Computation in Classical Logic and Dual Calculus

Daisuke Kimura

DOCTOR OF PHILOSOPHY

Department of Informatics,

School of Multidisciplinary Sciences,

The Graduate University for Advanced Studies

2006 (School Year)

March 2007

Acknowledgments

I would like to thank my supervisor, Professor Makoto Tatsuta, for his invaluable discussions, helpful advice and encouragement. I would also like to thank him that he gave me the chance to visit the University of Edinburgh and to discuss with many reseachers including Philip Wadler.

I would like to thank Kazushige Terui and Makoto Kanazawa for very helpful comments and discussions on this work.

I would like to thank Philip Wadler for valuable discussions and helpful comments. Particular, his work on the dual calculus gave me the starting point of my thesis. I am grateful to him for hosting me at the University of Edinburgh for a month. My visit was funded by Sokendai student dispatch program. In this time, I had many fruitful discussions with him about the dual calculus and his open question.

Finally, I wish to thank my family and friends, Kazuo Kimura, Keiko Kimura, Satoshi Kimura, Haruki Kimura, Tatsuya Abe, Ryo Yoshinaka, Camille Yamada, Takeshi Ozawa, Sebastien Duval and Atsuko Tanji, for their support and encouragement all along the way.

Contents

Chapter 1

Introduction

1.1 Background

The Curry-Howard correspondence

The Curry-Howard correspondence is a close relationship between computation and formal logic, which was first observed by Curry and Feys [10] and Howard [30]. Computation means the procedure that a computer follows according to its program, especially programs described by functional programming languages. Under the Curry-Howard correspondence, a formula of the logic is regarded as a type of a program, a proof is regarded as a program, and a normalization procedure of a proof is regarded as a computational procedure of a program. Therefore Howard called this correspondence ‘formulas-as-types’.

This correspondence plays the role of a bridge between theoretical computer science and proof theory. From the computer science point of view, the Curry-Howard correspondence gives the theoretical foundation of programming languages. This correspondence enables us to see a logical proof as a program with the proof of its correctness. Therefore it gives us a method ‘program extraction’ to verify a program, and obtain a correct one from a logical proof. On the other hand, from the proof theoretic point of view, this correspondence gives the interpretation of logical systems as programming languages. Logical systems that have such a property are called constructive logic; a typical example of such a logic is intuitionistic logic.

Although some mathematical models of computation have been proposed, the lambda calculus introduced by Church [7] is widely used today. This calculus has powerful expressive power though its simple grammar. The typed lambda calculus was also introduced by

Church, and is a foundation of functional programming languages. Gentzen introduced the two most widely used formulations of logic: natural deduction and sequent calculus, in both intuitionistic logic and classical logic. One of the most simple and essential formulation of the Curry-Howard correspondence given by Howard is the interpretation between proofs of propositional intuitionistic logic and terms of simply typed lambda calculus. For example, a derivation of simply typed lambda calculus

$$\dfrac{\dfrac{\dfrac{}{x : A \to B, v : A \to C \vdash v : A \to C} \quad \dfrac{}{z : A \vdash z : A}}{x : A \to B, v : A \to C, z : A \vdash vz : C}\ (\mathit{application})}{v : A \to C, z : A \vdash \lambda x.vz : (A \to B) \to C}\ (\mathit{abstraction})$$

corresponds to the following proof of intuitionistic logic

$$\dfrac{\dfrac{\dfrac{}{A \to B, A \to C \vdash A \to C} \quad \dfrac{}{A \vdash A}}{A \to B, A \to C, A \vdash C}\ (\to \mathit{Elim})}{A \to C, A \vdash (A \to B) \to C}\ (\to \mathit{Intro}) \qquad .$$

A lot of works has been done to extend the Curry-Howard correspondence. Girard [20] introduced system **F**, which corresponds to second-order propositional logic, and Reynolds [43] independently invented this system in his study of polymorphism in typed functional programming languages. Girard [20] extended this correspondence to higher-order intutionistic propositional logic. Coquand and Huet [8] proposed the Calculus of Constructions, and extended it to higher-order intuitionistic predicate logic. Moggi [37; 38] observed real programming language features such as non-termination, non-determinism and side-effects, and proposed the computational lambda calculus. Benton, Bierman and Paiva [6] extended the Curry-Howard correspondence to intutionistic modal propositional logic using Moggi's calculus.

Continuations and the $\lambda\mu$-calculus

In the recent years, extensions of the Curry-Howard correspondence that handle classical logic have been formulated. Felleisen [16] introduced the C operator to model `call/cc`, which is found in practical programming languages such as Scheme and SML/NJ. `call/cc` means '*call-with-current-continuation*', and it is one of the most typical examples of the operators that provide explicit handling of the current control continuation, *i.e.*, the current control context. This operator makes functional languages more expressive, for instance, exception handling and global jump, and allows us to describe more complicated programs.

Griffin [25] observed the type of Felleisen's C operator, and showed that `call/cc` corresponds to Peirce's Law, and extended Curry-Howard correspondence to classical logic. Here we give an informal explanation of `call/cc` operator. It is modelled by call-by-name simply typed lambda calculus with the following constants:

$$\begin{aligned} \mathtt{call/cc}_{A,B} &: ((A \to B) \to A) \to A \\ \mathtt{abort}_A &: \bot \to A \end{aligned}$$

where A, B are types and $\bot$ means the result type. Reduction rules about these constants are defined as follows.

$$\begin{aligned} E[\mathtt{call/cc}_{A,B}\, M] &\longrightarrow E[M(\lambda z^A.\mathtt{abort}_B\, E[z])] \\ E[\mathtt{abort}_A\, M] &\longrightarrow M \end{aligned}$$

where $E[-]$ is an evaluation context accepting a term of type A. Intuitively the `call/cc` operator carries the current context into its argument, and the `abort` operator aborts the current context. The following is an example of Scheme programs using `call/cc`.

```
(define multlist
  (lambda (inputlist)
    (call/cc
     (lambda (cc)
       (letrec ((ls*
                 (lambda (list)
                   (if (null? list)
                     1
                     (let ((x (car list)))
                        (if (= 0 x)
                            (cc 0)
                            (* x (ls* (cdr list))))))))))
        (ls* inputlist))))))
```

When `multlist` receives an integer list as its argument, it recursively multiplies the list elements. However, if it encounters an element equal to `0` during the calculation, `multlist` *immediately* returns 0.

In this line of works, the $\lambda\mu$-calculus introduced by Parigot [40] is well known. It corresponds to classical natural deduction, has a simple structure, and enjoys confluency and strong normalization. The $\lambda\mu$-calculus is as expressive as other popular functional programming languages with control operators. For example, above `call/cc` and `abort` are expressed by the following encodings:

$$
\begin{aligned}
(\texttt{call/cc}_{\texttt{A,B}}\ M)^* &= \mu\alpha^A.[\alpha]M^*(\lambda x^A \mu\beta^B.[\alpha]x)\\
(\texttt{abort}_{\texttt{A}}\ M)^* &= \mu\alpha^A.M^*
\end{aligned}
$$

Later, a call-by-value variant of the $\lambda\mu$-calculus was proposed by Ong and Stewart [39].

Computational duality and logical duality

Call-by-name strategy and call-by-value strategy have been well studied as evaluation strategies of programming languages. Filinsky [17] suggested that duality between call-by-name and call-by-value is clarified by the two notions of programs and continuations. Selinger [45] gave categorical semantics of the call-by-name $\lambda\mu$-calculus and the call-by-value one, and explained Filinski's duality in terms of categorical duality.

It is well known that the cut-elimination and normalization procedure of classical systems are non-deterministic. For example, if we consider the usual cut-elimination procedure of LK, then we can rewrite in the following two ways.

$$
\dfrac{\dfrac{\overline{A \vdash A}}{A \vdash A, C}\,Wk \qquad \dfrac{\overline{B \vdash B}}{C, B \vdash B}\,Wk}{A, B \vdash A, B}\,cut \quad \Longrightarrow \quad \dfrac{\overline{A \vdash A}}{\overline{A, B \vdash A, B}} \quad \text{and} \quad \dfrac{\overline{B \vdash B}}{\overline{A, B \vdash A, B}}
$$

This phenomenon does not depend on the formulation of classical logic, but depends on the duality of classical logic. There are numerous attempts to clarify the computational content of this duality of classical logic. Barbanera and Berardi [4] proposed the symmetric λ-calculus, which corresponds to natural deduction style of classical logic with a clear notion of duality. Curien and Herbelin [9] introduced the $\overline{\lambda}\mu\widetilde{\mu}$-calculus based on Gentzen's classical sequent calculus LK, and Wadler [48] proposed *the dual calculus*, which also corresponds to LK.

The feature of the dual calculus is that it has both of 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, the call-by-name and call-by-value strategies become dual strategies. Wadler [49] gave both directions of translation between the $\lambda\mu$-calculus and the dual calculus, and showed that these translations preserve the call-by-value and call-by-name strategies of each systems. 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 are the equational systems, and he showed only preservation property of the equality rules. This is because some rules are problematic to be introduced as reduction rules, such as (η)-rules. However, when we discuss his results from the point of view of correspondence between cut elimination procedure of sequent calculus and normalization procedure of natural deduction, we should consider reduction rules. In fact, Wadler noted an open question in his paper: whether one can replace the equations of his paper by reductions, and extend the properties for equations to those for reductions.

Constructive aspect of classical logic

There is also a proof-theoretical approach to extract computational content from classical proofs. The aim of this approach is to find a constructive classical system, which is complete w.r.t. classical provability and has a deterministic normalization procedure. There is a lot of works following this approach; FD and the $\lambda\mu$-calculus [40] by Parigot, LC by Girard [23], LKT and LKQ by Danos-Joinet-Schellinx [11], and *polarized linear logic* (LLP) by Laurent [33]. LLP is a variant of linear logic with a good denotational semantics in terms of coherent spaces. The most fundamental feature of LLP is that it has a clear distinction between *negative* formulas, for which structural rules can be freely used, and *positive* formulas, for which structural rules are forbidden. LLP is useful for understanding the constructive aspect of classical logic. In particular, LLP suggests some close relationships between the call-by-value / call-by-name computational duality and positive / negative logical duality.

Laurent defined two translations from the call-by-name and the call-by-value $\lambda\mu$-calculi into LLP, and showed their soundness, *i.e.* these translations preserve reductions. The call-by-name translation $(-)^\circ$ translates a classical formula into a negative one, in particular a classical implication $A \to B$ into a negative formula $!A^\circ \multimap B^\circ$. Therefore we call it "negative-translation" in this paper. On the other hand, the call-by-value translation $(-)^\bullet$ translates a classical formula into a positive one, in particular a classical implication $A \to B$ into a positive formula $!(A^\bullet \multimap ?B^\bullet)$. Therefore we call it "positive-translation" in this paper. Furthermore, Laurent showed fullness of the negative-translation, *i.e.*, every proof of A° is equivalent to an image of a proof of A in classical logic via the negative-translation in [34]. However, Laurent did not give a direct proof of the fullness of the positive-translation. Another work to be done is to give a term syntax for LLP. Although the formulations of LLP are given by the sequent calculus style and the proof-net style, proof-nets are mainly used to study LLP. However, it is natural and worth introducing a term syntax which corresponds to the sequent calculus style of LLP. Using such a term calculus, LLP will be understood better by comparing the proof-net style with the sequent calculus style, and by considering the relation to standard programming languages.

1.2 Contributions

The main theme of this thesis is to investigate the relationship between the computational duality and the logical duality. We will discuss this theme in the following two ways.

