrow \{ O \}
\]

This notation satisfies the following property.

**Theorem 28 (Soundness of \((-)_\uparrow \) )**

1. If \( DC \vdash M \equiv_n L \), then \( \lambda \mu \vdash M_\uparrow \equiv_n L_\uparrow \).
   \( \square \)

5.3 The Translation from CBV Dual Calculus into CBV \( \lambda \mu \)-calculus

Now we introduce the modified translation \((-)_\uparrow \) for the call-by-value calculus. The problematic cases of the naive translation were \( (\eta_L) \) and \( (\beta_-) \)-rules. To solve these problems, we introduce a notation: \((\lambda_n.x.S)O\). This means \( S^{O/\alpha}_n \) if \( O \) is a value, otherwise \( (\lambda x.S)O \). Our idea for solving the latter case is to modify the definition of \( M \circ K \) to \( (\lambda_n x.K_\uparrow(x))M_\uparrow \).

**Definition \((\lambda)_\uparrow : \text{CBV DC} \rightarrow \text{CBV } \lambda \mu \)**

We define the translation \((-)_\uparrow \) as follows:

\[
\text{K}_\uparrow \{ O \} \equiv \text{(\lambda}_n x.K_\uparrow(x)) \{ O \}
\]

\[\text{fst}[K_\uparrow \{ O \} \equiv K_\uparrow \{ \text{fst}(O) \}
\]

\[\text{snd}[K_\uparrow \{ O \} \equiv K_\uparrow \{ \text{snd}(O) \}
\]

\[\text{O}_\uparrow \{ O \} \equiv K_\uparrow \{ O \}
\]

\[\text{M}_\uparrow \{ O \} \equiv \text{(\lambda}_n x.S_\uparrow(x)) \{ O \}
\]

\[\text{N}_\uparrow \{ O \} \equiv \text{O}_\uparrow \{ O \}
\]

This notation satisfies the following property.

**Theorem 28 (Soundness of \((-)_\uparrow \) )**

1. If \( DC \vdash M \equiv_n L \), then \( \lambda \mu \vdash M_\uparrow \equiv_n L_\uparrow \).
   \( \square \)
(2) If $K : A \mid \Gamma \vdash_{dc} \Delta$ and $\Gamma \vdash_{\lambda \mu} \Delta \mid O : A$, then $\Gamma \vdash_{\lambda \mu} \Delta \mid K \{O\}$.

(3) If $\Gamma \mid S \vdash_{dc} \Delta$, then $\Gamma \vdash_{\lambda \mu} \Delta \mid S\{\}$.

**Proof.** These claims can be shown by the subject reduction property for the $\lambda \mu$-calculus using Proposition 23 (1) and Lemma 30.

**Lemma 32**

Let $O$ and $O'$ be terms of the $\lambda \mu$-calculus, $M$ and $N$ be terms, $V$ be a value, $K$ and $L$ be coterms, and $S$ be a statement of the dual calculus.

(1) If $\lambda \mu \vdash O \rightarrow_v O'$, then $\lambda \mu \vdash K\{O\} \rightarrow_v^* K\{O'\}$.

(2) $M\{V/x\} \equiv (M[\lambda y.L]\{\} \{V[y/x]\})\{\}$, $(K\{O\})\{V/x\} \equiv (K[\lambda y.L]\{\} \{V[y/x]\})\{O\}$, and $(K\{O\})\{V/x\} \equiv (K[\lambda y.L]\{\} \{V[y/x]\})\{O\}$, and $S\{V/x\} \equiv (S[\lambda y.L]\{\} \{V[y/x]\})\{\}$.

**Proof.** See Appendix.

**Lemma 33**

Let $O$ be a term of the $\lambda \mu$-calculus, $M$ be a term, $K$ and $L$ be coterms, and $S$ be a statement of the dual calculus. Then

$$\lambda \mu \vdash M\{\} \equiv (M[\lambda y.L]\{\} \{\}) \rightarrow_v^* (M[\lambda y.L]\{\} \{\})\{\},$$

$$\lambda \mu \vdash K\{O\} \equiv (K[\lambda y.L]\{\} \{\}) \rightarrow_v^* (K[\lambda y.L]\{\} \{\})\{\},$$

and

$$\lambda \mu \vdash S\{\} \equiv (S[\lambda y.L]\{\} \{\}) \rightarrow_v^* (S[\lambda y.L]\{\} \{\})\{\}.$$

**Proof.** See Appendix.

We now prove that the call-by-name translation $\rightarrow^c_\downarrow$ preserves reductions.

**Theorem 34 (Soundness of $\rightarrow^c_\downarrow$)**

(1) If $DC \vdash M \rightarrow_v^c N$, then $\lambda \mu \vdash M \rightarrow_v^c N\{\}$. 

(2) If $DC \vdash K \rightarrow_v^c L$, then $\lambda \mu \vdash K\{\} \rightarrow_v^c L\{\}$. 

(3) If $DC \vdash S \rightarrow_v^c T$, then $\lambda \mu \vdash S\{\} \rightarrow_v^c T\{\}$.

Moreover, in (1), (2), and (3), if $\rightarrow_v^c$ is $\rightarrow^c_{\beta}$, $\rightarrow^c_{\beta C}$, or $\rightarrow^c_{\beta c}$, then $\rightarrow_v^c$ can be replaced by $\rightarrow_v^c$.

**Proof.** See Appendix.

### 5.4 Reloading Property

Wadler (2005) showed that the compositions of his translations $\lambda \mu \rightarrow dual \rightarrow \lambda \mu$ and $dual \rightarrow \lambda \mu \rightarrow dual$ reload into the $\lambda \mu$-calculus and the dual calculus respectively. That is, they become identity maps up to the call-by-name/call-by-value equalities. When we consider the composition of our translations, we can correspond corresponding results as follows:

- a reloaded term by the call-by-name modified translations reduced from the original term by the call-by-name reductions (Proposition 35 (1), 36 (1)), and
- a reloaded term by the call-by-value modified translations reduced from the original term by the call-by-value reductions (Proposition 35 (2), 36 (2)).

**Proposition 35 (Reloading Property (1))**

Let $O$ be a term and $S$ be a statement of the $\lambda \mu$-calculus. Then

(1) $\lambda \mu \vdash S \rightarrow_v^c (S\{\})\{\}$.

**Proof.** See Appendix.

**Proposition 36 (Reloading Property (2))**

Let $M$ be a term, $K$ be a coterm, and $S$ be a statement of the dual calculus. Then

(1) $DC \vdash M \rightarrow_v^c (M\{\} \{\})\{\}$.

**Proof.** See Appendix.

We can obtain the Church-Rosser property for the $\lambda \mu$-calculus by using the Church-Rosser property for the dual calculus and the results in Section 4 and 5.

**Proposition 37 (Church-Rosser Property for $\lambda \mu$)**

(1) If $\lambda \mu \vdash O \rightarrow_v^c M$ and $\lambda \mu \vdash O \rightarrow_v^c M'$, then there exists a term $O'$ such that $\lambda \mu \vdash O \rightarrow_v^c O'$ and $\lambda \mu \vdash M' \rightarrow_v^c O'$.

(2) If $\lambda \mu \vdash O \rightarrow_v^c M$ and $\lambda \mu \vdash O \rightarrow_v^c M'$, then there exists a term $O'$ such that $\lambda \mu \vdash M \rightarrow_v^c O'$ and $\lambda \mu \vdash M' \rightarrow_v^c O'$.

**Proof.** See Appendix.

### 6. Duality between Call-by-name and Call-by-value

Duality is the essential feature of the dual
Duality for the dual calculus

- Types: disjunction is dual to conjunction, and negation is self-dual,
- Expressions: terms are dual to coterms, and statements are self-dual,
- Typing rules: right rules are dual to left rules, and cut is self-dual, and
- Evaluation strategies: call-by-value is dual to call-by-name.

In this section, following Wadler’s approach, we discuss the systems that do not involve implication, since duality is not defined for implication.

The duality translation from the dual calculus to itself is given as follows.

**Duality for the dual calculus**

$$(X)^\circ \equiv X$$

$$(A \& B)^\circ \equiv B^\circ \lor A^\circ$$

$$(A \lor B)^\circ \equiv B^\circ \& A^\circ$$

$$(-A)^\circ \equiv \neg A^\circ$$

$$(\langle M, N \rangle)^\circ \equiv \langle H. M^\circ, M^\circ \rangle$$

$$\langle M \text{inl}\rangle^\circ \equiv \text{snd}(M^\circ)$$

$$\langle N \text{inr}\rangle^\circ \equiv \text{fst}(N^\circ)$$

$$(S.\alpha)^\circ \equiv x. S^\circ \ (x.S)^\circ \equiv S^\circ.x$$

**Proposition 38 (Duality for the Dual Calculus)**

**(Involution)** Duality is an involution, that is, $A^{\circ\circ} = A$, $M^{\circ\circ} = M$, $K^{\circ\circ} = K$, and $S^{\circ\circ} = S$.

**(Expressions and typing rules)**

(a) For any term $M$ of the dual calculus, $M^\circ$ is a coterm.

If $M$ has type $A$, then $M^\circ$ also has type $A$, i.e.,

$$\Gamma \vdash \Delta \mid M : A \text{ implies } M^\circ : A \mid \Delta^\circ \vdash \Gamma^\circ,$$

where $\Gamma$ is $x_m : A_m, \ldots, x_1 : A_1^\circ$ for $\Gamma^\circ$ is $x_1 : A_1, \ldots, x_m : A_m$, and $\Delta^\circ$ is $\alpha_n : B_n^\circ, \ldots, \alpha_1 : B_1^\circ$ for $\Delta \equiv \alpha_1 : B_1, \ldots, \alpha_n : B_n$.

(b) For any coterm $K$ of the dual calculus, $K^\circ$ is a term.

If $K$ has type $A$, then $K^\circ$ also has type $A$, i.e.,

$$K : A \mid \Gamma \vdash \Delta \text{ implies } \Delta^\circ \vdash \Gamma^\circ \mid K^\circ : A.$$

(c) For any statement $S$ of the dual calculus, $S^\circ$ is also a statement, and

$$\Gamma \mid S \vdash \Delta \text{ implies } \Delta^\circ \mid S^\circ \vdash \Gamma^\circ.$$

**(Evaluation strategies)**

(a) If $DC \vdash M \longrightarrow_n N$, then $DC \vdash M^\circ \longrightarrow_n N^\circ$.

If $DC \vdash K \longrightarrow_n L$, then $DC \vdash K^\circ \longrightarrow_n L^\circ$.

If $DC \vdash S \longrightarrow_n T$, then $DC \vdash S^\circ \longrightarrow_n T^\circ$.

(b) If $DC \vdash M \longrightarrow^v N$, then $DC \vdash M^\circ \longrightarrow^v N^\circ$.

If $DC \vdash K \longrightarrow^v L$, then $DC \vdash K^\circ \longrightarrow^v L^\circ$.

If $DC \vdash S \longrightarrow^v T$, then $DC \vdash S^\circ \longrightarrow^v T^\circ$.

Wadler (2005) gave a translation between the call-by-name and call-by-value $\lambda\mu$-calculi by composing the translation, $(-)^\circ$, and his translations between the dual calculus and the $\lambda\mu$-calculus. He explained duality between the call-by-name $\lambda\mu$-calculus and the call-by-value $\lambda\mu$-calculus by purely syntactic techniques. We follow his approach. Since we gave the different translations for call-by-name and call-by-value in the previous sections, we introduce two distinct translations between the call-by-name and call-by-value $\lambda\mu$-caluli.