First, we will discuss the relationship between the computational duality of call-by-value / call-by-name and the logical duality of Gentzen's sequent calculus in chapter 2. Though the research in this approach has already been done by Wadler [49] using the $\lambda\mu$-calculus and the dual calculus, he did not consider these systems as reduction systems but as equational systems. However, when we discuss his results from a point of view of correspondence between computational procedure and cut elimination procedure of sequent calculus, we should consider reduction rules. So we refine the call-by-value and the call-by-name systems of the $\lambda\mu$-calculus and the dual calculus given in Wadler's paper. These systems are defined as reduction systems, and the main results of this chapter are Theorem 2.16, Theorem 2.21, Theorem 2.28, Theorem 2.34, Proposition 2.35, Proposition 2.36, and Theorem 2.40. The results of this chapter give the best possible answer to Wadler's open question: whether one can replace the equations of his paper by reductions, and extend the properties with equations to properties with reductions.

Second, we discuss the relationship between the computational duality call-by-value / call-by-name and the logical duality of positive / negative in chapter 3. We introduce a term calculus for (a sufficiently large fragment of) Laurent's polarized linear logic (LLP), called polarized dual calculus (DCP$^-$) which is based on the idea of the dual calculus. Laurent gave the two kinds of formulations for LLP: the sequent calculus style and the proof-net style. Proof-net is a well-known tool to observe computational properties of LLP, but there is no term syntax corresponding to the sequent style formulation of LLP. Hence it is natural to introduce a term syntax, which is compact and moreover well-related to standard functional programming languages. Then we define two translations from the call-by-name / the call-by-value $\lambda\mu$-calculi into DCP$^-$, and show their soundness (Theorem 3.4 and Theorem 3.7). These translations are almost straightforward adaptions of Laurent's, but the positive translation is slightly different. Finally, we prove the fullness of these translations in a way similar to the logical predicate method used by Hasegawa [27] (Theorem 3.19 and Theorem 3.27).

1.3 Overview of this thesis

In chapter 2, we give the best possible answer to Wadler's question. First, we analyze Wadler's results, and specify the problematic rules to solve his open question. Second, we refine the $\lambda\mu$-calculus and the dual calculus by the following steps. We adopt essential rules in the point of view of computation and normalization of proofs, and exclude the problematic and not essential rules. Then we give natural directions in the sense of computation and normalization of proofs. Third, we give a translation from the call-by-name $\lambda\mu$-calculus and the call-by-name dual calculus, and its inverse translation. We show that these translations preserve derivations and reductions. Further, we show the reloading properties of the call-by-name translations: the composition of the call-by-name translation become identity maps up to the call-by-name reductions. We also give call-by-value translations, and show the preservation and reloading properties for the call-by-value translations. Finally, we obtain our duality translations between the call-by-value $\lambda\mu$-calculus and the call-by-name one by composing our translations with duality on the dual calculus. Our results correspond to Wadler's, but they are based on reductions.

In chapter 3, we introduce a term calculus for a sufficiently large fragment of LLP, called polarized dual calculus (DCP$^-$), which is based on the idea of Wadler's dual calculus. Then we define two translations from the call-by-name / the call-by-value $\lambda\mu$-calculi into DCP$^-$, and show their soundness. These translations are almost straightforward adaptions of Laurent's, but the positive translation is slightly different. Finally, we prove the fullness of these translations in a way similar to the logical predicate method used by Hasegawa [27]. The notion of logical predicate (unary logical relation) is a well-established tool for studying the semantics of various typed lambda calculi. In particular, logical predicates for intuitionistic linear logic were introduced by Hasegawa [26] for category-theoretic models of linear logic, and applied to prove full completeness of Girard translation from the simply typed lambda calculus to the linear lambda calculus [27]. We adopt this method to show the fullness of Laurent's translations. The use of logical predicates allows us to give a *uniform* proof to the fullness of the two translations. In particular, single Basic Lemma is sufficient for both the positive- and the negative-translations.

Chapter 2

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

2.1 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[16] introduced the C operator to model `call/cc`, Griffin[25] 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[40] is well known. This calculus corresponds to classical natural deduction and has a simple structure, sufficient expressive power, and nice computational properties such as confluency and strong normalization. Later, a call-by-value (CBV) variant of the $\lambda\mu$-calculus was proposed by Ong and Stewart [39].

Duality

The call-by-name and call-by-value strategies have been well studied as evaluation strategies of programming languages. Filinski [17] suggested that duality between call-by-name and call-by-value is clarified by two notions of programs and continuations. Selinger [45] 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.

The dual calculus and Wadler's open question

Wadler[48; 49] 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 [49] 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 [44]. 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 $\lambda\mu$-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 of the $\lambda\mu$-calculus*
To simulate (ζ)-rule under Wadler's translation $(-)^*$, we need (β_R)-reductions of the dual

calculus in both directions. We give a typical example of this problem.

$$\begin{aligned}
((\mu\alpha.[\gamma]\lambda x.[\alpha]y)z)^* &\equiv \big(([x.(y \bullet \alpha)]\mathrm{not} \bullet \gamma).\alpha \bullet (z@\beta)\big) \\
&=^n_{(\beta_R)\text{-}red} \big([x.((y \bullet (z@\beta)))]\mathrm{not} \bullet \gamma\big).\beta \\
&=^n_{(\beta_R)\text{-}exp} \big([x.((y \bullet (z@\delta)).\delta \bullet \beta)]\mathrm{not} \bullet \gamma\big).\beta \\
&\equiv (\mu\beta.[\gamma]\lambda x.[\beta](yz))^*
\end{aligned}$$

Problem 2 ($\eta_\vee$)*-rule of the* $\lambda\mu$*-calculus:* $M =_n \mu(\alpha,\beta).[\alpha,\beta]M$
To simulate ($\eta_\vee$)-rule under Wadler's translation $(-)^*$, we need both of ($\eta_\vee$)-*reduction* and ($\eta_\vee$)-*expansion* of the dual calculus.

$$\begin{aligned}
(\mu(\alpha,\beta).[\alpha,\beta]M)^* &\equiv \Big(\Big\langle(\langle(M^* \bullet [\alpha,\beta]).\alpha\rangle\mathrm{inl} \bullet \gamma).\beta\Big\rangle\mathrm{inr} \bullet \gamma\Big).\gamma \\
&=^n_{(\eta_\vee)\text{-}exp} \Big(\Big\langle(\langle(M^* \bullet [\alpha,\beta]).\alpha\rangle\mathrm{inl} \bullet \gamma).\beta\Big\rangle\mathrm{inr} \bullet [x.(\langle x\rangle\mathrm{inl} \bullet \gamma), y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]\Big).\gamma \\
&=^n_{(\beta_\vee)} \big((\langle(M^* \bullet [\alpha,\beta]).\alpha\rangle\mathrm{inl} \bullet \gamma).\beta \bullet y.(\langle y\rangle\mathrm{inr} \bullet \gamma)\big).\gamma \\
&=^n_{(\beta_R)} \big(\langle(M^* \bullet [\alpha, y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]).\alpha\rangle\mathrm{inl} \bullet \gamma\big).\gamma \\
&=^n_{(\eta_\vee)\text{-}exp} \big(\langle(M^* \bullet [\alpha, y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]).\alpha\rangle\mathrm{inl} \bullet [x.(\langle x\rangle\mathrm{inl} \bullet \gamma), y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]\big).\gamma \\
&=^n_{(\beta_\vee)} \big((M^* \bullet [\alpha, y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]).\alpha \bullet x.(\langle x\rangle\mathrm{inl} \bullet \gamma)\big).\gamma \\
&=^n_{(\beta_R)} \big(M^* \bullet [x.(\langle x\rangle\mathrm{inl} \bullet \gamma), y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]).\gamma \\
&=^n_{(\eta_\vee)\text{-}red} \big(M^* \bullet \gamma).\gamma \\
&=^n_{(\eta_R)} M^*
\end{aligned}$$

However, if we simply omit ($\eta_\vee$)-rule to avoid this problem, then we meet another problem. If we want to obtain the equational correspondence formed by $(-)^*$ and $(-)_*$, which is one of Wadler's main results, then we should to show $((\mu(\alpha,\beta).S)^*)_* =_n \mu(\alpha,\beta).(S^*)_*$. We need ($\eta_\vee$)-rule to show this claim.

$$\begin{aligned}
((\mu(\alpha,\beta).S)^*)_* &\equiv \big(\langle(\langle S^*.\alpha\rangle\mathrm{inl} \bullet \gamma).\beta\rangle\mathrm{inr} \bullet \gamma\big).\gamma\big)_* \\
&\equiv \mu\gamma.[\gamma]\mu(\alpha',\beta').[\beta']\mu\beta.[\gamma]\mu(\alpha'',\beta'').[\alpha'']\mu\alpha.(S^*)_* \\
&=^{(\eta_\vee)\text{-}exp}_n \mu(\alpha''',\beta''').[\alpha''',\beta''']\mu\gamma.[\gamma]\mu(\alpha',\beta').[\beta']\mu\beta.[\gamma]\mu(\alpha'',\beta'').[\alpha'']\mu\alpha.(S^*)_* \\
&=^{(\zeta_\vee)}_n \mu(\alpha''',\beta''').[\alpha''',\beta''']\mu(\alpha',\beta').[\beta']\mu\beta.[\alpha''',\beta''']\mu(\alpha'',\beta'').[\alpha'']\mu\alpha.(S^*)_* \\
&=^{(\beta_\vee)}_n \mu(\alpha''',\beta''').[\beta''']\mu\beta.[\alpha''']\mu\alpha.(S^*)_* \\
&=^{(\beta_\mu)}_n \mu(\alpha''',\beta''').(S^*)_*[{}^{\alpha'''}/_{\alpha}, {}^{\beta'''}/_{\beta}] \\
&\equiv \mu(\alpha,\beta).(S^*)_*
\end{aligned}$$

Problem 3 ($\eta_\neg$)*-rule of the* $\lambda\mu$*-calculus:* $M =_n \lambda x.Mx$
To simulate ($\eta_\neg$)-rule under Wadler's translation $(-)^*$, we need both ($\eta_\neg$)-*reduction* and ($\eta_\neg$)-

expansion of the dual calculus.

$$
\begin{aligned}
(\lambda x.Mx)^* &\equiv [\, x.(M^* \bullet \mathrm{not}\langle x\rangle)\,]\mathrm{not} \\
&=^n_{(\eta_R)} ([\, x.(M^* \bullet \mathrm{not}\langle x\rangle)\,]\mathrm{not} \bullet \gamma).\gamma \\
&=^n_{(\eta_\neg)\text{-}exp} ([\, x.(M^* \bullet \mathrm{not}\langle x\rangle)\,]\mathrm{not} \bullet \mathrm{not}\langle\, ([\alpha]\mathrm{not} \bullet \gamma).\alpha\, \rangle).\gamma \\
&=^n_{(\beta_\neg)} (([\alpha]\mathrm{not} \bullet \gamma).\alpha \bullet x.(M^* \bullet \mathrm{not}\langle x\rangle)).\gamma \\
&=^n_{(\beta_L)} (M^* \bullet \mathrm{not}\langle\, ([\alpha]\mathrm{not} \bullet \gamma).\alpha\, \rangle).\gamma \\
&=^n_{(\eta_\neg)\text{-}red} (M^* \bullet \gamma).\gamma \\
&=^n_{(\eta_R)} M^*
\end{aligned}
$$

However, we need $(\eta_\neg)$-rule to simulate $x(\mu\alpha.S) =_v S[{}^{x\{-\}}/_{[\alpha]\{-\}}]$, which is defined in the call-by-value $\lambda\mu$-calculus as a part of (ζ)-rule. For example,

$$
\begin{aligned}
(x(\mu\alpha.[\beta]\lambda z.[\alpha]y))^* &\equiv x \bullet \mathrm{not}\Big\langle([z.(y \bullet \alpha)]\mathrm{not} \bullet \beta).\alpha\Big\rangle \\
&=^v_{(\eta_\neg)\text{-}exp} [y'.(x \bullet \mathrm{not}\langle y'\rangle)]\mathrm{not} \bullet \mathrm{not}\Big\langle([z.(y \bullet \alpha)]\mathrm{not} \bullet \beta).\alpha\Big\rangle \\
&=^v_{(\beta_\neg)} ([z.(y \bullet \alpha)]\mathrm{not} \bullet \beta).\alpha \bullet y'.(x \bullet \mathrm{not}\langle y'\rangle) \\
&=^v_{(\beta_L)} [z.(y \bullet y'.(x \bullet \mathrm{not}\langle y'\rangle))]\mathrm{not} \bullet \beta \\
&=^v_{(\beta_R)} [z.(x \bullet \mathrm{not}\langle y\rangle)]\mathrm{not} \bullet \beta \\
&\equiv ([\beta]\lambda z.(xy))^*
\end{aligned}
$$

For $(\eta_\supset)$-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 [13; 14]. 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 $\lambda\mu$-calculus and the dual calculus. Wadler added sum types to the $\lambda\mu$-calculus following Selinger [45]. This formulation is based on multiple-conclusioned sequents as follows.

$$
\frac{\Gamma \mid S \vdash_{\lambda\mu} \Delta, \alpha : A, \beta : B}{\Gamma \vdash_{\lambda\mu} \Delta \mid \mu(\alpha,\beta).S\ : A \vee B}
\qquad
\frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A \vee B}{\Gamma \mid [\alpha,\beta]M \vdash_{\lambda\mu} \Delta, \alpha : A, \beta : B}
$$

The formulation of sums in the dual calculus, on the other hand, is based on single-concluded sequents :

$$
\frac{\Gamma \vdash_{dc} \Delta \mid M : A}{\Gamma \vdash_{dc} \Delta \mid \langle M\rangle\mathrm{inl} : A \vee B}
\qquad
\frac{\Gamma \vdash_{dc} \Delta \mid N : B}{\Gamma \vdash_{dc} \Delta \mid \langle N\rangle\mathrm{inr} : A \vee B}
\qquad
\frac{\Gamma \vdash_{dc} \Delta \mid N : B}{[K, L] : A \vee B \mid \Gamma \vdash_{dc} \Delta \mid \langle N\rangle\mathrm{inr} : A \vee B}
$$

Our solution to this problem is to refine the formulation of sums in the $\lambda\mu$-calculus, and omit $(\eta_\vee)$-rule. We introduce sums of the $\lambda\mu$-calculus by using usual injections and case-expressions.

To avoid Problem 3, we remove $(\eta_\neg)$ and $(\eta_\supset)$-rules, and restrict the call-by-value $\lambda\mu$-calculus by omitting some rules that cannot be simulated without $(\eta_\neg)$ and $(\eta_\supset)$-rules.

We also encounter problems when we consider the inverse translation from the dual calculus into the $\lambda\mu$-calculus. Since they are similar to the above Problem 1, we can solve them by modifying Wadler's original translation. However, we also need different modifications for call-by-value and call-by-name.

Overview

In section 2, we present the detailed formulation of our call-by-value and call-by-name $\lambda\mu$-calculi, and compare them with the $\lambda\mu$-calculi given by Wadler (2005). In section 3, we present the dual calculus as a reduction system. In section 4, we define the call-by-name translation from the call-by-name $\lambda\mu$-calculus into the call-by-name dual calculus, and show that this translation preserve call-by-name reductions (Theorem 2.16). We also define the call-by-value translation, and show that it also preserves call-by-value reductions. In section 5, we give the inverse translations from the dual calculus into the $\lambda\mu$-calculus for each of call-by-name and call-by-value, and show that they preserves reductions (Theorem 2.28, 2.34). We also show that the compositions of the call-by-name translations become identity maps up to the call-by-name reductions, and show the similar property for the call-by-value translations (Proposition 2.35, 2.36). 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 2.40), but our results are based on reductions.

2.2 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.

In this paper, we follow Wadler for the types of the $\lambda\mu$-calculus, *i.e.*. let A and B range over types, then a type is atomic X, a conjunction $A \wedge B$, a disjunction $A \vee B$, a negation $\neg A$, or an implication $A \supset B$.

Types of the $\lambda\mu$-calculus

$$A, B ::= X \mid A \wedge B \mid A \vee B \mid A \supset B \mid \neg A$$

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 also follow Wadler for the expressions. The expressions consist of *terms* (denoted by $M, N, \ldots$) and *statements* (denoted by $S, T, \ldots$). A term is a variable x, a λ-abstraction $\lambda x.M$ or $\lambda x.S$, an implication application OM (where $O : A \supset B$), a projection $\mathrm{fst}(M)$ or $\mathrm{snd}(M)$, a pairing $\langle M, N\rangle$, a μ-abstraction $\mu\alpha.S$, or a term for sums. A statement is a covariable application $[\alpha]M$, a negation application 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 α in S is bound in the term $\mu\alpha.S$.

Our formulation of sums is different from Selinger's [45]. A term for sums is a left injection $\mathrm{inl}(M)$, a right injection $\mathrm{inr}(M)$, 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 $\delta(O, x.S, y.T)$.

Terms and statements of the $\lambda\mu$-calculus

$$\begin{aligned} M, N, O &::= x \mid \lambda x.M \mid \lambda x.S \mid MN \mid \mu\alpha.S \mid \mathrm{fst}(M) \mid \mathrm{snd}(N) \mid \langle M, N\rangle \\ &\quad \mid \mathrm{inl}(M) \mid \mathrm{inr}(N) \mid \delta(O, x.M, y.N) \\ S, T &::= [\alpha]M \mid MN \mid \delta(O, x.S, y.T) \end{aligned}$$

We consider the term modulo α-conversion of variables and covariables. The sets of *free variables* of M and S (denoted by $\mathrm{FV}(M)$ and $\mathrm{FV}(S)$), and the sets of *free covariables* of M and S (denoted by $\mathrm{FCV}(M)$ and $\mathrm{FCV}(S)$) are defined as usual. A *typing judgment* of the $\lambda\mu$-calculus takes the form $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$ or $\Gamma \mid S \vdash_{\lambda\mu} \Delta$, where Γ denotes a *λ-context, i.e.* $x_1 : A_1, \ldots, x_n : A_n$, and Δ denotes a *μ-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 figure 2.1.

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

$$([\alpha]M)[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}] \equiv \mathcal{T}\{\, M[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}]\,\}$$

$$\frac{}{\Gamma, x : A \vdash_{\lambda\mu} \Delta \mid x : A}\ \text{Ax}$$

$$\frac{\Gamma, x : A \vdash_{\lambda\mu} \Delta \mid M : B}{\Gamma \vdash_{\lambda\mu} \Delta \mid \lambda x.M : A \supset B}\ \supset \text{I} \qquad \frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A \supset B \quad \Gamma \vdash_{\lambda\mu} \Delta \mid N : A}{\Gamma \vdash_{\lambda\mu} \Delta \mid MN : B}\ \supset \text{E}$$

$$\frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A \wedge B}{\Gamma \vdash_{\lambda\mu} \Delta \mid \text{fst}(M) : A}\ \wedge\text{E}_1 \qquad \frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A \wedge B}{\Gamma \vdash_{\lambda\mu} \Delta \mid \text{snd}(M) : B}\ \wedge\text{E}_2$$

$$\frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A}{\Gamma \vdash_{\lambda\mu} \Delta \mid \text{inl}(M) : A \vee B}\ \vee\text{I}_1 \qquad \frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A}{\Gamma \vdash_{\lambda\mu} \Delta \mid \text{inr}(M) : A \vee B}\ \vee\text{I}_2$$

$$\frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A \quad \Gamma \vdash_{\lambda\mu} \Delta \mid N : B}{\Gamma \vdash_{\lambda\mu} \Delta \mid \langle M, N \rangle : A \wedge B}\ \wedge\text{I}$$

$$\frac{\Gamma \vdash_{\lambda\mu} \Delta \mid O : A \vee B \quad \Gamma, x : A \vdash_{\lambda\mu} \Delta \mid M : C \quad \Gamma, y : B \vdash_{\lambda\mu} \Delta \mid N : C}{\Gamma \vdash_{\lambda\mu} \Delta \mid \delta(O, x.M, y.N) : C}\ \vee\text{E}$$

$$\frac{\Gamma \vdash_{\lambda\mu} \Delta \mid O : A \vee B \quad \Gamma, x : A \mid S \vdash_{\lambda\mu} \Delta \quad \Gamma, y : B \mid T \vdash_{\lambda\mu} \Delta}{\Gamma \mid \delta(O, x.S, y.T) \vdash_{\lambda\mu} \Delta}\ \vee\text{E}$$

$$\frac{\Gamma, x : A \mid S \vdash_{\lambda\mu} \Delta}{\Gamma \vdash_{\lambda\mu} \Delta \mid \lambda x.S : \neg A}\ \neg\text{I} \qquad \frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : \neg A \quad \Gamma \vdash_{\lambda\mu} \Delta \mid N : A}{\Gamma \mid MN \vdash_{\lambda\mu} \Delta}\ \neg\text{E}$$

$$\frac{\Gamma \mid S \vdash_{\lambda\mu} \Delta, \alpha : A}{\Gamma \vdash_{\lambda\mu} \Delta \mid \mu\alpha.S : A}\ \text{Act} \qquad \frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A}{\Gamma \mid [\alpha]M \vdash_{\lambda\mu} \Delta, \alpha : A}\ \text{Pass}$$

Figure 2.1: Typing rules of the $\lambda\mu$-calculus

The other clauses are defined homomorphically like

$$(MN)[^{\mathcal{T}(-)}/_{[\alpha](-)}] \equiv M[^{\mathcal{T}(-)}/_{[\alpha](-)}]N[^{\mathcal{T}(-)}/_{[\alpha](-)}].$$

Note that $M[^{[\beta](-)}/_{[\alpha](-)}]$ and $S[^{[\beta](-)}/_{[\alpha](-)}]$ are sometimes written as $M[^{\beta}/_{\alpha}]$ and $S[^{\beta}/_{\alpha}]$ respectively.