**Definition** $(-)_o : CBN \lambda\mu \rightarrow CBV \lambda\mu$

Let $A$ be a type, and $M$ and $O$ be terms, and $S$ be a statement of the $\lambda\mu$-calculus. Then we define the translation $(-)_o$ as follows.

$$(A)_o \equiv A^\circ$$

$$M_o\{O\} \equiv ((M^\circ)_o)_{_{\circ \circ}}\{O\}$$

$$S_o \equiv ((S^\circ)_o)_{_{\circ \circ}}$$

**Definition** $(-)_\bullet : CBV \lambda\mu \rightarrow CBN \lambda\mu$

Let $A$ be a type, and $M$ and $O$ be terms, and $S$ be a statement of the $\lambda\mu$-calculus. Then we define the translation $(-)_\bullet$ as follows.

$$(A)_\bullet \equiv A^\circ$$

$$M_\bullet\{O\} \equiv ((M^\circ)_\bullet)_{_{\circ \circ}}\{O\}$$

$$S_\bullet \equiv ((S^\circ)_\bullet)_{_{\circ \circ}}$$

The following properties of these translations are easily shown.

**Proposition 39**

(1) For any term $M$ of the $\lambda\mu$-calculus, $M_o\{O\}$ and $M_\bullet\{O\}$ are statements of the $\lambda\mu$-calculus. For any statement $S$ of the $\lambda\mu$-calculus, $S_o$ and $S_\bullet$ are statements of the $\lambda\mu$-calculus.

(2) If $\Gamma \vdash \lambda\mu^\circ \mid M : A \text{ and } \Delta_o \vdash \lambda\mu D_o \mid O : A_o$, then $\Gamma \mid M_o\{O\} \vdash \lambda\mu \Delta_o$.

If $\Gamma \mid S \vdash \lambda\mu D_o$, then $\Gamma \mid S_o \vdash \lambda\mu D_o$.

(3) If $\Gamma \vdash \lambda\mu^\circ \mid M : A \text{ and } \Delta_\bullet \vdash \lambda\mu O_\bullet \mid O : A_\bullet$, then $\Gamma \mid M_\bullet\{O\} \vdash \lambda\mu \Delta_\bullet$.

If $\Gamma \mid S \vdash \lambda\mu D_\bullet$, then $\Gamma \mid S_\bullet \vdash \lambda\mu D_\bullet$.

Then, we obtain our final results.

**Theorem 40**

Let $M$, $N$, and $O$ be terms, and $S$ and $T$ be statements. Then the following hold.

(1) The translation $(-)_o$ preserves reductions.

$$\lambda\mu \vdash M \longrightarrow_n N \text{ implies } \lambda\mu \vdash M_o\{O\} \longrightarrow_n N_o\{O\}$$

$$\lambda\mu \vdash S \longrightarrow_n T \text{ implies } \lambda\mu \vdash S_o \longrightarrow_n T_o$$
(2) The translation \((-)_\bullet\) preserves reductions.  
\[
\lambda_\mu \vdash M \longrightarrow_v N \text{ implies } \lambda_\mu \vdash M_\bullet \{O\} \longrightarrow_n N_\bullet \{O\}
\]
\[
\lambda_\mu \vdash S \longrightarrow_v T \text{ implies } \lambda_\mu \vdash S_\bullet \longrightarrow_n T_\bullet
\]

(3) The composition of translations obtained by applying \((-)_\bullet\) after \((-)_\circ\) is identity up to the call-by-name reductions.  
\[
\lambda_\mu \vdash M \longrightarrow_v \mu_\alpha.\{M_\circ \{\alpha\}\}_\bullet
\]
\[
\lambda_\mu \vdash O_\circ \{M\} \longrightarrow_n \{M_\bullet \{O\}\}_\circ
\]
\[
\lambda_\mu \vdash S \longrightarrow_v (S_\circ)_\circ
\]

(4) The composition of translations obtained by applying \((-)_\circ\) after \((-)_\bullet\) is identity up to the call-by-value reductions.  
\[
\lambda_\mu \vdash M \longrightarrow_v \mu_\alpha.\{M_\bullet \{\alpha\}\}_\circ
\]
\[
\lambda_\mu \vdash O_\bullet \{M\} \longrightarrow_n \{M_\circ \{O\}\}_\circ
\]
\[
\lambda_\mu \vdash S \longrightarrow_v (S_\bullet)_\circ
\]

Proof. See Appendix. \(\square\)

Although Wadler gave the same translation which goes back and forth between the call-by-name and call-by-value \(\lambda\mu\)-calculi, we needed two different translations. However, although Wadler’s translation preserved only equations, our translations preserve reductions. This is the greatest advantage of our results.

Acknowledgments I thank Philip Wadler for the very fruitful discussions. I also thank Makoto Tatsuta, Kazushige Terui, and Makoto Kanazawa for very helpful comments and discussions on this work.

References

1) de Groote, P.: A CPS-translation of the \(\lambda\mu\)-calculus, Proc. 19th Intl. Coll. on Trees in Algebra and Programming, CAAP’94, Edinburgh, UK, 11–13 Apr. 1994, Vol.787, pp.85–99, Berlin, Springer-Verlag (1994).

2) de Groote, P.: Strong normalization of classical natural deduction with disjunction, 5th International Conference on Typed Lambda Calculi and Applications, TLCA’01, Lecture Notes in Computer Science, Vol.2044, pp.182–196, Springer-Verlag (2001).

3) Felleisen, M., Friedman, D.P., Kohlbecker, E. and Duba, B.: A syntactic theory of sequential control, Th. Comp. Sci., Vol.52, No.3, pp.205–237 (1987).

4) Filinski, A.: Declarative continuations and categorical duality, Master’s thesis, Univ. of Copenhagen (1989).

5) Griffin, T.G.: A formulae-as-types notion of control, Proc. 1990 Principles of Programming Languages Conference, pp.47–58, IEEE Computer Society Press (1990).

6) Ong, C.H.L. and Stewart, C.A.: A curry-howard foundation for functional computation with control, Proc. Symposium on Principles of Programming Languages, pp.215–227 (1997).

7) Parigot, M.: \(\mu\)-calculus: An algorithmic interpretation of classical natural deduction, Proc. International Conference on Logic Programming and Automated Deduction, Lecture Notes in Computer Science, Vol.624, pp.190–201, Springer-Verlag (1992).

8) Sabry, A. and Felleisen, M.: Reasoning about programs in continuation-passing style, Lisp and Symbolic Computation, Vol.6, No.3/4, pp.289–360 (1993).

9) Selinger, P.: Control categories and duality: on the categorical semantics of the lambda-mu calculus, Mathematical Structures in Computer Science, pp.207–260 (2001).

10) Wadler, P.: Call-by-Value is Dual to Call-by-Name, International Conference on Functional Programming, Uppsala, Sweden, pp.25–29 (2003).

11) Wadler, P.: Call-by-Value is Dual to Call-by-Name — Reloaded, Rewriting Techniques and Applications, Nara, Japan, pp.185–203, Springer (2005).

Appendix

A.1 The Full Definitions of Translations \((-)_\circ\) and \((-)_\bullet\)

The full definition of the translation \((-)_\circ\) from the call-by-name dual calculus into the call-by-name \(\lambda\mu\)-calculus is displayed in Fig. 8, and the full definition of the translation \((-)_\bullet\) from the call-by-value dual calculus into the call-by-value \(\lambda\mu\)-calculus is displayed in Fig. 9.

A.2 Proofs

Proof of Lemma 1 (1) is easily shown by induction on \(E_n\) and \(D_n\). (2) is shown by induction on \(M\) and \(S\) using (1). We give the key case.

\[
\langle\langle [\alpha]M\rangle\rangle [D_n(-)/[\alpha][-]]
\]
\[
\equiv \langle\langle [\alpha]M\rangle\rangle [D_n(-)/[\alpha][-]]
\]
\[
\equiv D_n\{ \langle\langle [\alpha]M\rangle\rangle [D_n(-)/[\alpha][-]] \}
\]
\[
=_{I.H.} D_n\{ \langle\langle [\alpha]M\rangle\rangle [D_n(-)/[\alpha][-]] \}
\]
\[
=_{(1)} D_n\{ \langle\langle [\alpha]M\rangle\rangle [D_n(-)/[\alpha][-]] \}
\]
\[
=_{(1)} D_n\{ \langle\langle [\alpha]M\rangle\rangle [D_n(-)/[\alpha][-]] \}
\]

(3) is also shown by induction on \(E_n\) and \(D_n\). For example, the case of \(\delta(E_n,x.D_n(x),y.D_n(y))\) can be proved as follows.
\[
\begin{align*}
(x)_2 & \equiv x \\
(M, N)_2 & \equiv (M_2, N_2) \\
(M, \text{inl})_2 & \equiv \text{inl}(M_2) \\
(N, \text{inr})_2 & \equiv \text{inr}(N_2) \\
(K, \text{not})_2 & \equiv \lambda x.K_2(x) \\
(x, M)_2 & \equiv \lambda x.M_2 \\
(S, \alpha)_2 & \equiv \mu \alpha.S_2 \\
(M \bullet K)_2 & \equiv K_2[M_2]
\end{align*}
\]

\textbf{Fig. 8} Translation \((-)_2\) : CBN dual calculus \(\longrightarrow\) CBN \(\lambda \mu\)-calculus.

\[
\begin{align*}
\delta(E_n, x.D_n(x), y.D_n(y))\{\mu \alpha.S\} \\
& \equiv D_n(\mu \beta.D_n\{\mu \alpha.\beta.E_n\{\mu \alpha.S\}\}) \\
& \equiv L_n[D_n(\mu \beta.D_n\{\mu \alpha.\beta.E_n\{\mu \alpha.S\}\})] \\
& \equiv \zeta_n[D_n(\mu \beta.D_n\{\mu \alpha.\beta.E_n\{\mu \alpha.S\}\})] \\
& \equiv S[D_n(\mu \beta.D_n\{\mu \alpha.\beta.E_n\{\mu \alpha.S\}\})] \\
& \equiv \delta(E_n, x.D_n(x), y.D_n(y))\{\mu \alpha.S\}
\end{align*}
\]

\textbf{Fig. 9} Translation \((-)_1\) : CBV dual calculus \(\longrightarrow\) CBV \(\lambda \mu\)-calculus.

\[
\begin{align*}
\equiv \mu \gamma.\mu(\lambda x.[\gamma]\langle E_n'(x)\rangle) \\
(\mu \alpha.\beta)[\mu(\lambda \alpha.\beta')(\alpha\gamma.\langle O\rangle)] \\
= \beta_n.\mu \gamma.\mu(\lambda x.[\gamma]\langle E_n'(x)\rangle) \\
= \gamma_n.\mu \gamma.\mu(\lambda x.[\gamma]\langle E_n'(x)\rangle) \\
= \mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
(\mu \beta.\lambda x.[\gamma]\langle E_n'(x)\rangle) \langle O\rangle \\
(\text{by Lem 1 (1)}) \\
= \beta_n.\mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
= \gamma_n.\mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
(\text{by Lem 1 (3)}) \\
= \mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
= \mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
(\text{by Lem 1 (1)})
\end{align*}
\]
where \(M\) is an abbreviation of \((\lambda x.[\gamma]\langle E_n'(x)\rangle)\). The other rules of \((\beta_n)\)-rule can also be shown similarly.

Case of \((\zeta)\)-rule :
\[
\langle E_n(\mu \alpha.S) \rangle = \mu \gamma.\mu(\lambda x.[\gamma]\langle E_n'(x)\rangle) \\
(\mu \alpha.\beta)[\mu(\lambda \alpha.\beta')(\alpha\gamma.\langle O\rangle)] \\
(\text{by Lem 1 (1)}) \\
= \mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
(\mu \beta.\lambda x.[\gamma]\langle E_n'(x)\rangle) \langle O\rangle \\
(\text{by Lem 1 (3)}) \\
= \mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
(\mu \beta.\lambda x.[\gamma]\langle E_n'(x)\rangle) \langle O\rangle \\
(\text{by Lem 1 (2)}) \\
= \mu \gamma.\lambda x.[\gamma]\langle E_n'(x)\rangle \langle O\rangle \\
(\mu \beta.\lambda x.[\gamma]\langle E_n'(x)\rangle) \langle O\rangle \\
(\text{by Lem 1 (1)})
\]
Case of $(\pi)$-rule:

$$\langle \delta(O, x.M, y.N) \rangle$$

$$=n \frac{\text{Lem} \mathfrak{L}_n}{\mathcal{E}_n} \{ \langle \delta(O, x.M, y.N) \rangle \}$$

$$= \mathcal{E}_n \{ \mu_\gamma.(\lambda.y.\gamma)[\langle N \rangle](\mu_\beta.(\lambda.x.\gamma)[\langle M \rangle]) (\mu_\alpha.[\alpha, \beta][\langle O \rangle]) \}$$

$$= \mathcal{E}_n \{ \mu_\gamma.(\lambda.y.\gamma)[\langle N \rangle](\mu_\beta.\langle \lambda.x.\gamma \rangle)[\langle M \rangle]) (\mu_\alpha.[\alpha, \beta][\langle O \rangle]) \}$$

The other rule of the $(\pi)$-rule can also be shown similarly.