2.2.1 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. *Call-by-name evaluation term and statement contexts*

$$E_n, E'_n, E''_n ::= \{-\} \mid E_n M \mid \text{fst}(E_n) \mid \text{snd}(E_n) \mid \delta(E_n, x.E'_n\{x\}, y.E''_n\{y\})$$

$$
\begin{array}{lll}
(\beta_{\supset}) & (\lambda x.M)N \longrightarrow_n M[{}^{N}/_{x}] & \\
(\beta_{\wedge}) & \mathrm{fst}\langle M, N\rangle \longrightarrow_n M & \\
 & \mathrm{snd}\langle M, N\rangle \longrightarrow_n N & \\
(\beta_{\vee}) & \delta(\mathrm{inl}(O), x.E_n\{x\}, y.E'_n\{y\}) \longrightarrow_n E_n\{O\} & \\
 & \delta(\mathrm{inr}(O), x.E_n\{x\}, y.E'_n\{y\}) \longrightarrow_n E'_n\{O\} & \\
 & \delta(\mathrm{inl}(O), x.D_n\{x\}, y.D'_n\{y\}) \longrightarrow_n D_n\{O\} & \\
 & \delta(\mathrm{inr}(O), x.D_n\{x\}, y.D'_n\{y\}) \longrightarrow_n D'_n\{O\} & \\
(\beta_{\neg}) & (\lambda x.S)N \longrightarrow_n S[{}^{N}/_{x}] & \\
(\zeta) & E_n\{\mu\alpha.S\} \longrightarrow_n \mu\beta.S[{}^{[\beta]E_n\{-\}}/_{[\alpha]\{-\}}] & (\text{where } E_n \text{ is not } \{-\}) \\
 & D_n\{\mu\alpha.S\} \longrightarrow_n S[{}^{D_n\{-\}}/_{[\alpha]\{-\}}] & \\
(\eta_{\mu}) & M \longrightarrow_n \mu\alpha.[\alpha]M & (\text{where } \alpha \notin \mathrm{FCV}(M)) \\
(\pi) & E_n\{\delta(O, x.M, y.N)\} \longrightarrow_n \delta(O, x.E_n\{M\}, y.E_n\{N\}) & (\text{where } E_n \text{ is not } \{-\}) \\
 & D_n\{\delta(O, x.M, y.N)\} \longrightarrow_n \delta(O, x.D_n\{M\}, y.D_n\{N\}) & \\
(\nu) & \delta(O, x.S, y.T) \longrightarrow_n (\lambda y.T)\,\mu\beta.\delta(O, x.S, y.[\beta]y) & \\
 & & (\text{if } T \text{ is not a simple form w.r.t. } y) \\
 & \delta(O, x.S, y.D_n\{y\}) \longrightarrow_n (\lambda x.S)\,\mu\alpha.\delta(O, x.[\alpha]x, y.D_n\{y\}) & \\
 & & (\text{if } S \text{ is not a simple form w.r.t. } x)
\end{array}
$$

Figure 2.2: Reduction rules of the call-by-name $\lambda\mu$-calculus

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

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 $\longrightarrow_n$, is defined as the compatible closure of the rules in figure 2.2. We write $\longrightarrow_n{}^*$ for the reflexive transitive closure of $\longrightarrow_n$. Similarly, we write $\longrightarrow_n{}^+$ and $=_n$ for the transitive closure and the reflexive symmetric transitive closure of $\longrightarrow_n$ 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 $\lambda\mu \vdash M \longrightarrow_n N$ 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.

(β)-rules reduce a deconstructor applied to a constructor. Note that ($\beta_{\vee}$)-rule has an unusual and 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, (ζ)-rules substitute an evaluation context and a statement context for a covalue, and (η_μ)-rule introduces a μ-abstraction applied to a covariable application. (π)-rules correspond to the permutative conversions, and (ν)-rules expand a case statement $\delta(M, x.S, y.T)$ when S or T is not a simple form, 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 $\lambda\mu^{wad}$ and write the call-by-name and call-by-value variants of $\lambda\mu^{wad}$ by $\lambda\mu_n^{wad}$ and $\lambda\mu_v^{wad}$ respectively. Detailed definitions of these systems can be found in appendix. We compare our call-by-name system with the $\lambda\mu_n^{wad}$-calculus. The differences between them are summarized in the following three points:

- Our system is based on reduction relations while $\lambda\mu_n^{wad}$ is based on equations,
- the formulation of sums in our system is different from that in $\lambda\mu_n^{wad}$, and
- our system does not have (η)-rules related to implications, negations, pairs, and sums while his system does have them.

We give two translations, $[\![-]\!]$ and $\langle\!\langle - \rangle\!\rangle$, between our $\lambda\mu$ and $\lambda\mu^{wad}$ that interpret sums as follows.

Translation $[\![-]\!]$: *from* $\lambda\mu^{wad}$ *into our* $\lambda\mu$

$$[\![\mu(\alpha,\beta).S\,]\!] \equiv \mu\gamma.[\![S\,]\!][{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}]$$
$$[\![[\alpha,\beta]M]\!] \equiv \delta([\![M]\!], x.[\alpha]x, y.[\beta]y)$$

The other clauses are defined homomorphically like $[\![MN]\!] \equiv [\![M]\!][\![N]\!]$.
Translation $\langle\!\langle - \rangle\!\rangle$: *from our* $\lambda\mu$ *into* $\lambda\mu^{wad}$

$$\langle\!\langle \mathrm{inl}(M)\rangle\!\rangle \equiv \mu(\alpha,\beta).[\alpha]\langle\!\langle M\rangle\!\rangle$$
$$\langle\!\langle \mathrm{inr}(N)\rangle\!\rangle \equiv \mu(\alpha,\beta).[\beta]\langle\!\langle N\rangle\!\rangle$$
$$\langle\!\langle \delta(O, x.M, y.N)\rangle\!\rangle \equiv \mu\gamma.\Big(\lambda y.[\gamma]\langle\!\langle N\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big)$$
$$\langle\!\langle \delta(O, x.S, y.T)\rangle\!\rangle \equiv \Big(\lambda y.\langle\!\langle T\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.\langle\!\langle S\,\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big)$$

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

We define the call-by-name system $\lambda\mu_n^\eta$ as the one generated by the rules in figure 2.2 and the following (η)-rules,

$$(\eta_\supset) \qquad M \longrightarrow_n \lambda x.Mx \qquad\qquad (\text{where } M : A \supset B)$$

$$
\begin{array}{lll}
(\eta_\wedge) & M \longrightarrow_n \langle \mathrm{fst}(M), \mathrm{snd}(M)\rangle & (\text{where } M : A \wedge B)\\
(\eta_\vee) & M \longrightarrow_n \delta(M, x.\mathrm{inl}(x), y.\mathrm{inr}(y)) & (\text{where } M : A \vee B)\\
(\eta_\neg) & M \longrightarrow_n \lambda x.Mx & (\text{where } M : \neg A)
\end{array}
$$

$\lambda\mu_n^\eta$ is equivalent to Wadler's call-by-name system $\lambda\mu_n^{wad}$ 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 $\lambda\mu_n^\eta$-calculus. Then, we define the call-by-name contexts $\overline{E_n}$ and $\overline{D_n}$ of the $\lambda\mu_n^{wad}$-calculus as follows.

$$
\begin{array}{rlrl}
\overline{\{-\}} \equiv \{-\} & & \overline{E_n N} \equiv \overline{E_n}\langle\!\langle N\rangle\!\rangle &\\
\overline{\mathrm{fst}(E_n)} \equiv \mathrm{fst}(\overline{E_n}) & & \overline{\mathrm{snd}(E_n)} \equiv \mathrm{snd}(\overline{E_n}) &\\
\multicolumn{4}{c}{\overline{\delta(E_n, x.E'_n\{x\}, y.E''_n\{y\})} \equiv \mu\gamma.[\gamma]\overline{E''_n}\Big\{\mu\beta.[\gamma]\overline{E'_n}\{\mu\alpha.[\alpha,\beta]\overline{E_n}\}\Big\}}\\
\overline{[\alpha]E_n} \equiv [\alpha]\overline{E_n} & & \overline{E_n N} \equiv \overline{E_n}\langle\!\langle N\rangle\!\rangle &\\
\multicolumn{4}{c}{\overline{\delta(E_n, x.D_n\{x\}, y.D'_n\{y\})} \equiv \overline{D'_n}\Big\{\mu\beta.\overline{D_n}\{\mu\alpha.[\alpha,\beta]\overline{E_n}\}\Big\}}
\end{array}
$$

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

Lemma 2.1

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

(2) $\lambda\mu_n^{wad} \vdash \langle\!\langle M\rangle\!\rangle[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}] =_n \langle\!\langle M[{}^{D_n\{-\}}/{}_{[\alpha]\{-\}}]\rangle\!\rangle$ and $\lambda\mu_n^{wad} \vdash \langle\!\langle S\rangle\!\rangle[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}] =_n \langle\!\langle S[{}^{D_n\{-\}}/{}_{[\alpha]\{-\}}]\rangle\!\rangle$ hold.

(3) $\lambda\mu_n^{wad} \vdash \overline{E_n}\{\mu\alpha.S\} =_n \mu\beta.S[{}^{[\beta]\overline{E_n}\{-\}}/{}_{[\alpha]\{-\}}]$ and $\lambda\mu_n^{wad} \vdash \overline{D_n}\{\mu\alpha.S\} =_n S[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}]$ hold.

Proof. (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.

$$
\begin{aligned}
\langle\!\langle [\alpha]M\rangle\!\rangle[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}] &\equiv ([\alpha]\langle\!\langle M\rangle\!\rangle)[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}] \equiv \overline{D_n}\{\langle\!\langle M\rangle\!\rangle[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}]\}\\
&\overset{I.H.}{=}_n \overline{D_n}\{\langle\!\langle M[{}^{D_n\{-\}}/{}_{[\alpha]\{-\}}]\rangle\!\rangle\} \overset{(1)}{=}_n \langle\!\langle D_n\{M[{}^{D_n\{-\}}/{}_{[\alpha]\{-\}}]\}\rangle\!\rangle \equiv \langle\!\langle ([\alpha]M)[{}^{D_n\{-\}}/{}_{[\alpha]\{-\}}]\rangle\!\rangle
\end{aligned}
$$

(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.

$$
\begin{aligned}
\overline{\delta(E_n, x.D_n\{x\}, y.D'_n\{y\})}\{\mu\alpha.S\} &\equiv \overline{D'_n}\Big\{\mu\beta.\overline{D_n}\{\mu\alpha.[\alpha,\beta]\overline{E_n}\{\mu\alpha.S\}\}\Big\}\\
&\overset{I.H.}{=}_n \overline{D'_n}\Big\{\mu\beta.\overline{D_n}\{\mu\alpha.[\alpha,\beta]\mu\alpha'.S[{}^{[\alpha']\overline{E_n}\{-\}}/{}_{[\alpha]\{-\}}]\}\Big\}\\
&=_{\zeta_\vee} \overline{D'_n}\Big\{\mu\beta.\overline{D_n}\{\mu\alpha.S[{}^{[\alpha,\beta]\overline{E_n}\{-\}}/{}_{[\alpha]\{-\}}]\}\Big\}
\end{aligned}
$$

$$
\begin{aligned}
&\overset{I.H.}{=}_n \overline{D'_n}\Big\{\mu\beta.S\,[{}^{[\alpha,\beta]\overline{E_n}\{-\}}/_{[\alpha]\{-\}}][{}^{\overline{D_n}\{-\}}/_{[\alpha]\{-\}}]\Big\}\\
&\overset{I.H.}{=}_n S\,[{}^{[\alpha,\beta]\overline{E_n}\{-\}}/_{[\alpha]\{-\}}][{}^{\overline{D_n}\{-\}}/_{[\alpha]\{-\}}][{}^{\overline{D'_n}\{-\}}/_{[\beta]\{-\}}]\\
&\equiv S\,[{}^{([\alpha,\beta]\overline{E_n}\{-\})[{}^{\overline{D_n}\{-\}}/_{[\alpha]\{-\}}][{}^{\overline{D'_n}\{-\}}/_{[\beta]\{-\}}]}/_{[\alpha]\{-\}}]\\
&\overset{(*)}{\equiv} S\,[{}^{\overline{D'_n}\{\mu\beta.\overline{D_n}\{\mu\alpha.[\alpha,\beta]\overline{E_n}\{-\}\}\}}/_{[\alpha]\{-\}}]\\
&\equiv S\,[{}^{\overline{\delta(E_n,x.D_n\{x\},y.D'_n\{y\})}\{-\}}/_{[\alpha]\{-\}}]
\end{aligned}
$$

where I.H. means the induction hypothesis, and $(*)$ is by the definition of the substitution for a covariable. $\square$

Proposition 2.2

(1) If $\lambda\mu_n^\eta \vdash M =_n N$, then $\lambda\mu_n^{wad} \vdash \langle\!\langle M\rangle\!\rangle =_n \langle\!\langle N\rangle\!\rangle$, and if $\lambda\mu_n^\eta \vdash S =_n T$, then $\lambda\mu_n^{wad} \vdash \langle\!\langle S\rangle\!\rangle =_n \langle\!\langle T\rangle\!\rangle$.

(2) If $\lambda\mu_n^{wad} \vdash M =_n N$, then $\lambda\mu_n^\eta \vdash [\![M]\!] =_n [\![N]\!]$, and if $\lambda\mu_n^{wad} \vdash S =_n T$, then $\lambda\mu_n^\eta \vdash [\![S]\!] =_n [\![T]\!]$.

(3) $\lambda\mu_n^\eta \vdash [\![\langle\!\langle M\rangle\!\rangle]\!] =_n M$ and $\lambda\mu_n^\eta \vdash [\![\langle\!\langle S\rangle\!\rangle]\!] =_n S$.

(4) $\lambda\mu_n^{wad} \vdash \langle\!\langle [\![M]\!]\rangle\!\rangle =_n M$ and $\lambda\mu_n^{wad} \vdash \langle\!\langle [\![S]\!]\rangle\!\rangle =_n S$.

Proof. (1) We can show this by induction on the call-by-name equation of the $\lambda\mu_n^\eta$-calculus. We consider only the rules about sums, *i.e.*, $(\beta_\vee)$, (ζ), (π), (ν), and $(\eta_\vee)$-rules.
Case of $(\beta_\vee)$-rule :

$$
\begin{aligned}
&\langle\!\langle \delta(\mathrm{inl}(O), x.E'_n\{x\}, y.E''_n\{y\})\rangle\!\rangle\\
&\quad\equiv \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle E''_n\{y\}\rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle E'_n\{x\}\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle \mathrm{inl}(O)\rangle\!\rangle)\big)\\
&\quad\equiv \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle E''_n\{y\}\rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle E'_n\{x\}\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\mu(\alpha',\beta').[\alpha']\langle\!\langle O\rangle\!\rangle)\big)\\
&\quad=_{\beta_\vee} \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle E''_n\{y\}\rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle E'_n\{x\}\rangle\!\rangle)(\mu\alpha.[\alpha]\langle\!\langle O\rangle\!\rangle)\big)\\
&\quad=_{\eta_\mu} \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle E''_n\{y\}\rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle E'_n\{x\}\rangle\!\rangle)\langle\!\langle O\rangle\!\rangle\big)\\
&\quad=_n \mu\gamma.\big(\lambda y.[\gamma]\overline{E''_n}\{y\}\big)\big(\mu\beta.(\lambda x.[\gamma]\overline{E'_n}\{x\})\langle\!\langle O\rangle\!\rangle\big) && \text{(by Lem 2.1 (1))}\\
&\quad=_{\beta_\neg} \mu\gamma.[\gamma]\overline{E''_n}\big\{\mu\beta.[\gamma]\overline{E'_n}\{\langle\!\langle O\rangle\!\rangle\}\big\}\\
&\quad=_n \mu\gamma.[\gamma]\overline{E'_n}\{\langle\!\langle O\rangle\!\rangle\} && \text{(by Lem 2.1 (3))}\\
&\quad=_{\eta_\mu} \overline{E'_n}\{\langle\!\langle O\rangle\!\rangle\}\\
&\quad=_n \langle\!\langle E'_n\{O\}\rangle\!\rangle && \text{(by Lem 2.1 (1))}
\end{aligned}
$$

The other rules of the $(\beta_\vee)$-rule can also be shown similarly.
Case of (ζ)-rule :

$$\begin{aligned}
\langle\!\langle E_n\{\mu\alpha.S\}\rangle\!\rangle &=_n \overline{E_n}\{\langle\!\langle \mu\alpha.S\rangle\!\rangle\} && \text{(by Lem 2.1 (1))}\\
&\equiv \overline{E_n}\{\mu\alpha.\langle\!\langle S\rangle\!\rangle\} =_n \mu\beta.\langle\!\langle S\rangle\!\rangle[{}^{[\beta]\overline{E_n}\{-\}}/{}_{[\alpha]\{-\}}] && \text{(by Lem 2.1 (3))}\\
&=_n \langle\!\langle \mu\beta.S\,[{}^{[\beta]E_n\{-\}}/{}_{[\alpha]\{-\}}]\rangle\!\rangle && \text{(by Lem 2.1 (2))}\\
\langle\!\langle D_n\{\mu\alpha.S\}\rangle\!\rangle &=_n \overline{D_n}\{\langle\!\langle \mu\alpha.S\rangle\!\rangle\} =_n \overline{D_n}\{\mu\alpha.\langle\!\langle S\rangle\!\rangle\}\\
&=_n S\,[{}^{\overline{D_n}\{-\}}/{}_{[\alpha]\{-\}}] =_n \langle\!\langle S\,[{}^{D_n\{-\}}/{}_{[\alpha]\{-\}}]\rangle\!\rangle
\end{aligned}$$

Case of (π)-rule :

$$\begin{aligned}
&\langle\!\langle E_n\{\delta(O,x.M,y.N\}\rangle\!\rangle =_n \overline{E_n}\Big\{\langle\!\langle \delta(O,x.M,y.N)\rangle\!\rangle\Big\} && \text{(by Lem 2.1 (1))}\\
&\quad\equiv \overline{E_n}\Big\{\mu\gamma.\Big(\lambda y.[\gamma]\langle\!\langle N\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big)\Big\}\\
&\quad=_n \mu\gamma'.\Big(\lambda y.[\gamma']\overline{E_n}\{\langle\!\langle N\rangle\!\rangle\}\Big)\Big(\mu\beta.(\lambda x.[\gamma']\overline{E_n}\{\langle\!\langle M\rangle\!\rangle\})(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big) && \text{(by Lem 2.1 (3))}\\
&\quad=_n \mu\gamma'.\Big(\lambda y.[\gamma']\langle\!\langle E_n\{N\}\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.[\gamma']\langle\!\langle E_n\{M\}\rangle\!\rangle\})(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big) && \text{(by Lem 2.1 (1))}\\
&\quad\equiv \langle\!\langle \delta(O,x.E_n\{M\},y.E_n\{N\})\rangle\!\rangle
\end{aligned}$$

The other rule of the (π)-rule can also be shown similarly.
Case of (ν)-rule :

$$\begin{aligned}
\langle\!\langle \delta(O,x.S,y.T)\rangle\!\rangle &\equiv \Big(\lambda y.\langle\!\langle T\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.\langle\!\langle S\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big)\\
&=_n \Big(\lambda y.\langle\!\langle T\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda y'.[\beta]y')(\mu\beta'.(\lambda x.\langle\!\langle S\rangle\!\rangle)(\mu\alpha.[\alpha,\beta']\langle\!\langle O\rangle\!\rangle))\Big)\\
&\equiv \Big(\lambda y.\langle\!\langle T\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda y'.\langle\!\langle [\beta]y'\rangle\!\rangle)(\mu\beta'.(\lambda x.\langle\!\langle S\rangle\!\rangle)(\mu\alpha.[\alpha,\beta']\langle\!\langle O\rangle\!\rangle))\Big)\\
&\equiv \langle\!\langle (\lambda y.T)\,\mu\beta.\delta(O,x.S,y.[\beta]y)\rangle\!\rangle\\
\langle\!\langle \delta(O,x.S,y.D_n\{y\})\rangle\!\rangle &\equiv \Big(\lambda y.\langle\!\langle D_n\{y\}\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.\langle\!\langle S\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big)\\
&=_n \Big(\lambda y.\overline{D_n}\{y\}\Big)\Big(\mu\beta.(\lambda x.\langle\!\langle S\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big)\\
&=_n \overline{D_n}\Big\{\mu\beta.(\lambda x.\langle\!\langle S\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O\rangle\!\rangle)\Big\}\\
&=_n \overline{D_n}\Big\{\mu\beta.\Big(\lambda x.\langle\!\langle S\rangle\!\rangle\Big)\Big(\mu\alpha.(\lambda y.[\beta]y)(\mu\beta'.(\lambda x.[\alpha]x)(\mu\alpha'.[\alpha',\beta']\langle\!\langle O\rangle\!\rangle))\Big)\Big\}\\
&=_n \Big(\lambda x.\langle\!\langle S\rangle\!\rangle\Big)\Big(\mu\alpha.(\lambda y.\overline{D_n}\{y\})(\mu\beta'.(\lambda x.[\alpha]x)(\mu\alpha'.[\alpha',\beta']\langle\!\langle O\rangle\!\rangle))\Big)\\
&=_n \Big(\lambda x.\langle\!\langle S\rangle\!\rangle\Big)\Big(\mu\alpha.(\lambda y.\langle\!\langle D_n\{y\}\rangle\!\rangle)(\mu\beta'.(\lambda x.\langle\!\langle [\alpha]x\rangle\!\rangle)(\mu\alpha'.[\alpha',\beta']\langle\!\langle O\rangle\!\rangle))\Big)\\
&\equiv \Big(\lambda x.\langle\!\langle S\rangle\!\rangle\Big)\Big(\mu\alpha.\langle\!\langle \delta(O,x.[\alpha]x,y.D_n\{y\})\rangle\!\rangle\Big)\\
&\equiv \langle\!\langle (\lambda x.S)\,(\mu\alpha.\delta(O,x.[\alpha]x,y.D_n\{y\}))\rangle\!\rangle
\end{aligned}$$

Case of $(\eta_\vee)$-rule :

$$\langle\!\langle \delta(M,x.\mathrm{inl}(x),y.\mathrm{inr}(y))\rangle\!\rangle \equiv \mu\gamma.\Big(\lambda y.[\gamma]\langle\!\langle \mathrm{inr}(y)\rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle \mathrm{inr}(x)\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\Big)$$

$$
\begin{aligned}
&\equiv \mu\gamma.\big(\lambda y.[\gamma]\mu(\alpha'',\beta'').[\beta'']y\big)\big(\mu\beta.(\lambda x.[\gamma]\mu(\alpha',\beta').[\alpha']x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big)\\
&=_{\eta_\vee} \mu(\alpha_1,\beta_1).[\alpha_1,\beta_1]\mu\gamma.\big(\lambda y.[\gamma]\mu(\alpha'',\beta'').[\beta'']y\big)\big(\mu\beta.(\lambda x.[\gamma]\mu(\alpha',\beta').[\alpha']x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big)\\
&=_{\zeta_\vee} \mu(\alpha_1,\beta_1).\big(\lambda y.[\alpha_1,\beta_1]\mu(\alpha'',\beta'').[\beta'']y\big)\big(\mu\beta.(\lambda x.[\alpha_1,\beta_1]\mu(\alpha',\beta').[\alpha']x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big)\\
&=_{\beta_\vee} \mu(\alpha_1,\beta_1).\big(\lambda y.[\beta_1]y\big)\big(\mu\beta.(\lambda x.[\alpha_1]x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big)\\
&=_{(\beta_\neg)} \mu(\alpha_1,\beta_1).[\beta_1]\mu\beta.[\alpha_1]\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle\\
&=_{\beta_\mu} \mu(\alpha_1,\beta_1).[\alpha_1,\beta_1]\langle\!\langle M\rangle\!\rangle\\
&=_{\eta_\vee} \langle\!\langle M\rangle\!\rangle
\end{aligned}
$$

(2) We can show this by induction on the call-by-name equation of the $\lambda\mu_n^{wad}$-calculus. We consider only the rules about sums, *i.e.*, $(\beta_\vee)$, $(\zeta_\vee)$, $(\eta_\vee)$-rules.