Case of $(\nu)$-rule:

$$\langle \delta(O, x.S, y.T) \rangle \equiv (\lambda.y.\langle T \rangle)(\mu_\beta.(\lambda.x.\langle S \rangle) (\mu_\alpha.[\alpha, \beta][\langle O \rangle])$$

$$= n (\lambda.y.\langle T \rangle)(\mu_\beta.\langle \lambda.y.\langle T \rangle \rangle)(\mu_\beta'.(\lambda.x.\langle S \rangle))$$

$$= \mathcal{E}_n \{ \mu_\beta.(\lambda.x.\langle S \rangle)(\mu_\alpha.[\alpha, \beta][\langle O \rangle]) \}$$

$$= \mathcal{E}_n \{ \mu_\beta.\langle \lambda.x.\langle S \rangle \rangle(\mu_\alpha.[\alpha, \beta][\langle O \rangle]) \}$$

$$= \mathcal{E}_n \{ \mu_\beta.\langle \lambda.x.\langle S \rangle \rangle(\mu_\alpha.[\alpha, \beta][\langle O \rangle]) \}$$

(2) We can show this by induction on the call-by-name equation of the $\mu_\nu$-calculus. We consider only the rules about sums, i.e., $(\beta_\nu)$, $(\zeta_\nu)$, and $(\eta_\nu)$-rules.

$(\beta_\nu)$-rule:

$$[[[[\alpha', \beta']\mu_\nu(\alpha, \beta).S]]]$$

$$= \delta([[\mu_\nu(\alpha, \beta).S], x.[\alpha']x, y.[\beta']y])$$

$$= \delta(\mu_\gamma.\langle S \rangle[[\gamma]inl(-)/[\alpha](-), [\gamma]inr(-)/[\beta](-)], x.\langle \alpha' \rangle x, y.\langle \beta' \rangle y)$$

$$= \zeta [[S][\delta(inl(-), x.\langle \alpha' \rangle x, y.\langle \beta' \rangle y)/[\alpha](-), \delta(inr(-), x.\langle \alpha' \rangle x, y.\langle \beta' \rangle y)/[\beta](-)]$$

$$= \beta_\nu [[S][\alpha](-)/[\alpha](-), [\beta'](-)/[\beta](-)]$$

$$= [[S][\alpha'/\alpha, \beta'/\beta]]$$

$(\zeta_\nu)$-rule:

$$[[[\alpha, \beta]\mu_\nu(\gamma, S)]$$

$$= \delta([[\mu_\nu(\gamma, S), x.\langle \alpha \rangle x, y.\langle \beta \rangle y])$$

$$= \delta(\mu_\gamma.\langle S \rangle[[\gamma]inl(-)/[\alpha](-), [\gamma]inr(-)/[\beta](-)], x.\langle \alpha \rangle x, y.\langle \beta \rangle y)$$

$$= \zeta [[S][\delta(inl(-), x.\langle \alpha \rangle x, y.\langle \beta \rangle y)/[\gamma](-)]$$

$$= [[S][\alpha\beta](-)/[\gamma](-)]$$

$(\eta_\nu)$-rule:

$$[[[\mu_\nu(\alpha, \beta)]]]$$

$$= \delta([[\mu_\nu(\alpha, \beta)], x.\langle \alpha \rangle x, y.\langle \beta \rangle y)$$

$$= \delta(\mu_\gamma.\langle S \rangle[[\gamma]inl(-)/[\alpha](-), [\gamma]inr(-)/[\beta](-)], x.\langle \alpha \rangle x, y.\langle \beta \rangle y)$$

$$= [[S][\alpha\beta](-)/[\gamma](-)]$$

$$(\eta_\nu)$$-rule:

$$[[[\mu_\nu(\alpha, \beta)]]]$$

$$= \delta([[\mu_\nu(\alpha, \beta)], x.\langle \alpha \rangle x, y.\langle \beta \rangle y)$$

$$= \delta(\mu_\gamma.\langle S \rangle[[\gamma]inl(-)/[\alpha](-), [\gamma]inr(-)/[\beta](-)], x.\langle \alpha \rangle x, y.\langle \beta \rangle y)$$

$$= [[S][\alpha\beta](-)/[\gamma](-)]$$
Case of \( \text{inl}(M, x, [\alpha]x, y, [\beta]y) \):

\[
\equiv \mu_{\gamma}.\delta([\lambda y.\gamma]N, [\mu\alpha, [\alpha]](\lambda x.\gamma)[M]x, y, [\beta]y))
\]

The rest of the cases are shown similarly.

Case of \( \delta(O, x, S, y, T) \): this case is similar to the above case.

(4) We can show this by induction on term \( M \) and statement \( S \) of the \( \lambda \mu u \beta \)-calculus. We consider only \( [\alpha, \beta]S \) and \( \alpha, \beta \).
\[\equiv \mu \gamma. \tilde{N}(\mu \beta.(\lambda x.[\gamma]\langle M \rangle))\]
\[(\mu \alpha.[\alpha, \beta]\langle \text{inl}(V) \rangle)\]
\[\equiv \mu \gamma. \tilde{N}(\mu \beta.(\lambda x.[\gamma]\langle M \rangle))\]
\[(\mu \alpha.[\alpha, \beta]\mu(\alpha', \beta', \lambda^\prime\langle V \rangle))\]
\[= (\beta_a) \mu \gamma. \tilde{N}(\mu \beta.(\lambda x.[\gamma]\langle M \rangle))(\mu \alpha.[\alpha]\langle V \rangle)\]
\[= (\eta_a) \mu \gamma. \tilde{N}(\mu \beta.(\lambda x.[\gamma]\langle M \rangle))(\langle V \rangle)\]
\[= (\zeta) \mu \gamma.(\lambda^\prime \langle M \rangle)(\langle V \rangle)\]
\[= (\beta_a) \mu \gamma.[\gamma]\langle M \langle V \rangle/x \rangle\]
\[\equiv \mu \gamma.[\gamma]\langle M[V/x] \rangle\]
\[= \eta_a \mu \gamma.[\gamma]\langle N/V/x \rangle\]

where \(\tilde{N}\) is an abbreviation of \((\lambda y.[\gamma]\langle N \rangle)\).

\((*)\) comes from the claim: \(\langle M \rangle[[V]/x] \equiv \langle M[V/x] \rangle\) and \(\langle S \rangle[[V]/x] \equiv \langle S[V/x] \rangle\). This claim is shown by a straightforward induction on \(M\) and \(S\). The other rules of the \((\beta_a)\)-rule can also be shown similarly.

Case of \((\zeta)\)-rule : This case can be shown by a case analysis of the evaluation contexts. The key case is when \(E_e\) is \(\delta(\{\}, x.M, y.N)\) and \(D_e\) is \(\delta(\{\}, x.S, y.T)\).

Subcase of \(E_e\) is \(\delta(\{\}, x.M, y.N)\):
\[\langle \delta(\mu \alpha.S, x.M, y.N) \rangle\]
\[\equiv \mu \gamma.(\lambda y.[\gamma]\langle N \rangle)\]
\[= (\beta_a) \mu \gamma.(\lambda y.[\gamma]\langle N \rangle)(\mu \alpha.[\alpha', \beta'] \lambda y.[\gamma]\langle S \rangle)\]
\[= \zeta \mu \gamma. \tilde{N}(\mu \beta'. \tilde{N}(\mu \alpha'. \lambda y.[\gamma]\langle S \rangle)[\langle \alpha', \beta' \rangle/\{\alpha\}\{\{\} \}])\]
\[= (\eta_a) \mu \gamma. \tilde{N}(\mu \beta'. \tilde{N}(\mu \alpha'. \lambda y.[\gamma]\langle S \rangle)[\langle \alpha', \beta' \rangle/\{\alpha\}\{\{\} \}])\]
\[\equiv \mu \gamma.[\gamma]\langle S[V]/\{\alpha\}\{\{\} \} \rangle\]
\[\equiv \mu \gamma.[\gamma]\langle \delta(\{\}, x.M, y.N) \rangle\]
\[\equiv \mu \gamma.[\gamma]\langle \delta(\{\}, x.M, N) \rangle\]

where \(\tilde{M}\) and \(\tilde{N}\) are abbreviations of \((\lambda x.[\gamma]\langle M \rangle)\) and \((\lambda y.[\gamma]\langle N \rangle)\) respectively. \((*)\) is shown by a straightforward induction on terms and statements.

Case of \((\pi)\)-rule : We can obtain \(=_{\pi}\) from \([\alpha]\delta(O, x.M, y.N) =_{\pi} \delta(O, x, [\alpha]M, y, [\alpha]N)\) with \((\eta_a)\)-rule and \((\zeta)\)-rule in the following way. Let \(E\) be \(E_e\) or \((\lambda x.M)\{\{\} \} \).

\[E(\delta(O, x.M, y.N))\]
\[=_{\eta_a} E(\mu \alpha.[\alpha]\delta(O, x.M, y.N))\]
\[=_{\zeta} \mu \beta.\delta(O, x, [\beta]E\{M\}, y, [\beta]E\{N\})\]
\[=_{\zeta} \mu \beta.\beta'(O, x.E\{M\}, y.E\{N\})\]
\[=_{\eta_a} \delta(O, x.E\{M\}, y.E\{N\})\]

Similarly, we can obtain \(D_e(\delta(O, x.M, y.N)) =_{\pi} \delta(O, x.D_e\{M\}, y.D_e\{N\})\). Therefore, it is sufficient to prove \(\langle \delta(\{\}, x.M, y.N) \rangle =_{\pi} \langle \delta(O, x, [\alpha]M, y, [\alpha]N) \rangle\).

\[\langle \delta(\{\}, x.M, y.N) \rangle\]
\[\equiv \mu \gamma.(\lambda y.[\gamma]\langle N \rangle)(\mu \beta'.(\lambda x.[\gamma]\langle M \rangle))(\mu \alpha'. \lambda y.[\gamma]\langle O \rangle)\]
\[=_{\beta_a} \mu \gamma.(\lambda y.[\gamma]\langle N \rangle)\]
\[= \beta_a \mu \gamma.[\gamma]\langle O \rangle\]
\[\equiv \delta(O, x, [\alpha]M, y, [\alpha]N)\]

Case of \((\text{comp})\)-rule : To show this case, it is sufficient to have \(\langle \alpha \rangle(\langle \lambda x.(\lambda y.[\gamma]\langle N \rangle)\rangle \rangle =_{\pi} \langle \lambda x.(\lambda y.[\gamma]\langle M \rangle)\rangle \rangle\) by the discussion similar to that for \((\pi)\)-rule.

\[\langle \alpha \rangle(\langle \lambda x.(\lambda y.[\gamma]\langle M \rangle)\rangle \rangle\]
\[= \langle \alpha \rangle(\langle \lambda x.\lambda y.[\gamma]\langle M \rangle \rangle)\]
\[\equiv \langle \alpha \rangle. \langle \lambda x.(\lambda y.[\gamma]\langle M \rangle)\rangle \]