$(\beta_\vee)$-rule :

$$
\begin{aligned}
\llbracket[\alpha',\beta']\mu(\alpha,\beta).S\rrbracket &\equiv \delta\big(\llbracket\mu(\alpha,\beta).S\rrbracket, x.[\alpha']x, y.[\beta']y\big)\\
&\equiv \delta\big(\mu\gamma.\llbracket S\rrbracket[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}], x.[\alpha']x, y.[\beta']y\big)\\
&=_{\zeta} \llbracket S\rrbracket[{}^{\delta(\mathrm{inl}\{-\},x.[\alpha']x,y.[\beta']y)}/_{[\alpha]\{-\}}, {}^{\delta(\mathrm{inr}\{-\},x.[\alpha']x,y.[\beta']y)}/_{[\beta]\{-\}}]\\
&=_{\beta_\vee} \llbracket S\rrbracket[{}^{[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\beta']\{-\}}/_{[\beta]\{-\}}] \equiv \llbracket S[{}^{\alpha'}/_{\alpha}, {}^{\beta'}/_{\beta}]\rrbracket
\end{aligned}
$$

$(\zeta_\vee)$-rule :

$$
\begin{aligned}
\llbracket[\alpha,\beta]\mu\gamma.S\rrbracket &\equiv \delta(\llbracket\mu\gamma.S\rrbracket, x.[\alpha]x, y.[\beta]y) \equiv \delta(\mu\gamma.\llbracket S\rrbracket, x.[\alpha]x, y.[\beta]y)\\
&=_{\zeta} \llbracket S\rrbracket[{}^{\delta(\{-\},x.[\alpha]x,y.[\beta]y)}/_{[\gamma]\{-\}}] \overset{(*)}{\equiv} \llbracket S[{}^{[\alpha,\beta]\{-\}}/_{[\gamma]\{-\}}]\rrbracket
\end{aligned}
$$

$(*)$ is shown by induction on terms and statements. The key case is as follows.

$$
\begin{aligned}
&\llbracket[\gamma]M\rrbracket[{}^{\delta(\{-\},x.[\alpha]x,y.[\beta]y)}/_{[\gamma]\{-\}}] \equiv ([\gamma]\llbracket M\rrbracket)[{}^{\delta(\{-\},x.[\alpha]x,y.[\beta]y)}/_{[\gamma]\{-\}}]\\
&\quad\equiv \delta\big(\llbracket M\rrbracket[{}^{\delta(\{-\},x.[\alpha]x,y.[\beta]y)}/_{[\gamma]\{-\}}], x.[\alpha]x, y.[\beta]y\big)\\
&\quad\overset{I.H.}{\equiv} \delta\big(\llbracket M[{}^{[\alpha,\beta]\{-\}}/_{[\gamma]\{-\}}]\rrbracket, x.[\alpha]x, y.[\beta]y\big)\\
&\quad\equiv \llbracket[\alpha,\beta](M[{}^{[\alpha,\beta]\{-\}}/_{[\gamma]\{-\}}])\rrbracket \equiv \llbracket([\gamma]M)[{}^{[\alpha,\beta]\{-\}}/_{[\gamma]\{-\}}]\rrbracket
\end{aligned}
$$

$(\eta_\vee)$-rule :

$$
\begin{aligned}
\llbracket\mu(\alpha,\beta).[\alpha,\beta]M\rrbracket &\equiv \mu\gamma.\llbracket[\alpha,\beta]M\rrbracket[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}]\\
&\equiv \mu\gamma.\delta(\llbracket M\rrbracket, x.[\alpha]x, y.[\beta]y)[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}]\\
&\equiv \mu\gamma.\delta(\llbracket M\rrbracket, x.[\gamma]\mathrm{inl}(x), y.[\gamma]\mathrm{inr}(y))\\
&=_{\pi} \mu\gamma.[\gamma]\delta(\llbracket M\rrbracket, x.\mathrm{inl}(x), y.\mathrm{inr}(y))
\end{aligned}
$$

$$
\begin{aligned}
&=_{\eta_\mu} \delta(\llbracket M \rrbracket, x.\mathrm{inl}(x), y.\mathrm{inr}(y)) \\
&=_{\eta_\vee} \llbracket M \rrbracket
\end{aligned}
$$

(3) We can show this by induction on term M and statement S of the $\lambda\mu_n^\eta$-calculus. We consider the cases of $\mathrm{inl}(M)$, $\mathrm{inr}(M)$, $\delta(O, x.M, y.N)$ and $\delta(O, x.S, y.T)$.
Case of $\mathrm{inl}(M)$:

$$
\begin{aligned}
\llbracket \langle\!\langle \mathrm{inl}(M) \rangle\!\rangle \rrbracket &\equiv \llbracket \mu(\alpha,\beta).[\alpha]\langle\!\langle M \rangle\!\rangle \rrbracket \equiv \mu\gamma.\Big([\alpha]\llbracket \langle\!\langle M \rangle\!\rangle \rrbracket\Big)[^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}] \\
&\equiv \mu\gamma.[\gamma]\mathrm{inl}\Big(\llbracket \langle\!\langle M \rangle\!\rangle \rrbracket\Big) \overset{I.H.}{=}_n \mu\gamma.[\gamma]\mathrm{inl}(M) =_{\eta_\mu} \mathrm{inl}(M)
\end{aligned}
$$

The case of $\mathrm{inr}(M)$ can also be shown similarly.
Case of $\delta(O, x.M, y.N)$:

$$
\begin{aligned}
\llbracket \langle\!\langle \delta(O, x.M, y.N) \rangle\!\rangle \rrbracket &\equiv \llbracket \mu\gamma.\Big(\lambda y.[\gamma]\langle\!\langle N \rangle\!\rangle\Big)\Big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle O \rangle\!\rangle)\Big)\rangle\!\rangle \\
&\equiv \mu\gamma.\Big(\lambda y.[\gamma]\llbracket \langle\!\langle N \rangle\!\rangle \rrbracket\Big)\Big(\mu\beta.(\lambda x.[\gamma]\llbracket \langle\!\langle M \rangle\!\rangle \rrbracket)(\mu\alpha.\llbracket\, [\alpha,\beta]\langle\!\langle O \rangle\!\rangle \,\rrbracket)\Big) \\
&\equiv \mu\gamma.\Big(\lambda y.[\gamma]\llbracket \langle\!\langle N \rangle\!\rangle \rrbracket\Big)\Big(\mu\beta.(\lambda x.[\gamma]\llbracket \langle\!\langle M \rangle\!\rangle \rrbracket)(\mu\alpha.\delta(\llbracket \langle\!\langle O \rangle\!\rangle \rrbracket, x.[\alpha]x, y.[\beta]y))\Big) \\
&\overset{I.H.}{=}_n \mu\gamma.\Big(\lambda y.[\gamma]N\Big)\Big(\mu\beta.(\lambda x.[\gamma]M)(\mu\alpha.\delta(O, x.[\alpha]x, y.[\beta]y))\Big) \\
&\overset{(*)}{=}_n \mu\gamma.\delta(O, x.[\gamma]M, y.[\gamma]N) \\
&=_{(\pi)} \mu\gamma.[\gamma]\delta(O, x.M, y.N) \\
&=_{(\eta_\mu)} \delta(O, x.M, y.N)
\end{aligned}
$$

$(*)$ is shown by case analysis of M and N. If M and N are not simple forms w.r.t. x and y respectively, then we have

$$
\begin{aligned}
\mu\gamma.\delta(O, x.[\gamma]M, y.[\gamma]N) &=_{(\nu)} \mu\gamma.(\lambda y.[\gamma]N)(\mu\beta.\delta(O, x.[\gamma]M, y.[\beta]y)) \\
&=_{(\nu)} \mu\gamma.\Big((\lambda y.[\gamma]N)(\mu\beta.(\lambda x.[\gamma]M)(\mu\alpha.\delta(O, x.[\alpha]x, y.[\beta]y)))\Big)
\end{aligned}
$$

If M is $E_n\{x\}$ and N is not a simple form w.r.t. y, then we have

$$
\begin{aligned}
\mu\gamma.\delta(O, x.[\gamma]E_n\{x\}, y.[\gamma]N) &=_{(\nu)} \mu\gamma.(\lambda y.[\gamma]N)(\mu\beta.\delta(O, x.[\gamma]E_n\{x\}, y.[\beta]y)) \\
&=_{(\zeta)} \mu\gamma.\Big((\lambda y.[\gamma]N)(\mu\beta.[\gamma]E_n\{\mu\alpha.\delta(O, x.[\alpha]x, y.[\beta]y)\})\Big) \\
&=_{(\beta_\supset)} \mu\gamma.\Big((\lambda y.[\gamma]N)(\mu\beta.(\lambda x.[\gamma]E_n\{x\})(\mu\alpha.\delta(O, x.[\alpha]x, y.[\beta]y)))\Big)
\end{aligned}
$$

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 $\lambda\mu_n^{wad}$-calculus. We consider only $\mu(\alpha,\beta).S$ and $[\alpha,\beta]M$.
Case of $\mu(\alpha,\beta).S$:

$$\begin{aligned}
\langle\!\langle [\![\mu(\alpha,\beta).S]\!] \rangle\!\rangle &\equiv \langle\!\langle \mu\gamma.[\![S]\!][{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}] \rangle\!\rangle \\
&\equiv \mu\gamma.\langle\!\langle [\![S]\!][{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}] \rangle\!\rangle \\
&\equiv \mu\gamma.\langle\!\langle [\![S]\!] \rangle\!\rangle[{}^{[\gamma]\langle\langle \mathrm{inl}\{-\}\rangle\rangle}/_{[\alpha]\{-\}}, {}^{[\gamma]\langle\langle \mathrm{inr}\{-\}\rangle\rangle}/_{[\beta]\{-\}}] \\
&\equiv \mu\gamma.\langle\!\langle [\![S]\!] \rangle\!\rangle[{}^{[\gamma]\mu(\alpha',\beta').[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mu(\alpha'',\beta'').[\beta'']\{-\}}/_{[\beta]\{-\}}] \\
&=_{(\eta_\vee)} \mu(\alpha_1,\beta_1).[\alpha_1,\beta_1]\mu\gamma.\langle\!\langle [\![S]\!] \rangle\!\rangle[{}^{[\gamma]\mu(\alpha',\beta').[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mu(\alpha'',\beta'').[\beta'']\{-\}}/_{[\beta]\{-\}}] \\
&=_{(\zeta_\vee)} \mu(\alpha_1,\beta_1).\langle\!\langle [\![S]\!] \rangle\!\rangle[{}^{[\alpha_1,\beta_1]\mu(\alpha',\beta').[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\alpha_1,\beta_1]\mu(\alpha'',\beta'').[\beta'']\{-\}}/_{[\beta]\{-\}}] \\
&=_{(\beta_\vee)} \mu(\alpha,\beta).\langle\!\langle [\![S]\!] \rangle\!\rangle[{}^{[\alpha_1]\{-\}}/_{[\alpha]\{-\}}, {}^{[\beta_1]\{-\}}/_{[\beta]\{-\}}] \\
&\equiv \mu(\alpha,\beta).\langle\!\langle [\![S]\!] \rangle\!\rangle \\
&\overset{I.H.}{=_n} \mu(\alpha,\beta).S
\end{aligned}$$