Case of \((\text{name})\)-rule : We can easily show \(\lambda \mu e \vdash \langle E_i\{O\} \rangle =_{\pi} \langle \lambda x.(\lambda y.[\gamma]\langle M \rangle)\rangle\) by a case analysis of \(E_i\). On the other hand, we have \(\langle \alpha \rangle O \equiv \alpha \langle O \rangle =_{\text{comp}} \langle \lambda x.[\alpha]x\rangle O \equiv \langle \lambda x.[\alpha]x\rangle O\). Therefore we obtain \(\lambda \mu e \vdash \langle E_i\{O\} \rangle =_{\pi} \langle \lambda x.(\lambda y.[\gamma]\langle M \rangle)\rangle\), since \(\lambda \mu e \vdash E_i\{O\} =_{\pi} \langle \lambda x.\lambda y.[\gamma]\langle M \rangle \rangle\). \(\lambda \mu e \vdash \langle \alpha \rangle O \equiv =_{\pi} \langle \lambda x.[\alpha]x\rangle O\) can be shown from \(\lambda \mu e \vdash \alpha \langle O \rangle \equiv =_{\pi} \langle \lambda x.[\alpha]x\rangle O\) with \((\zeta), (\eta_a)\) and \((\text{comp})\)-rules as follows.

\[E_e\{O\} =_{\eta_a} E_e[\mu \alpha.[\alpha]O]\]
\[=_{\pi} E_e[\mu \alpha.(\lambda x.[\alpha]x)O]\]
\[=_{\pi} \mu \beta.(\lambda x.[\beta]E_e\{x\})O\]
\[=_{\pi} \text{name} \mu \beta.(\lambda x.\lambda y.[\gamma]\langle M \rangle)\]
\[=_{\eta_a} \langle \lambda x.\lambda y.[\gamma]\langle M \rangle\rangle\]

Case of \((\eta_a)\)-rule : We can obtain \(=_{\pi}\) from \([\alpha]\delta(O, x.M, y.N) =_{\pi} \delta(O, x, [\alpha]M, y, [\alpha]N)\) with \((\eta_a)\)-rule and \((\zeta)\)-rule in the following way. Let \(E\) be \(E_e\) or \((\lambda x.M)\{\{\} \} \).

\[E(\delta(O, x.M, y.N))\]
\[=_{\eta_a} E(\mu \alpha.[\alpha]\delta(O, x.M, y.N))\]
\[=_{\zeta} \mu \beta.\delta(O, x, [\beta]E\{M\}, y, [\beta]E\{N\})\]
\[=_{\zeta} \mu \beta.\delta(O, x.E\{M\}, y.E\{N\})\]
\[=_{\eta_a} \delta(O, x.E\{M\}, y.E\{N\})\]

Similarly, we can obtain \(D_e(\delta(O, x.M, y.N)) =_{\pi} \delta(O, x.D_e\{M\}, y.D_e\{N\})\). Therefore, it is sufficient to prove \(\langle \delta(O, x, [\alpha]M, y, [\alpha]N) \rangle =_{\pi} \langle \delta(O, x, [\alpha]M, y, [\alpha]N) \rangle\).
\[ (\mu_{\beta}.(\lambda x. [\alpha_1, \beta_1] L_x) \overline{M}) =_{\beta_{\eta}} \mu (\alpha_1, \beta_1). (\lambda y. [\beta_1] y)(\mu_{\beta}.(\lambda x. [\alpha_1] x) \overline{M}) \]
\[ =_{\text{name}} \mu (\alpha_1, \beta_1). (\beta_1) \mu_{\beta} [\alpha_1] \overline{M} \]
\[ =_{\beta_{\nu}} \mu (\alpha_1, \beta_1). [\alpha_1, \beta_1] \langle \overline{M} \rangle \]
\[ =_{\nu_{\nu}} \langle \overline{M} \rangle \]

where \( \overline{M}, L_x, \) and \( R_y \) are abbreviations of \( (\mu_{\alpha}. [\alpha, \beta] \langle \overline{M} \rangle), (\langle \text{inr} \rangle(x)) \) and \( (\langle \text{inr} \rangle(y)) \) respectively.

(2) We can show this by induction on the call-by-name equation of the \( \Lambda_{\mu_\text{wad}} \)-calculus. We consider only the rules about sums, i.e., \((\beta_{\nu}), (\zeta), (\eta_{\nu}), (\text{name}), \) and \((\text{comp})\)-rules. Case of \((\beta_{\nu})\)-rule:
\[ [\alpha', \beta'] [\mu (\alpha, \beta). S] \]
\[ \equiv [\delta ([\mu (\alpha, \beta). S], x. [\alpha'] x, y. [\beta'] y)] \]
\[ \equiv [\delta (\mu_{\beta}. [\beta] y)] [\langle \text{inl} \rangle(-)/[\alpha_\eta(-)], [\gamma] \text{inr}(-)/[\beta(-)]] \]
\[ \equiv [\langle \zeta \rangle [S] \delta \langle \text{inl} \rangle(-)/[\alpha'(-)], [y. \beta' y)/[\beta(-)]] \]
\[ \equiv [\langle \zeta \rangle [S] \delta \langle \text{inl} \rangle(-)/[\alpha_\eta(-)], [\gamma] \text{inr}(-)/[\beta(-)]] \]
\[ \equiv [S' \alpha'/\alpha' \beta'/\beta] \]

(\#) can be shown by the following claim: \( \delta (\text{inl}(M), x. [\alpha] x, y. [\beta] y) = [\alpha] M \) for any \( M \).

We show this. If \( M \) is a value, then the claim is obtained by \((\beta_{\nu})\)-rule. If \( M \) is not a value, then
\[ \delta (\text{inl}(M), x. [\alpha] x, y. [\beta] y) \]
\[ =_{\text{name}} (\lambda z. \delta (z, x. [\alpha] x, y. [\beta] y)) \langle \text{inl} \rangle(M) \]
\[ =_{\text{comp}} (\lambda z'. (\lambda z. \delta (z, x. [\alpha] x, y. [\beta] y))) \langle \text{inl} \rangle(z') M \]
\[ =_{\beta_{\gamma}} (\lambda z'. [\lambda z] z') M =_{\text{name}} [\alpha] M \]

Case of \((\zeta)\)-rule: To show this case, we introduce evaluation singular context \( E_w \) and singular statement context \( D_w \) of the \( \Lambda_{\mu_\text{wad}} \)-calculus.

\[ E_w := \{ - \} N \mid V \{ - \} \mid \langle V, \{ - \} \rangle \mid \langle \{ - \}, M \rangle \]
\[ \mid \text{fst}(\{ - \}) \mid \text{snd}(\{ - \}) \]
\[ D_w := [\alpha] \{ - \} \mid [\alpha, \beta] \{ - \} \mid \{ - \} M \mid V \{ - \} \]

It is easily shown that if we have the following claims:

\[ \langle E_w [\mu_{\alpha}. S] \rangle =_{\nu} [\mu_{\beta}. S [\beta] E_w] [\alpha_\eta(-)] \]
\[ \langle D_w [\mu_{\alpha}. S] \rangle =_{\nu} [S' [\beta] \{ - \} [\alpha_\eta(-)] \]

then we can show this case. We can prove the claim by a case analysis of \( E_w \) and \( D_w \). We consider the key cases:

- \( E_w \) is \( x \{ - \} \):
\[ \langle x \mu_{\alpha}. S \rangle =_{\eta_{\nu}} (\lambda z. x z) \mu_{\alpha}. [S] \]
\[ =_{\zeta} \mu_{\beta}. [S] [\beta] x \{ - \} /[\alpha_\eta(-)] \]
\[ =_{\nu} \mu_{\beta}. [S] [\beta] x \{ - \} /[\alpha_\eta(-)] \]

(\#) can be shown by induction on terms and statements. The key case is proved as follows.

\[ \langle [\alpha] M \rangle [\beta] x \{ - \} /[\alpha_\eta(-)] \]
\[ =_{\langle [\alpha] M \rangle} [\beta] x \{ - \} /[\alpha_\eta(-)] \]
\[ =_{\langle [\alpha] M \rangle} [\beta] x \{ - \} /[\alpha_\eta(-)] \]

\( - D_w \) is \( x \{ - \} \): this can be shown in a way similar to the above case.

- \( D_w \) is \( [\alpha, \beta] \{ - \} \):
\[ \langle [\alpha, \beta] \mu_{\gamma}. S \rangle =_{\langle [\alpha, \beta] \mu_{\gamma}. S \rangle} [\beta] \mu_{\gamma}. [S] \]
\[ =_{\langle [\alpha, \beta] \mu_{\gamma}. S \rangle} [\beta] \mu_{\gamma}. [S] \]
\[ =_{\langle [\alpha, \beta] \mu_{\gamma}. S \rangle} [\beta] \mu_{\gamma}. [S] \]

(\#) is already shown in the proof of Proposition 2.

Case of \((\eta_{\nu})\)-rule:
\[ \langle [\mu_{\alpha}. \beta]. [\alpha, \beta] M \rangle \]
\[ =_{\mu_{\gamma}} [\alpha, \beta] M \]
\[ [\gamma] \text{inl}(-)/[\alpha_\eta(-)], [\gamma] \text{inr}(-)/[\beta(-)] \]
\[ =_{\mu_{\gamma}} [\lambda \gamma]. \delta ([\gamma] \text{inl}(x), y. [\gamma] \text{inr}(y)) \]
\[ =_{\text{(\#)}} [\lambda \gamma]. \delta ([\gamma] \text{inl}(x), y. [\gamma] \text{inr}(y)) \]
\[ =_{\text{(\#)}} [\lambda \gamma]. \delta ([\gamma] \text{inl}(x), y. [\gamma] \text{inr}(y)) \]

Case of \((\text{name})\)-rule: Let \( D \) be a statement context of the \( \Lambda_{\mu_\text{wad}} \)-calculus, then we can obtain \((\text{name})\)-rule of the \( \Lambda_{\mu_\text{wad}} \)-calculus from \( \Lambda_{\mu_\text{wad}} \) \( \langle [\lambda \gamma]. \text{inl}(x), y. [\gamma] \text{inr}(y) \rangle \) in the following way.

\[ D \{ M \} =_{\text{(\#)}} D \{ \mu_{\alpha}. [\alpha] M \} \]
\[ =_{\text{(\#)}} D \{ \mu_{\alpha}. (\lambda x. [\alpha] x) M \} \]

\[ =_{\text{(\#)}} (\lambda x. D \{ x \}) M \]
Therefore, it is sufficient to prove \((\lambda x.[\alpha]x[M])\) and statement \(S\) of the \(\lambda_{wad}^\mu\)-calculus. We consider only \(\mu(\alpha, \beta).S\) and \([\alpha, \beta]M\).

Case of \((\alpha, \beta)\): Let \(D\) be a term context of the \(\lambda_{wad}^\mu\)-calculus, then we can obtain \((\alpha, \beta)\)-rule of the \(\lambda_{wad}^\mu\)-calculus from \([\alpha]((\lambda x.M)N) = \mu(\lambda x.[\alpha]M)N\), \((\eta)\), and \((\zeta)\)-rule in the following way.

\[
D\{\lambda x.M.N\} =_\eta D\{\mu(\alpha, \beta)(\lambda x.M)N\}
\]

Therefore, it is sufficient to prove \([\alpha](\lambda x.M)N\] and \([\lambda x.D\{M\}]\) for \((\alpha, \beta)\) case.

(3) We can show this by induction on \(T\) and \(M\) of the \(\lambda_{wad}^\mu\)-calculus. We consider \(\text{inl}(M)\), \(\text{inr}(M)\), \(\text{inr}(\alpha, \beta, \gamma, \delta, \eta, \xi, \mu)\), and \(\text{inr}(\alpha, \beta, \gamma, \delta, \eta, \xi, \mu)\).

\[
\text{Case of } \text{inl}(M)\text{:}
\]

\[
\begin{align*}
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\end{align*}
\]

\[
\begin{align*}
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\end{align*}
\]

\[
\begin{align*}
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\end{align*}
\]

\[
\begin{align*}
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\mu(\alpha, \beta).\gamma & \equiv \mu(\alpha, \beta).\gamma \\
\end{align*}
\]

The other cases are shown similarly.

(4) We can show this by induction on \(T\) and \(M\) and \(\text{inr}(\alpha, \beta).\gamma\).
\(\gamma, y.([y] \text{inr} \circ \gamma)]\). Note that these equations are the \(DC^n\) equation, that is, Wadler’s system (2005).

Case of \(\text{inr}(O)\): this case is proved in a way similar to the above case.

Case of \(\delta(O, x, M, y, N)\):

\[
\langle\delta(O, x, M, y, N)\rangle^* = \langle\delta(O, x, M, y, N)\rangle^* = \langle\delta(O, x, M, y, N)\rangle^* = \langle\delta(O, x, M, y, N)\rangle^*
\]

where \(\hat{M}\) and \(\hat{N}\) are abbreviations of \(x.([M]^\gamma)\) and \(y.([N]^\gamma)\) respectively.

Case of \(\delta(O, x, S, y, T)\): this case is proved in a way similar to the above case.

(2) is also proved by an induction on \(M\) and \(S\). The key cases are terms and statements for sums, and these cases are shown in a way similar to (1).

**Proof of Lemma 12** We prove (1), (2), and (3) by a simultaneous induction on \(M, S\). If (1) is shown for some term \(M\), (2) of \(M\) is easily shown by: \(\langle\delta(O, x, M, y, N)\rangle^* = \langle\delta(O, x, M, y, N)\rangle^* = \langle\delta(O, x, M, y, N)\rangle^* = \langle\delta(O, x, M, y, N)\rangle^*\). Therefore we prove (1) and (3).

Case of \(x\): this case is immediate.

Case of \(\lambda x.M, \langle M, N \rangle, \text{inl}(O), \text{inr}(O)\): these cases are easily shown by the induction hypothesis of (2).

Case of \(MN\) (\(MN\) is a term, \(M\) is not a \(\lambda\)-abstraction): this case is also easily shown by the induction hypotheses of (1) and (2).

Case of \(\lambda x.S\): this case is easily shown by the induction hypothesis of (3).

Case of \(MN\) (\(MN\) is a statement, \(M\) is not a \(\lambda\)-abstraction): this case is also easily shown by the induction hypotheses of (1) and (2).

Case of \(\delta(O, x, S, y, T)\): this case is also easily shown by the induction hypotheses of (2) and (3).

Case of \([\alpha]M\): this case is also easily shown by the induction hypothesis of (1).
(3) this claim can be shown by an induction on term $O$.

Cases of $x, \lambda x.M, \lambda x.S, \mu \alpha.S, \langle M, N \rangle, \text{inl}(M)$, and $\text{inr}(M)$: these are easily shown by the definition.

Case of $MN$ ($MN$ is a term, $M$ is not a $\lambda$-abstraction):

$$MN : \gamma K$$

$$\equiv M : \gamma N^z @ K \xrightarrow{I.H.} M^z \bullet (N^z @ K)$$

$$\xrightarrow{\gamma \text{name}} (M : \gamma N^z @ (\alpha)).\alpha \bullet K$$

$$\equiv (MN) : \gamma (\alpha).\alpha \bullet K$$

Case of Case of $\langle \lambda x.M \rangle N:

$$\langle \lambda x.M \rangle N : \gamma K$$

$$\equiv \langle (\lambda x.M) N : \gamma \alpha \rangle . \alpha \bullet K$$

$$\equiv \langle (\lambda x.M) N^z \rangle : \gamma K$$

Case of $\text{fst}(O)$:

$$\text{fst}(O) : \gamma K$$

$$\equiv O : \gamma \text{fst}[K] \xrightarrow{I.H.} O^z \bullet \text{fst}[K]$$

$$\xrightarrow{\gamma \text{name}} (O^z \bullet \text{fst}[\alpha]).\alpha \bullet K$$

$$\equiv (\text{fst}(O) : \gamma \alpha \bullet K \equiv \text{fst}(O)^z \bullet K$$

Case of $\text{snd}(O)$: this case is shown in a way similar to the above case.

Case of $\delta(O, x.M, y.N)$:

$$\delta(O, x.M, y.N) : \gamma K$$

$$\equiv (O^z, [x.(M : \gamma \alpha), y.(N : \gamma \alpha)]).\alpha \bullet K$$

$$\equiv (\delta(O, x.M, y.N) : \gamma \alpha ).\alpha \bullet K$$

$$\equiv (\delta(O, x.M, y.N)^z \bullet K$$

(4) If $M$ is not a $\lambda$-abstraction, then the claim is immediately shown. We consider the remaining cases.

$$\langle \lambda x.M \rangle : (N^z @ P) \equiv \langle \lambda x.M^z \rangle \bullet (N^z @ P)$$

$$\xrightarrow{\gamma \beta_1} N^z \bullet x.(M^z \bullet P)$$

$$\equiv (\lambda x.S) : \text{not}(N^z) \equiv [x.S^z] \text{not} \bullet \text{not}(N^z)$$

$$\xrightarrow{\gamma \beta_1} N^z \bullet x.S^z \equiv ((\lambda x.S)N)^z$$

Proof of Lemma 15 (1) is proved by an induction on $E_n$ and $D_n$. For $E_n N$, we can prove the claim by the induction hypothesis, since $E_n M$ is not a $\lambda$-abstraction by the assumption of $M$. We now consider $\delta(E_n, x, E'_n \{x\}, y, E''_n \{y\})$ and $\delta(E_n, x, D_n \{x\}, y, D'_n \{y\})$.

Case of $\delta(E_n, x, E'_n \{x\}, y, E''_n \{y\})$:

$$\delta(E_n \{x\}, x, E'_n \{x\}, y, E''_n \{y\}) : \gamma P$$

$$\equiv E_n \{x\} : \gamma x.E'_n \{x\}, y, E''_n \{y\}$$

$$\xrightarrow{I.H.} E_n \{x\} : \gamma x.(E'_n \{x\} : \gamma P),$$

$$y.(E''_n \{y\} : \gamma P)$$

$\mathbf{Lem 14(2)}$

$$\equiv E_n \{x\} : \gamma \Phi(E'_n \{x\}, \Phi(E''_n \{y\}))$$

$$\equiv E_n \{x\} : \gamma \Phi(E'_n \{x\}, \Phi(E''_n \{y\}))$$

$$\xrightarrow{I.H.} E_n \{x\} : \gamma \Phi(E'_n \{x\}, \Phi(E''_n \{y\}))$$

$\mathbf{Lem 14(4)}$

$$\equiv E_n \{x\} : \gamma \Phi(E'_n \{x\}, \Phi(E''_n \{y\}))$$

This case is proved in a way similar to the above case.

The other cases are easily shown using the induction hypothesis.

(2) is proved by an induction on $E_n$ and $D_n$. If $E_n$ is $E_n N$, then

$$M : \gamma \Phi(E_n N, P) \equiv E_n \{x\} \Phi(E_n, N^z @ P)$$

$\mathbf{Lem 14(4)}$

$$\equiv E_n \{x\} \Phi(E_n, N^z @ P)$$

If $D_n$ is $E_n N$, this case is also proved by the induction hypothesis and Lemma 14(4). If $E_n$ is $\delta(E_n, x, E'_n \{x\}, y, E''_n \{y\})$, then

$$M : \gamma \Phi(\delta(E_n, x, E'_n \{x\}, y, E''_n \{y\}), P)$$

$$\equiv E_n \{x\} \Phi(\delta(E'_n \{x\}, \Phi(E''_n \{y\}))$$

$$\equiv E_n \{x\} \Phi(\delta(E'_n \{x\}, \Phi(E''_n \{y\}))$$

$\mathbf{I.H.}$

$$\equiv E_n \{x\} \Phi(\delta(E'_n \{x\}, \Phi(E''_n \{y\}))$$

This case is proved in a way similar to the above case.

The remaining cases are proved easily using the induction hypothesis.

(3) is proved by an induction on $M$ and $S$. The key case is $[\alpha] M$.

$$[\alpha] M : \gamma \Phi([\Phi(D_n)] / \alpha) \equiv [M : \gamma \alpha] \Phi([D_n] / \alpha)$$

$\mathbf{I.H.}$

$$\equiv [M : \gamma \alpha] \Phi([D_n] / \alpha)$$

If $D_n$ is $\delta(E_n, x, D_n \{x\}, y, D'_n \{y\})$, this case can be shown in a way similar to the above case.
Proof of Theorem 16

□

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.

The other cases are proved easily using the induction hypothesis.
The key case is:

\[ (4) \text{ can be shown by an induction on } E. \]

Case of \( \text{inr} (-) \): this case is proved in a way similar to the above case.

(2) can be shown by a case analysis of \( E \) and \( D_v \). We give the key case of \( D_v \) as follows:

\[ \lambda x . S : v \Psi(\{ \cdot \} N) \equiv \lambda x . S : v \text{ not}(N^\downarrow) \]

\[ \equiv [x . S^\dagger] \text{ not} \cdot \text{ not}(N^\downarrow) \rightarrow_v^\downarrow (\lambda x) N \cdot x . S^\dagger \]

For \( E \), the key case is: \( M \) is \( \lambda x . M \) and \( E \) is \( \{ \cdot \} N \). This case is shown in a way similar to the key case of \( D_v \).

(3) can be shown by a case analysis of \( E_i \).

Case of \( \{ \cdot \} N^\dagger \):

\[ (M : v \cdot \Psi(\{ \cdot \}, N, K)) \]

Case of \( \{ V, \cdot \} \):

\[ M : v \cdot \Psi(\{ V, \cdot \}, K) \equiv (M : v \cdot y . (V^\dagger, y) \cdot K) \]

Case of \( \text{inl} (-) \):

\[ M : v \cdot \Psi(\text{inl}(-), K) \equiv (M : v \cdot x . \langle x \rangle \cdot K) \]

The key case is:

\[ ([\alpha] M)[(\Psi(E, \beta)/\alpha)] \equiv (M : v \cdot \alpha)[(\Psi(E, \beta)/\alpha)] \]

\[ (\beta) : \text{if } (\lambda x . M) V : v K \equiv V : v x . (M : v K) \]

Case of \( \text{inr} (-) \): this case is proved in a way similar to the above case.

(4) can be shown by an induction on \( M \) and \( S \). The key case is:

\[ ([\alpha] M)[(\Psi(E, \beta)/\alpha)] \equiv (M : v \cdot \alpha)[(\Psi(E, \beta)/\alpha)] \]

\[ (\beta) : \text{if } (\lambda x . M) V : v K \equiv V : v x . (M : v K) \]

The other rule of \( (\beta) \) is proved similarly.

(\( \zeta \)) Let \( E_{\lambda} \) be an elimination context of \((\lambda x . M)\{ \cdot \}, \) then

\[ E_{\lambda} \{ \mu \alpha . S \} : v K \equiv \mu \alpha . S : v \Psi(E_{\lambda} \cdot K) \]

The other rule of \( (\zeta) \) is proved similarly.

(\( \pi \)) Let \( E_{\lambda} \{ \delta(O, x . M, y . N) \} : v K \]

The other rule of \( (\pi) \) is proved similarly.
\[\begin{align*}
\equiv O : v \rightarrow (x.(M : v \Psi(E_{e\lambda}, K)),
y.(N : v \Psi(E_{e\lambda}, K)))
\end{align*}\]

\[\begin{align*}
\text{Lemma 20}(2) \\
\rightarrow \text{rt} \quad O : v \rightarrow (x.(E_{e\lambda}\{M : v\} \rightarrow K),
y.(E_{e\lambda}\{N : v\} \rightarrow K))
\end{align*}\]

\[\begin{align*}
\equiv \delta(O, x.E_{e\lambda}\{M\} : y.E_{e\lambda}\{N\} : v) \rightarrow K
\end{align*}\]

The other rule of (\(\zeta\)) is proved similarly.