Case of $[\alpha,\beta]M$:

$$\begin{aligned}
\langle\!\langle [\![[\alpha,\beta]M]\!] \rangle\!\rangle &\equiv \langle\!\langle\, \delta([\![M]\!], x.[\alpha]x, y.[\beta]y) \,\rangle\!\rangle \\
&\equiv \big((\lambda y.[\beta]y)(\mu\beta'.(\lambda x.[\alpha]x)(\mu\alpha'.[\alpha',\beta']\langle\!\langle [\![M]\!] \rangle\!\rangle))\big) \\
&=_{(\beta_\supset)} [\beta]\mu\beta'.[\alpha]\mu\alpha'.[\alpha',\beta']\langle\!\langle [\![M]\!] \rangle\!\rangle \\
&=_{(\beta_\mu)} [\alpha,\beta]\langle\!\langle [\![M]\!] \rangle\!\rangle \\
&\overset{I.H.}{=_n} [\alpha,\beta]M
\end{aligned}$$

□

2.2.2 The call-by-value $\lambda\mu$-calculus

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

$$V, W ::= x \mid \lambda x.M \mid \langle V, W\rangle \mid \mathrm{inl}(V) \mid \mathrm{inr}(W) \mid \lambda x.S$$

We also use a notion of the call-by-value evaluation 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.*, $(\lambda x.M)\{-\}$. 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.*, $(\lambda x.S)\{-\}$.
Call-by-value evaluation term and statement contexts

$$\begin{aligned}
E_v &::= (\lambda x.M)\{-\} \mid E_e \mid E_i \\
E_e &::= \{-\}M \mid \mathrm{fst}(\{-\}) \mid \mathrm{snd}(\{-\}) \mid \delta(\{-\}, x.M, y.N) \\
E_i &::= \mathrm{inl}(\{-\}) \mid \mathrm{inr}(\{-\}) \mid \langle \{-\}, M\rangle \mid \langle V, \{-\}\rangle \\
D_v &::= (\lambda x.S)\{-\} \mid D_e \\
D_e &::= [\alpha]\{-\} \mid \{-\}M \mid \delta(\{-\}, x.S, y.T)
\end{aligned}$$

The one-step *call-by-value reduction* relation for the $\lambda\mu$-calculus, denoted by $\longrightarrow_v$, is defined as the compatible closure of the rules in figure 2.3. We write $\longrightarrow_v{}^*$ and $\longrightarrow_v{}^+$ for the reflexive transitive closure and the transitive closure of $\longrightarrow_v$ respectively. (β)-rules reduce the deconstructor applied to a constructor with call-by-value restrictions, (ζ)-rules substitute a call-by-value evaluation context and a statement context for a covalue, (η_μ)-rule introduces a μ-abstraction applied to a covariable application, and (π)-rules correspond to the permutative conversions. The (*name*)-rules push the next term to be evaluated out as an argument of the function. These rules correspond to the (*name*)-rule of the $\lambda\mu_v^{wad}$-calculus. The (*comp*)-rules are associativity rules, which correspond to the (*comp*)-rule of the $\lambda\mu_v^{wad}$-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,
- the formulation of sums in our system is different from that in $\lambda\mu_v^{wad}$,
- we defined values differently: a projection from a value is not a value in our system, and
- our system does not have (η)-rules related to implications, negations, pairs, and sums while his system does have them.

$(\beta_{\supset})$	$(\lambda x.M)V \longrightarrow_v M[{}^{V}/_{x}]$	
$(\beta_{\wedge})$	$\mathrm{fst}\langle V, W\rangle \longrightarrow_v V$	
	$\mathrm{snd}\langle V, W\rangle \longrightarrow_v W$	
$(\beta_{\vee})$	$\delta(\mathrm{inl}(V), x.M, y.N) \longrightarrow_v M[{}^{V}/_{x}]$	
	$\delta(\mathrm{inr}(V), x.M, y.N) \longrightarrow_v N[{}^{V}/_{x}]$	
	$\delta(\mathrm{inl}(V), x.S, y.T) \longrightarrow_v S[{}^{V}/_{x}]$	
	$\delta(\mathrm{inr}(V), x.S, y.T) \longrightarrow_v T[{}^{V}/_{x}]$	
$(\beta_{\neg})$	$(\lambda x.S)V \longrightarrow_v S[{}^{V}/_{x}]$	
(ζ)	$E_{e\lambda}\{\mu\alpha.S\} \longrightarrow_v \mu\beta.S[{}^{[\beta]E_{e\lambda}\{-\}}/_{[\alpha]\{-\}}]$	(where $E_{e\lambda}$ is E_e or $(\lambda x.M)\{-\}$)
	$D_v\{\mu\alpha.S\} \longrightarrow_v S[{}^{D_v\{-\}}/_{[\alpha]\{-\}}]$	
$(comp)$	$E_{e\lambda}\{(\lambda x.M)N\} \longrightarrow_v (\lambda x.E_{e\lambda}\{M\})N$	
	$D_v\{(\lambda x.M)N\} \longrightarrow_v (\lambda x.D_v\{M\})N$	
(π)	$E_{e\lambda}\{\delta(O, x.M, y.N)\} \longrightarrow_v \delta(O, x.E_{e\lambda}\{M\}, y.E_{e\lambda}\{N\})$	
	$D_v\{\delta(O, x.M, y.N)\} \longrightarrow_v \delta(O, x.D_v\{M\}, y.D_v\{N\})$	
(η_{μ})	$M \longrightarrow_v \mu\alpha.[\alpha]M$	(where $\alpha \notin \mathrm{FCV}(M)$)
$(name)$	$E_{ie}\{O\} \longrightarrow_v (\lambda x.E_{ie}\{x\})O$	(where O is not a value, E_{ie} is E_i or E_e)
	$D_e\{O\} \longrightarrow_v (\lambda x.D_e\{x\})O$	(where O is not a value)

Figure 2.3: Reduction rules of the call-by-value $\lambda\mu$-calculus

We introduce the call-by-value calculus $\lambda\mu_v^\eta$ as the system generated by the rules in figure 2.3 and the following (η)-rules.

$$
\begin{array}{llr}
(\eta_\supset) & V \longrightarrow_n \lambda x.Vx & (V : A \supset B) \\
(\eta_\wedge) & V \longrightarrow_n \langle \mathrm{fst}(V), \mathrm{snd}(V) \rangle & (V : A \wedge B) \\
(\eta_\vee) & M \longrightarrow_n \delta(M, x.\mathrm{inl}(x), y.\mathrm{inr}(y)) & (M : A \vee B) \\
(\eta_\neg) & V \longrightarrow_n \lambda x.Vx & (V : \neg A)
\end{array}
$$

We define the $\lambda\mu_v^{wad-}$-calculus as the restricted system of the $\lambda\mu_v^{wad}$-calculus obtained by excluding a projection from a value from definition of values. We again consider translations $\langle\!\langle - \rangle\!\rangle$ and $[\![-]\!]$ given in the previous subsection. Our call-by-value system $\lambda\mu_v^\eta$ with (η)-rules is equivalent to the $\lambda\mu_v^{wad-}$-calculus as the equational systems.

Proposition 2.3

(1) If $\lambda\mu_v^\eta \vdash M =_v N$, then $\lambda\mu_v^{wad-} \vdash \langle\!\langle M \rangle\!\rangle =_v \langle\!\langle N \rangle\!\rangle$, and if $\lambda\mu_v^\eta \vdash S =_n T$, then $\lambda\mu_v^{wad-} \vdash \langle\!\langle S \rangle\!\rangle =_v \langle\!\langle T \rangle\!\rangle$.

(2) If $\lambda\mu_v^{wad-} \vdash M =_v N$, then $\lambda\mu_v^\eta \vdash [\![M]\!] =_v [\![N]\!]$, and if $\lambda\mu_v^{wad-} \vdash S =_v T$, then $\lambda\mu_v^\eta \vdash [\![S]\!] =_v [\![T]\!]$.

(3) $\lambda\mu_v^\eta \vdash [\![\langle\!\langle M \rangle\!\rangle]\!] =_v M$ and $\lambda\mu_v^\eta \vdash [\![\langle\!\langle S \rangle\!\rangle]\!] =_v S$.

(4) $\lambda\mu_v^{wad-} \vdash \langle\!\langle [\![M]\!] \rangle\!\rangle =_v M$ and $\lambda\mu_v^{wad-} \vdash \langle\!\langle [\![S]\!] \rangle\!\rangle =_v S$.

Proof. (1) We can show this by induction on the call-by-value equation of the $\lambda\mu_v^\eta$-calculus. We consider only the rules about sums, *i.e.*, $(\beta_\vee)$, (ζ), (π), $(comp)$, $(name)$, and $(\eta_\vee)$-rules.
Case of $(\beta_\vee)$-rule :

$$
\begin{aligned}
\langle\!\langle \delta(\mathrm{inl}(V), x.M, y.N) \rangle\!\rangle &\equiv \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle N \rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle \mathrm{inl}(V) \rangle\!\rangle)\big) \\
&\equiv \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle N \rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\mu(\alpha',\beta').[\alpha']\langle\!\langle V \rangle\!\rangle)\big) \\
&=_{(\beta_\vee)} \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle N \rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)(\mu\alpha.[\alpha]\langle\!\langle V \rangle\!\rangle)\big) \\
&=_{(\eta_\mu)} \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle N \rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)\langle\!\langle V \rangle\!\rangle\big) \\
&=_{(\zeta)} \mu\gamma.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)\langle\!\langle V \rangle\!\rangle \\
&=_{(\beta_\neg)} \mu\gamma.[\gamma]\langle\!\langle M \rangle\!\rangle[{}^{\langle\langle V \rangle\rangle}/_x] \\
&\overset{(*)}{\equiv} \mu\gamma.[\gamma]\langle\!\langle M[{}^V/_x] \rangle\!\rangle \\
&=_{\eta_\mu} \langle\!\langle M[{}^V/_x] \rangle\!\rangle
\end{aligned}
$$