\(\eta_n\): \(\mu\alpha.\{M : v\} \rightarrow (\{\alpha\}M)\rightarrow [K/\alpha] \rightarrow (M : v)\alpha \rightarrow (K \rightarrow v) K
\]

\(\text{name}\): Let \(O\) not be a value. Then \(E_{e}\{O\} : v \rightarrow K
\]

\[\begin{align*}
\text{Lemma 20}(1) \\
\rightarrow \text{rt} \quad O : v \rightarrow (x.(E_{e}\{x\} : \Psi(E_{e}, K)), K)
\end{align*}\]

\[\begin{align*}
\rightarrow \text{rt} \quad O : v \rightarrow (x.(E_{e}\{x\} : \Psi(E_{e}, K)), K)
\end{align*}\]

Induction cases (1) and (2): These cases are similar to the proof for induction cases of call-by-name.

**Proof of Lemma 22** (1) is proved by an induction on \(M, K,\) and \(S\). We give the cases of sums, i.e., \((M)\mathrm{inl}, (N)\mathrm{inl},\) and \([K, L]\).

Case of \((M)\mathrm{inl}\):

\[\begin{align*}
\equiv (M)\mathrm{inl}. \equiv \mu\alpha.\{\alpha\}M \rightarrow (x.(\mu\alpha.\{\alpha\}M), (x.\mu\alpha.\{\alpha\}M))
\end{align*}\]

\[\begin{align*}
\equiv \mu\gamma.\{\gamma\}M \rightarrow (x.(\mu\gamma.\{\gamma\}M), (x.\mu\gamma.\{\gamma\}M))
\end{align*}\]

\[\begin{align*}
\equiv \mu\gamma.\{\gamma\}M \rightarrow (x.(\mu\gamma.\{\gamma\}M), (x.\mu\gamma.\{\gamma\}M))
\end{align*}\]

Case of \((\alpha\beta)\mathrm{inr}\): This case is proved in a way similar to the above case.

Case of \([K, L]\):

\[\begin{align*}
\equiv L_{e}\{\mu\beta.K\rightarrow \{\mu\alpha.\{\alpha\}O\}\}
\end{align*}\]

\[\begin{align*}
\equiv L_{e}\{\mu\beta.K\rightarrow \{\mu\alpha.\{\alpha\}O\}\}
\end{align*}\]

\[\begin{align*}
\equiv K_{e}\{\mu\alpha.\{\alpha\}O\}
\end{align*}\]

\[\begin{align*}
\equiv K_{e}\{\mu\alpha.\{\alpha\}O\}
\end{align*}\]

\[\begin{align*}
\equiv \delta(O, x.K_{e}\{x\}, y.L_{e}\{y\})
\end{align*}\]

\[\begin{align*}
\equiv [K, L]_{e}\{O\}
\end{align*}\]

\((\alpha\beta)\) comes from the claim: \(K_{e}\{\mu\alpha.S\} =_{\text{rt}} [K_{e}\{O\}]_{e}\{O\}\). This claim is proved by a straightforward induction on \(K\).

(2) is proved by an induction on \(M\) and \(S\). The key cases are terms and statements for sums, and these cases are shown in a way similar to (1).

**Proof of Theorem 28** The claims are proved by simultaneous induction on the reduction relation \(\rightarrow \) and \(\rightarrow^\ast\). Base cases are shown as follows.

\(\beta_2\): \(\delta(O, x.K_{e}\{x\}, y.L_{e}\{y\})
\]

\[\begin{align*}
\equiv P_{e}\{\mu\alpha.\{\alpha\}O\}
\end{align*}\]

\[\begin{align*}
\equiv (P_{e}\{\mu\alpha.\{\alpha\}O\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.K_{e}\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.M_{e}\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet\bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet\bullet\bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.M_{e}\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet\bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet\bullet\bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.M_{e}\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet\bullet P)\{x\})_{e}\{O\}
\end{align*}\]

\[\begin{align*}
\equiv (\lambda x.(M \bullet\bullet\bullet P)\{x\})_{e}\{O\}
\end{align*}\]
\[(\lambda x.(K_{\downarrow} [\text{fst}(x)])) O \equiv (\lambda x.([\text{fst}][K_{\downarrow}][x])) O \equiv \text{fst}[K_{\downarrow}][O]\]

Case of \(\text{snd}[K]_{\downarrow}\): This case can be shown in a way similar to the above case.

Case of \(\text{not}(M)\):
\[
\text{not}(M)_{\downarrow}\{O\} \equiv O M_{\downarrow} \longrightarrow_{(\text{name})} (\lambda z.M_{\downarrow}) O \\
\equiv (\lambda z.\text{not}(M)_{\downarrow}[z]) O \equiv \text{not}(M)_{\downarrow}[O]
\]

Case of \(M@K\):
\[
(M@K)_{\downarrow}\{O\} \equiv K_{\downarrow} [O M_{\downarrow}] \equiv (\lambda z.K_{\downarrow}[z])(O M_{\downarrow}) \\
\longrightarrow_{(\text{comp})} (\lambda x.(\lambda z.K_{\downarrow}[z])(x.M_{\downarrow})) O \\
\equiv (\lambda x.(K_{\downarrow}[xM_{\downarrow}]) O \\
\equiv (\lambda x.(M@K)_{\downarrow}\{x\}) O \equiv (M@K)_{\downarrow}[O]
\]

Case of \(x:S\):
\[
(x.S)_{\downarrow}\{O\} \equiv (\lambda x.S_{\downarrow}) O \equiv (\lambda z.S_{\downarrow}[\ast/x]) O \\
\equiv (\lambda z.(x.S_{\downarrow}[z]) O \equiv (x.S)_{\downarrow}[O] \quad \square
\]

**Proof of Lemma 30** The claims are shown by a simultaneous induction on \(M, K\), and \(S\). We consider the key cases.

Case of \(\text{fst}[K]_{\downarrow}\):
\[
\text{fst}[K_{\downarrow}][S] O \equiv K_{\downarrow}\{\text{fst}(O)\} \quad \text{I.H.} \\
\equiv (\lambda x.M_{\downarrow}) O \longrightarrow_{v}^{*} K_{\downarrow}\{\text{fst}(O)\}
\]

Case of \(\text{snd}[K]_{\downarrow}\): This case is proved similarly.

Case of \(M@K\):
\[
(M@K)_{\downarrow}\{O\} \equiv K_{\downarrow}\{O M_{\downarrow}\} \quad \text{I.H.} \\
\equiv (\lambda x.K_{\downarrow}[x]) O \equiv (M@K)_{\downarrow}[O]
\]

Case of \(x:S\):
\[
(x.S)_{\downarrow}\{O\} \equiv (\lambda x.S_{\downarrow}) O \quad \text{I.H.} \\
\equiv (\lambda x.S_{\downarrow}[x]) O \equiv (x.S)_{\downarrow}[O]
\]

Case of \(M \bullet K\):
\[
(M \bullet K)_{\downarrow}\equiv K_{\downarrow}\{M_{\downarrow}\} \quad \text{I.H.} \\
\equiv K_{\downarrow}[M_{\downarrow}] \equiv (M \bullet K)_{\downarrow} \quad \square
\]

**Proof of Lemma 32** (1) The claim is proved by induction on \(K\).

(2) We claim that if we have \((K_{\downarrow}[O])[x/v]\equiv (K[\uparrow/x])_{\downarrow}\{O[\uparrow/x]\}\), then we obtain \((K_{\downarrow}[O]) [V/x] \equiv (K[\uparrow/x])_{\downarrow}\{O[\uparrow/x]\}\). We show this claim. Assume \(O\) is a value, then \(O[\uparrow/x]\) is also a value. Hence we have
\[
(K_{\downarrow}[O]) [V/x] \equiv (K_{\downarrow}[O]) [V/x] \\
\equiv (K[\uparrow/x])_{\downarrow}\{O[\uparrow/x]\}
\]

If \(O\) is not a value, then \(O[\uparrow/x]\) is also not a value, so we have
\[
(K_{\downarrow}[O]) [V/x] \equiv ((\lambda z.K_{\downarrow}[z]) O)[V/x] \\
\equiv (\lambda z.(K_{\downarrow}[z])_{\downarrow}\{O[\uparrow/x]\}) \\
\equiv (\lambda z.(K[\uparrow/x])_{\downarrow}\{z\}) (O[\uparrow/x]) \\
\equiv (K[\uparrow/x])_{\downarrow}\{O[\uparrow/x]\}.
\]

Therefore we show the other claims by induction on \(M, K,\) and \(S\). We give the key case: \(x_{\downarrow}[V/x] \equiv x[V/x] \equiv V_{\downarrow}\). □

**Proof of Lemma 33** We can prove this lemma by a simultaneous induction on \(M, K,\) and \(S\). We give two cases.

Case of \(\alpha\):
\[
(\alpha_{\downarrow}[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\alpha[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\lambda y.L_{\downarrow}(y))[O[\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\longrightarrow_{\ast}^{*} (\lambda y.L_{\downarrow}(y))[O[\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv L_{\downarrow}[O[\lambda y.L_{\downarrow}(y)](-)[\alpha][-])
\]

Case of \(\text{fst}[K]_{\downarrow}\):
\[
((\text{fst}[K]_{\downarrow}[O]) [\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\text{fst}[O])[\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\text{fst}[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\text{fst}[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\text{fst}[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\text{fst}[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-]) \\
\equiv (\text{fst}[O][\lambda y.L_{\downarrow}(y)](-)[\alpha][-])
\]

**Proof of Theorem 34** (1) – (3) are proved by simultaneous induction on the reduction relation \(\longrightarrow^{*}\). We claim that if \(K_{\downarrow}[O] \longrightarrow^{*}_{v} L_{\downarrow}[O]\), then \(K_{\downarrow}[O] \longrightarrow^{*}_{v} L_{\downarrow}[O]\). We first show this claim. If \(O\) is a value, the claim is immediately shown. Otherwise, \(K_{\downarrow}[O] \equiv (\lambda x.K_{\downarrow}[x]) O \longrightarrow^{*}_{v} (\lambda x.L_{\downarrow}[x]) O \equiv L_{\downarrow}[O]\).

We often use the following shortcuts:

(a) \((V \bullet K)_{\downarrow}\equiv K_{\downarrow}[V_{\downarrow}] \equiv K_{\downarrow}[V_{\downarrow}]\)

(b) \((\lambda x.K_{\downarrow}[x])_{\downarrow} \equiv (\lambda x,K_{\downarrow}[x])_{\downarrow} \\
\equiv K_{\downarrow}[M_{\downarrow}] \equiv (M \bullet K)_{\downarrow}\)

(c) \((\lambda x.S_{\downarrow})_{\downarrow} \equiv (\lambda x.S_{\downarrow})_{\downarrow} \\
\equiv (\lambda x,S_{\downarrow})_{\downarrow} \equiv (x.S)_{\downarrow} \equiv (x.S)_{\downarrow}[M_{\downarrow}] \equiv (M \bullet x.S)_{\downarrow}\)

Base cases are shown as follows.

\((\beta_{\downarrow})\) \((\lambda x.M) \bullet (N@K)_{\downarrow}\)
\((a)\) \((N@K)\updownarrow\{(\lambda x. M)\}\)
\[\equiv K_{\updownarrow}\{(\lambda x. M)\}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. M_{\updownarrow})\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. M_{\updownarrow})\]

\((b)\)
\[\lambda x. (M_{\updownarrow}K)_{\updownarrow}\]
\[\equiv (\lambda x. (M_{\updownarrow}K)_{\updownarrow})N_{\updownarrow}\]
\[\equiv (\lambda x. (M_{\updownarrow}K)_{\updownarrow})N_{\updownarrow}\]

\((\beta_k)\)
\((\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]

\((\beta_R)\)
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]

\((\eta_L)\)
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]

\((\text{name})\):  Note that \(M_{\updownarrow}\) is not a value because \(M\) is not a value.
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]

\((\text{comp})\):  \(\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]

\((\text{inf})\)  \(\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]

\((\text{str})\)  \(\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[\equiv (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow (\lambda\alpha. K\uparrow\{\langle\alpha, x\rangle\})\uparrow N_{\updownarrow}\]
\[y'.(y.(N : n \cdot P))_z \{y'\}\]

\[\equiv [x.(M : n \cdot P), y.(N : n \cdot P)]_z (O)\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} [x.(M : n \cdot P), y.(N : n \cdot P)]_z ((O^2)_z]\]

\[\equiv (O^2 \cdot [x.(M : n \cdot P), y.(N : n \cdot P)]_z)\]

\[\equiv (\delta(O, x.M, y.N) : n \cdot P)_z\]

Case of \(\lambda x.M\):

\[P_\sharp \{\lambda x.M\} \overset{1.H}{\longrightarrow}^{\delta^*} P_\sharp \{\lambda x.(M^\sharp)_z\}\]

\[\equiv ((\lambda x.M^\sharp) \cdot P)_z \equiv (\lambda x.M : n \cdot P)_z\]

Case of \(MN(MN)\) is a term, and \(M\) is not a \(\lambda\)-abstraction:

\[P_\sharp \{MN\} \overset{1.H}{\longrightarrow}^{\delta^*} P_\sharp \{M(N^\sharp)_z\}\]

\[\equiv (N^\sharp@P)_z (M)\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} (M^\sharp : n \cdot (N^\sharp@P))_z \equiv (MN : n \cdot P)_z\]

Case of \((\lambda x.M)N\):

\[P_\sharp \{(\lambda x.M)N\} \overset{(\beta_\gamma)}{\longrightarrow} P_\sharp \{M[N/x]\}_z\]

\[\equiv (x.(P_\sharp \{M\}))_z (N)\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} (x.(M : n \cdot P))_z (N)\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} (x.(M : n \cdot P))_z ((N^\sharp)_z)\]

\[\equiv (N^\sharp \cdot x.(M : n \cdot P))_z\]

\[\equiv ((\lambda x.M)N : n \cdot P)_z\]

Case of \(\lambda x.S\):

\[P_\sharp \{(\lambda x.S)\} \overset{1.H}{\longrightarrow}^{\delta^*} P_\sharp \{(\lambda x.(S^\sharp)_z)\}\]

\[\equiv P_\sharp \{\lambda y.(x.S^\sharp)_z\} \equiv ([x.S^\sharp] \cdot \cdot P)_z\]

\[\equiv (\lambda x.S : n \cdot P)_z\]

Case of \(\mu x.S\):

\[P_\sharp \{\mu x.S\} \overset{1.H}{\longrightarrow}^{\delta^*} P_\sharp \{\mu x.(S^\sharp)_z\}\]

\[\overset{(\kappa)}{\longrightarrow} ((S^\sharp)_z P^\sharp[\alpha/\alpha] \cdot \cdot (-1))\]

\[\overset{\delta^*}{\longrightarrow} (S^\sharp[\cdot P/\alpha])_z \equiv (\mu x.S : n \cdot P)_z\]

Case of \([a]O\):

\[\equiv [a]O \equiv (\alpha^*_z \{O\})_z\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} (O : n \cdot \alpha)_z\]

\[\equiv ((\alpha_\sharp \{O\})_z)_z\]

Case of \(\delta(O, x.S, y.T)\):

\[\delta(O, x.S, y.T) \overset{1.H}{\longrightarrow}^{\delta^*} \delta(O, x.(S^\sharp)_z, y.(T^\sharp)_z)\]

\[\equiv \delta(O, x'.(x'.S^\sharp)_z \{x'\})\]

\[\overset{y'.(y'.(y'.T^\sharp)_z \{y'\})}{\longrightarrow}^{(\delta^* \cdot \cdot \cdot (O^2)_z)}\]

\[\equiv [x.S^\sharp, y.T^\sharp]_z \{O\}\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} [x.S^\sharp, y.T^\sharp]_z \{((O^2)_z)\}\]

\[\equiv (O^2 \cdot [x.S^\sharp, y.T^\sharp])_z\]

\[\equiv (\delta(O, x.S, y.T)^\sharp)_z\]

Case of \(MN(MN)\) is a statement, and \(M\) is not a \(\lambda\)-abstraction:

\[MN \overset{1.H}{\longrightarrow}^{\delta^*} M(N^\sharp)_z \equiv \text{not} (\langle N^\sharp \rangle_M^\sharp)\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} (M : n \cdot \text{not} (\langle N^\sharp \rangle_M^\sharp))_z \equiv ((MN)^\sharp)_z\]

Case of \((\lambda x.S)N\):

\[(\lambda x.S)N \overset{1.H}{\longrightarrow}^{\delta^*} (\lambda x.(S^\sharp)_z)N\]

\[\overset{(\beta_\gamma)}{\longrightarrow} (S^\sharp)_z [N/x]\]

\[\equiv (x.S^\sharp)_z [N] \overset{1.H}{\longrightarrow}^{\delta^*} (x.S^\sharp)_z ((N^\sharp)_z)\]

\[\equiv (N^\sharp \cdot x.S^\sharp)_z \equiv (((\lambda x.S)N)^\sharp)_z\]

(2) We can easily show \(O \overset{\delta^*}{\longrightarrow} (O^1)^\top\) from \(K_\top [O] \overset{\delta^*}{\longrightarrow} (O : v \cdot K)^\top\) in a way similar to the proof in (1). In the following, we prove the rest of (2) by a simultaneous induction on \(O\) and \(S\).

Case of \(x : K_\top [x] \equiv (x \cdot K)_\top \equiv (x : v \cdot K)_\top\)

Case of \((M, N)\):

\[K_\top [(M, N)] \overset{1.H}{\longrightarrow}^{\delta^*} K_\top [((M^\top)_\top, (N^\top)_\top)]\]

\[\equiv K_\top [(M^\top, N^\top)_\top] \equiv ((M, N)^\top \cdot K)_\top\]

\[\overset{\delta^*}{\longrightarrow} ((M, N) : n \cdot v)\]

Case of \(\text{fst}(O)\):

\[K_\top [\text{fst}(O)] \equiv \text{fst}[K_\top \{O\}^\top\]

\[\overset{1.H}{\longrightarrow}^{\delta^*} \text{fst}[K_\top \{O^1\}^\top\]

\[\overset{\text{Lem. 29}}{\longrightarrow}^{\delta^*} \text{fst}[K_\top \{O^1\}^\top \equiv (O^\top \cdot \text{fst}[K])_\top \equiv (O : v \cdot \text{fst}[K])_\top \equiv \text{fst}(O : v \cdot K)_\top\]

The case of \(\text{snd}(O)\) is shown similarly.

Case of \(\text{inl}(O)\):

\[K_\top [\text{inl}(O)] \overset{1.H}{\longrightarrow}^{\delta^*} K_\top [\text{inl}(O^\top)_\top]\]

\[\equiv K_\top [(O^\top)^\top] \equiv ((O^\top)^\top \cdot \text{inl})_\top \equiv (\text{inl}(O) : v \cdot K)_\top\]

The case of \(\text{inr}(O)\) is shown similarly.

Case of \(\delta(O, x.M, y.N)\):

\[K_\top [\delta(O, x.M, y.N)] \equiv (\lambda z.K_\top \{z\}) \delta(O, x.M, y.N) \overset{(\pi)}{\longrightarrow} \delta(O, x.(\lambda z.K_\top \{z\} : M), y.(\lambda z.K_\top \{z\}) : N) \overset{\delta^*}{\longrightarrow} \delta(O, x.K_\top [M], y.K_\top [N]) \overset{1.H}{\longrightarrow}^{\delta^*} \delta(O, x.(M : \cdot v \cdot K), y.(N : v \cdot K)) \overset{1.H}{\longrightarrow}^{\delta^*} \delta(O, x'.(x'.(M : \cdot v \cdot K))_\top \{x'\}, y'.(y'.(N : v \cdot K))_\top \{y'\})\]
\[
\equiv [x.(M : v \ K), y.(N : v \ K)]_{\{O\}}
\]

\[
\text{Lem 29}
\]
\[
\xrightarrow{v}^* [x.(M : v \ K), y.(N : v \ K)]_{\{O\}}
\]

\[
\text{I.H.}
\]
\[
\xrightarrow{v}^* (O :_v [x.(M : v \ K), y.(N : v \ K)])
\]
\[
\equiv (\delta(O, x.M, y.N) :_v \ K)_{\dagger}
\]

Case of \(\lambda x.M\):
\[
\text{I.H.}
\]
\[
K_\dagger[\lambda x.M] \rightarrow^v \star K_\dagger[\lambda x.(M^\dagger)]_{\dagger}
\]
\[
\equiv ((\lambda x.M^\dagger) \bullet K)_{\dagger} \equiv (\lambda x.M :_v K)_{\dagger}
\]

Case of \(MN(MN)\) is a term, and \(M\) is not a \(\lambda\)-abstraction:
\[
K_\dagger[MN] \rightarrow^v \star K_\dagger[M(N^\dagger)]_{\dagger}
\]
\[
\equiv (N^\dagger @ K)_{\dagger} \{ M \} \rightarrow^v (N^\dagger @ K)_{\dagger} \{ M \}
\]

\[
\text{I.H.}
\]
\[
\rightarrow^v (M :_v (N^\dagger @ K))_{\dagger} \equiv (MN :_v K)_{\dagger}
\]

Case of \((\lambda x.M)N\):
\[
K_\dagger[(\lambda x.M)N] \equiv (\lambda z.K_\dagger \{z\})(\lambda x.M)N
\]
\[
\rightarrow_{(\text{comp})} (\lambda x.(\lambda z.K_\dagger \{z\})M)N
\]
\[
\rightarrow^v (x.(\lambda x.M)_\dagger)N
\]

\[
\text{I.H.}
\]
\[
\rightarrow^v (\lambda x.(M :_v K)_{\dagger})N
\]

\[
\text{I.H.}
\]
\[
\rightarrow^v (N :_v x.(M :_v K))_{\dagger}
\]
\[
\equiv ((\lambda x.M)N :_v K)_{\dagger}
\]

Case of \(\lambda x.S\):
\[
K_\dagger[\lambda x.S] \rightarrow^v \star K_\dagger[\lambda x.(S^\dagger)]_{\dagger}
\]
\[
\equiv K_\dagger[\lambda y.(x.S^\dagger)]_{\dagger} \{ y \}
\]
\[
\equiv ([x.S^\dagger]_{\not \bullet} \ K)_{\dagger} \equiv (\lambda x.S :_v K)_{\dagger}
\]

Case of \(\mu x.S\):
\[
K_\dagger[\mu x.S] \rightarrow^v \star K_\dagger[\mu x.(S^\dagger)]_{\dagger}
\]
\[
\equiv (\lambda z.K_\dagger \{z\})_{\mu x.}(S^\dagger)_{\dagger}
\]
\[
\rightarrow_{(\eta)} (S^\dagger)_{\dagger} \{ (\lambda z.K_\dagger \{z\})_{/-\{x\}}/-\{x\} \}
\]

\[
\text{Lem 33}
\]
\[
\rightarrow^v (S^\dagger)_{\dagger} \{K / \alpha\}_{\dagger} \equiv (\mu x.S :_v K)_{\dagger}
\]

Case of \([\alpha]O\):
\[
\text{Lem 29}
\]
\[
[\alpha]O \equiv _v \alpha_{\dagger} \{O\}
\]
\[
\text{I.H.}
\]
\[
\rightarrow^v (O :_v \alpha)_{\dagger} \equiv (([\alpha]O)_{\dagger})_{\dagger}
\]

Case of \(\delta(O, x.S, y.T)\):
\[
\text{I.H.}
\]
\[
\delta(O, x.S, y.T) \rightarrow^v \delta(O, x.(S^\dagger), y.(T^\dagger))_{\dagger}
\]
\[
\equiv \delta(O, x'.((x.S^\dagger)_{\dagger} \{x'\}),
\]
\[
y' .((y.T^\dagger)_{\dagger} \{y'\}))
\]
\[
\equiv [x.S^\dagger, y.T^\dagger]_{\{O\}}
\]

\[
\text{Lem 29}
\]
\[
\xrightarrow{v}^* [x.S^\dagger, y.T^\dagger]_{\{O\}}
\]

\[
\text{I.H.}
\]
\[
\rightarrow^v (O :_v [x.S^\dagger, y.T^\dagger])_{\dagger}
\]
\[
\equiv (\delta(O, x.S, y.T^\dagger))_{\dagger}
\]

Case of \(MN(MN)\) is a statement, and \(M\) is not a \(\lambda\)-abstraction:
\[
MN \rightarrow^v \star M(N^\dagger)_{\dagger} \equiv not(N^\dagger)_{\dagger} \{ M \}
\]

\[
\text{I.H.}
\]
\[
\rightarrow^v (M :_v not(N^\dagger))_{\dagger} \equiv ((MN^\dagger)_{\dagger})_{\dagger}
\]

Case of \((\lambda x.S)N\):
\[
(\lambda x.S)N \rightarrow^v \star (\lambda x.(S^\dagger))N
\]
\[
\rightarrow^v (x.S^\dagger)_{\dagger}[N]
\]

\[
\text{I.H.}
\]
\[
\rightarrow^v (N :_v x.S^\dagger)_{\dagger} \equiv (((\lambda x.S)N^\dagger)_{\dagger})_{\dagger}
\]

\[
\square
\]

Proof of Proposition 36 (1) If we establish the following claims: (a) \(M \bullet Q \rightarrow^*_n (M_2 :_n Q);\)
(b) \(O^\dagger \bullet K \rightarrow^*_n K_\dagger\{O\}\) if \(K\) is not a covalue;
(c) \((O :_n P) \rightarrow^*_n P_\dagger\{O\}^\dagger);\) and (d) \(S \rightarrow^*_n (S_2)^\dagger),\) we can easily obtain (1). Therefore, we show these claims by a simultaneous induction on \(M, K, P,\) and \(S.\)

Case of \(x: \ x \bullet Q \equiv x :_n P\)

Case of \(\langle M\rangle\text{inl}::
\[
\text{I.H.}(a)
\]
\[
\langle M\rangle\text{inl} \bullet Q \rightarrow^*_n ((M_2)^\dagger)\text{inl} \bullet Q
\]
\[
\equiv \text{inl}(M_2) :_n Q \equiv \langle M\rangle\text{inl} :_n Q
\]

The case of \(\langle M\rangle\text{inr}\) is shown similarly.

Case of \(\langle M, N\rangle::
\[
\text{I.H.}(a)
\]
\[
\langle M, N\rangle \bullet Q \rightarrow^*_n ((M_2)^\dagger, (N_2)^\dagger) \bullet Q
\]
\[
\equiv ((M_2, N_2) :_n Q) \equiv ((M, N)_{\dagger} :_n Q)
\]

Case of \([K]\text{not}::
\[
[K]\text{not} \bullet Q \rightarrow^*_n ((M_2)^\dagger)\text{not} \bullet Q
\]
\[
\rightarrow^*_n [x.(x \bullet K)\text{not} \bullet Q
\]

\[
\text{I.H.}(b)
\]
\[
\rightarrow^*_n [x.(K_\dagger \{x\})^\dagger]\text{not} \bullet Q
\]
\[
\equiv (\lambda x.(K_\dagger \{x\}) \cdot :_n Q) \equiv ([K]\text{not} :_n Q)
\]

Case of \(\lambda x.M::
\[
\text{I.H.}(a)
\]
\[
(\lambda x.M) \bullet P \rightarrow^*_n (\lambda x.(M_2)^\dagger) \bullet P
\]
\[
\equiv (\lambda x.M_2 :_n P) \equiv ((\lambda x.M)_{\dagger} :_n P)
\]

Case of \(S\alpha::
\[
\text{I.H.}(d)
\]
\[
S\alpha \bullet Q \rightarrow^*_n (S_2)^\dagger, \alpha \bullet Q
\]
\[
\rightarrow^*_n (S_2)^\dagger([\alpha])_{\dagger}
\]
\[
\equiv (\mu x.S_2 :_n Q) \equiv ((S\alpha) :_n Q)
Case of \(\alpha:\)

\((O :: n \alpha) \equiv ([\alpha]O)^2 \equiv (\alpha^2\{O\})^2\)

Case of \([P, Q]:\)

\((O :: n [P, Q])\)

\[\begin{align*}
O :: n_{(\eta L)} (O :: [x. (x \bullet P), y. (y \bullet Q)])
\end{align*}\]

\[\text{I.H.(c)}\]

\[\begin{align*}
\rightarrow n^* (O :: [x. (P \_x{x})^2, y. (Q \_y{y})^2])
\end{align*}\]

\[\text{I.H.(a)}\]

\[\begin{align*}
\rightarrow n^* O^2 \bullet [x. (P \_x{x})^2, y. (Q \_y{y})^2]
\end{align*}\]

\[\equiv \delta (O, x. P \_x{x}, y. Q \_y{y})^2\]

\[\equiv ([P, Q]^2\{O\})^2\]

Case of \(\text{fst}[P]:\)

\((O :: n \text{fst}[P]) \equiv (\text{fst}(O) :: n P)\)

\[\text{I.H.(c)}\]

\[\begin{align*}
\rightarrow n^* (P \_x{x}\{\text{fst}(O)\})^2 \equiv (\text{fst}[P]^2\{O\})^2
\end{align*}\]

The case of \(\text{snd}[P]\) can be shown similarly.

Case of \(\text{not}(M):\)

\((O :: n \text{not}(M)) \equiv n^* (O :: n \text{not}((M)^2))\)

\[\text{I.H.(a)}\]

\[\begin{align*}
\rightarrow n^* (OM \_x{x})^2 \equiv (\text{not}(M)^2\{O\})^2
\end{align*}\]

Case of \([M \_K]:\)

\((O :: n (M \_K)) \equiv n^* (O :: ((M)^2)^2)\)

\[\text{I.H.(a)}\]

\[\begin{align*}
\rightarrow n^* (OM \_x{x} \_K x) \rightarrow n^* (P \_x{x}\{OM \_x{x}\})^2
\end{align*}\]

\[\equiv ([M \_K]^2\{O\})^2\]

Case of \([K, Q]\) (where \([K, Q]\) is not a covalue):

\((O^2 \bullet [K, L])\)

\[\text{I.H.(b)}\]

\[\begin{align*}
\rightarrow n^* O^2 \bullet [x. (x \bullet K), y. (y \bullet L)]
\end{align*}\]

\[\equiv ([K, L]^2\{O\})^2\]

Case of \(\text{fst}[K]\) (where \(K\) is not a covalue):

\((O^2 \bullet \text{fst}[K])\)

\[\text{I.H.(b)}\]

\[\begin{align*}
\rightarrow n^* (\text{fst}[K])^2 \equiv ([K]^2\{\text{fst}(O)\})^2
\end{align*}\]

The case of \(\text{snd}[K]\) can be shown similarly.

Case of \(\text{M@K}\) (where \(K\) is not a covalue):

\((O^2 \bullet \text{M@K})\)

\[\text{I.H.(b)}\]

\[\begin{align*}
\rightarrow n^* (\text{M@K})^2 \equiv ([\text{M@K}]^2\{\text{O}\})^2
\end{align*}\]
Case of $S.\alpha$:

\[
S.\alpha \rightarrow^{I.H.} (S_1) \mapsto (S_1) \downarrow \beta / \alpha, \beta
\]

\[
\equiv (\mu \alpha, S_1 \cdot \beta, \beta) \equiv (\mu \alpha, S_1) \downarrow
\]

\[
\equiv ((S.\alpha)) \downarrow
\]

Case of $\alpha$:

\[(O \cdot \alpha) \equiv ([\alpha]O) \downarrow \equiv (\alpha_1 \{O\}) \downarrow
\]

Case of $[K, L]$:

\[(O \cdot [K, L]) \equiv ([K, L]) \downarrow
\]

\[
\equiv (O \cdot v [x \cdot (x \cdot K), y \cdot (y \cdot L)])
\]

\[
\equiv (O \cdot v [x \cdot (K \cdot \{x\}), y \cdot (L \cdot \{y\})])
\]

\[
\equiv \delta (O, x, K \cdot \{x\}, y, L \cdot \{y\}) \downarrow
\]

\[
\equiv ([K, L] \{O\}) \downarrow
\]

Case of $\text{fst}[K]$:

\[(O \cdot \text{fst}[K]) \equiv (\text{fst}(O) \cdot \text{v} K)
\]

\[
\equiv (I.H.) \quad \text{v} \ast (K \cdot (I.H.)(O)) \downarrow
\]

\[
\equiv (I.H.) \quad \text{v} \ast (K \cdot \text{fst}(O)) \downarrow
\]

\[
\equiv (I.H.) \quad \text{v} \ast (K \cdot \text{fst}(O)) \downarrow \equiv (\text{fst}[K] \{O\}) \downarrow
\]

snd$[K]$ can be shown similarly.

Case of $\text{not}(M)$:

\[(O \cdot \text{not}(M)) \equiv (O \cdot \text{not}((M_1) \downarrow))
\]

\[
\equiv (O \cdot \text{not}((M_1) \downarrow)) \downarrow \equiv (\text{not}(M) \{O\}) \downarrow
\]

Case of $M \cdot K$:

\[(O \cdot v (M \cdot K)) \equiv (O \cdot (M_1) \downarrow \cdot \{K\})
\]

\[
\equiv (O \cdot \text{not}((M_1) \downarrow)) \downarrow \equiv (\text{not}(M) \{O\}) \downarrow
\]

\[
\equiv (I.H.) \quad \text{v} \ast (OM_1) \downarrow \equiv (M \cdot K) \{O\} \downarrow
\]

Case of $x.S$:

\[(O \cdot v \cdot x.S) \equiv (O \cdot v \cdot (S_1) \downarrow)
\]

\[
\equiv ((\lambda x.S_1)O) \downarrow \equiv (x.S_1) \{O\} \downarrow
\]

Case of $M \cdot K$:

\[(M \cdot K) \equiv (M_1) \downarrow \cdot K
\]

\[
\equiv (\beta \eta) (M_1 \cdot v) \downarrow \equiv (K \cdot \{M_1\}) \downarrow
\]

\[
\equiv (\lambda \alpha. \alpha \cdot \{M\} \downarrow \equiv \mu \alpha. \{M \cdot \alpha\} \downarrow
\]

\[
\equiv \mu \alpha. \alpha \cdot \{M\} \equiv (\lambda \alpha. \alpha \cdot \{M \cdot \alpha\} \downarrow
\]

\[
\equiv (\alpha \cdot \beta \cdot \alpha) \equiv \{M \} \equiv (\alpha \cdot \beta \cdot \alpha) \equiv \{M \} \downarrow
\]

\[
\equiv (\beta \cdot \beta \cdot \alpha) \equiv \{M \} \equiv (\alpha \cdot \beta) \equiv \{M \} \downarrow
\]

\[
\equiv (\beta \cdot \beta \cdot \alpha) \equiv \{M \} \equiv (\alpha \cdot \beta) \equiv \{M \} \downarrow
\]

\[
\equiv (\beta \cdot \beta \cdot \alpha) \equiv \{M \} \equiv (\alpha \cdot \beta) \equiv \{M \} \downarrow
\]

(4) can be shown in a way similar to (3). □

(Received August 21, 2006)

(Received January 9, 2007)

(Received April 11, 2007)

Daisuke Kimura was born in 1977. He received his M.S. degree from Kyoto University in 2003. He has been a Ph.D. student at the Graduate University for Advanced Studies since 2003. He has done research area on theoretical computer science and mathematical logic. His current research interest is type theory for classical logic. He is a member of IPSJ and JSSST.