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

Case of (ζ)-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 $\delta(\{-\}, x.M, y.N)$:

$$\begin{aligned}
\langle\langle\delta(\mu\alpha.S, x.M, y.N)\rangle\rangle &\equiv \mu\gamma.\big(\lambda y.[\gamma]\langle\langle N\rangle\rangle\big)\big(\mu\beta'.(\lambda x.[\gamma]\langle\langle M\rangle\rangle)\mu\alpha'.[\alpha',\beta']\mu\alpha.\langle\langle S\rangle\rangle\big)\\
&=_{(\zeta)} \mu\gamma.\big(\lambda y.[\gamma]\langle\langle N\rangle\rangle\big)\big(\mu\beta'.(\lambda x.[\gamma]\langle\langle M\rangle\rangle)\mu\alpha'.\langle\langle S\rangle\rangle[{}^{[\alpha',\beta']\{-\}}/_{[\alpha]\{-\}}]\big)\\
&=_{(\zeta)} \mu\gamma.\langle\langle S\rangle\rangle[{}^{[\alpha',\beta']\{-\}}/_{[\alpha]\{-\}}][{}^{(\lambda x.[\gamma]\langle\langle M\rangle\rangle)\{-\}}/_{[\alpha']\{-\}}][{}^{(\lambda y.[\gamma]\langle\langle N\rangle\rangle)\{-\}}/_{[\beta']\{-\}}]\\
&\equiv \mu\gamma.\langle\langle S\rangle\rangle[{}^{([\alpha',\beta']\{-\})[{}^{(\lambda x.[\gamma]\langle\langle M\rangle\rangle)\{-\}}/_{[\alpha']\{-\}}][{}^{(\lambda y.[\gamma]\langle\langle N\rangle\rangle)\{-\}}/_{[\beta']\{-\}}]}/_{[\alpha]\{-\}}]\\
&\equiv \mu\gamma.\langle\langle S\rangle\rangle[{}^{(\lambda y.[\gamma]\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma]\langle\langle M\rangle\rangle)(\mu\alpha'.[\alpha',\beta']\{-\}))}/_{[\alpha]\{-\}}]\\
&=_{(\beta_\mu)} \mu\gamma.\langle\langle S\rangle\rangle[{}^{[\gamma]\mu\gamma'.(\lambda y.[\gamma']\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma']\langle\langle M\rangle\rangle)(\mu\alpha'.[\alpha',\beta']\{-\}))}/_{[\alpha]\{-\}}]\\
&\overset{(*)}{\equiv} \mu\gamma.\langle\langle S[{}^{[\gamma]\delta(\{-\},x.M,y.N)}/_{[\alpha]\{-\}}]\rangle\rangle \equiv \langle\langle\mu\gamma.S[{}^{[\gamma]\delta(\{-\},x.M,y.N)}/_{[\alpha]\{-\}}]\rangle\rangle
\end{aligned}$$

(*) is shown by a straightforward induction on terms and statements. We abbreviate $M[{}^{[\gamma]\mu\gamma'.(\lambda y.[\gamma']\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma']\langle\langle M\rangle\rangle)(\mu\alpha'.[\alpha',\beta']\{-\}))}/_{[\alpha]\{-\}}]$ and $S[{}^{[\gamma]\mu\gamma'.(\lambda y.[\gamma']\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma']\langle\langle M\rangle\rangle)(\mu\alpha'.[\alpha',\beta']\{-\}))}/_{[\alpha]\{-\}}]$ in $\widetilde{M}$ and $\widetilde{S}$ respectively. The key case is proved in this way.

$$\begin{aligned}
&\langle\langle\widetilde{[\alpha]O}\rangle\rangle \equiv [\alpha]\widetilde{\langle\langle O\rangle\rangle} \equiv [\gamma]\mu\gamma'.(\lambda y.[\gamma']\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma']\langle\langle M\rangle\rangle)(\mu\alpha'.[\alpha',\beta']\widetilde{\langle\langle O\rangle\rangle}))\\
&\overset{I.H.}{\equiv} [\gamma]\mu\gamma'.(\lambda y.[\gamma']\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma']\langle\langle M\rangle\rangle)(\mu\alpha'.[\alpha',\beta']\langle\langle O[{}^{[\gamma]\delta(\{-\},x.M,y.N)}/_{[\alpha]\{-\}}]\rangle\rangle))\\
&\equiv \langle\langle\, [\gamma]\delta(O[{}^{[\gamma]\delta(\{-\},x.M,y.N)}/_{[\alpha]\{-\}}], x.M, y.N)\,\rangle\rangle\\
&\equiv \langle\langle\, ([\alpha]O)[{}^{[\gamma]\delta(\{-\},x.M,y.N)}/_{[\alpha]\{-\}}]\,\rangle\rangle
\end{aligned}$$

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

$$\begin{aligned}
E\{\delta(O, x.M, y.N)\} &=_{(\eta_\mu)} E\{\mu\alpha.[\alpha]\delta(O, x.M, y.N)\} =_v E\{\mu\alpha.\delta(O, x.[\alpha]M, y.[\alpha]N)\}\\
&=_{(\zeta)} \mu\beta.\delta(O, x.[\beta]E\{M\}, y.[\beta]E\{N\})\} =_v \mu\beta.[\beta]\delta(O, x.E\{M\}, y.E\{N\})\}\\
&=_{(\eta_\mu)} \delta(O, x.E\{M\}, y.E\{N\})\}.
\end{aligned}$$

We can obtain $D_v\{\delta(O, x.M, y.N)\} =_v \delta(O, x.D_v\{M\}, y.D_v\{N\})$ in a similar way. Therefore, it is sufficient to prove $\langle\langle[\alpha]\delta(O, x.M, y.N)\rangle\rangle =_v \langle\langle\delta(O, x.[\alpha]M, y.[\alpha]N)\rangle\rangle$.

$$\begin{aligned}
\langle\langle[\alpha]\delta(O, x.M, y.N)\rangle\rangle &\equiv [\alpha]\mu\gamma.(\lambda y.[\gamma]\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\gamma]\langle\langle M\rangle\rangle)\mu\alpha'.[\alpha',\beta']\langle\langle O\rangle\rangle)\\
&=_{(\beta_\mu)} (\lambda y.[\alpha]\langle\langle N\rangle\rangle)(\mu\beta'.(\lambda x.[\alpha]\langle\langle M\rangle\rangle)\mu\alpha'.[\alpha',\beta']\langle\langle O\rangle\rangle)
\end{aligned}$$

$$\equiv \langle\!\langle\, \delta(O, x.[\alpha]M, y.[\alpha]N) \,\rangle\!\rangle$$

Case of (*comp*)-rule : To show this case, it is sufficient to have $\langle\!\langle[\alpha]((\lambda x.M)N)\rangle\!\rangle =_v \langle\!\langle(\lambda x.[\alpha]M)N\rangle\!\rangle$ by the discussion similar to that for (π)-rule.

$$\begin{aligned}\langle\!\langle\, [\alpha]((\lambda x.M)N) \,\rangle\!\rangle &=_v [\alpha]((\lambda x.\langle\!\langle M\rangle\!\rangle)\langle\!\langle N\rangle\!\rangle \\ &=_{(comp)} (\lambda x.[\alpha]\langle\!\langle M\rangle\!\rangle)\langle\!\langle N\rangle\!\rangle \equiv \langle\!\langle\, (\lambda x.[\alpha]M)N \,\rangle\!\rangle\end{aligned}$$

Case of (*name*)-rule : We can easily show $\lambda\mu_v^{wad-} \vdash \langle\!\langle E_i\{O\}\rangle\!\rangle =_v \langle\!\langle(\lambda x.E_i\{x\})O\rangle\!\rangle$ by a case analysis of E_i. On the other hand, we have

$$\langle\!\langle[\alpha]O\rangle\!\rangle \equiv [\alpha]\langle\!\langle O\rangle\!\rangle =_{(comp)} (\lambda x.[\alpha]x)\langle\!\langle O\rangle\!\rangle \equiv \langle\!\langle\, (\lambda x.[\alpha]x)O \,\rangle\!\rangle,$$

therefore we obtain $\lambda\mu_v^{wad-} \vdash \langle\!\langle E_e\{O\}\rangle\!\rangle =_v \langle\!\langle(\lambda x.E_e\{x\})O\rangle\!\rangle$, since $\lambda\mu_v \vdash E_e\{O\} =_v (\lambda x.E_e\{x\})O$ can be shown from $\lambda\mu \vdash [\alpha]O =_v (\lambda x.[\alpha]x)O$ with (ζ), (η_μ) and (*comp*)-rules as follows.

$$\begin{aligned}E_e\{O\} &=_{(\eta_\mu)} E_e\{\mu\alpha.[\alpha]O\} =_v E_e\{\mu\alpha.(\lambda x.[\alpha]x)O\} =_{(\zeta)} \mu\beta.((\lambda x.[\beta]E_e\{x\})O) \\ &=_{(name)} \mu\beta.[\beta]((\lambda x.E_e\{x\})O) =_{(\eta_\mu)} (\lambda x.E_e\{x\})O\end{aligned}$$

Case of ($\eta_\vee$)-rule :

$$\begin{aligned}&\langle\!\langle\delta(M, x.\mathrm{inl}(x), y.\mathrm{inr}(y))\rangle\!\rangle \equiv \mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle\mathrm{inr}(y)\rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle\mathrm{inr}(x)\rangle\!\rangle)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big) \\ &\equiv \mu\gamma.\big(\lambda y.[\gamma]\mu(\alpha'',\beta'').[\beta'']y\big)\big(\mu\beta.(\lambda x.[\gamma]\mu(\alpha',\beta').[\alpha']x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big) \\ &=_{(\eta_\vee)} \mu(\alpha_1,\beta_1).[\alpha_1,\beta_1]\mu\gamma.\big(\lambda y.[\gamma]\mu(\alpha'',\beta'').[\beta'']y\big)\big(\mu\beta.(\lambda x.[\gamma]\mu(\alpha',\beta').[\alpha']x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big) \\ &=_{(\zeta_\vee)} \mu(\alpha_1,\beta_1).\big(\lambda y.[\alpha_1,\beta_1]\mu(\alpha'',\beta'').[\beta'']y\big)\big(\mu\beta.(\lambda x.[\alpha_1,\beta_1]\mu(\alpha',\beta').[\alpha']x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big) \\ &=_{(\beta_\vee)} \mu(\alpha_1,\beta_1).\big(\lambda y.[\beta_1]y\big)\big(\mu\beta.(\lambda x.[\alpha_1]x)(\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle)\big) \\ &=_{(name)} \mu(\alpha_1,\beta_1).[\beta_1]\mu\beta.[\alpha_1]\mu\alpha.[\alpha,\beta]\langle\!\langle M\rangle\!\rangle \\ &=_{(\beta_\mu)} \mu(\alpha_1,\beta_1).[\alpha_1,\beta_1]\langle\!\langle M\rangle\!\rangle \\ &=_{(\eta_\vee)} \langle\!\langle M\rangle\!\rangle\end{aligned}$$

(2) We can show this by induction on the call-by-name equation of the $\lambda\mu_n^{wad}$-calculus. We consider only the rules about sums, *i.e.*, ($\beta_\vee$), (ζ), ($\eta_\vee$), (*name*), and (*comp*)-rules.
Case of ($\beta_\vee$)-rule :

$$\begin{aligned}[\![[\alpha',\beta']\mu(\alpha,\beta).S]\!] &\equiv \delta\big([\![\mu(\alpha,\beta).S]\!], x.[\alpha']x, y.[\beta']y\big) \\ &\equiv \delta\big(\mu\gamma.[\![S]\!][{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}], x.[\alpha']x, y.[\beta']y\big) \\ &=_{(\zeta)} [\![S]\!][{}^{\delta(\mathrm{inl}\{-\},x.[\alpha']x,y.[\beta']y)}/_{[\alpha]\{-\}}, {}^{\delta(\mathrm{inr}\{-\},x.[\alpha']x,y.[\beta']y)}/_{[\beta]\{-\}}] \\ &\overset{(*)}{=}_v [\![S]\!][{}^{[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\beta']\{-\}}/_{[\beta]\{-\}}] \equiv [\![S[{}^{\alpha'}/_{\alpha}, {}^{\beta'}/_{\beta}]]\!]\end{aligned}$$

$(*)$ can be shown by the following claim: $\delta(\mathrm{inl}(M), x.[\alpha]x, y.[\beta]y) =_v [\alpha]M$ for any M. We show this. If M is a value, then the claim is obtained by $(\beta_\vee)$-rule. If M is not a value, then

$$\begin{aligned}
&\delta(\mathrm{inl}(M), x.[\alpha]x, y.[\beta]y) =_{(name)} (\lambda z.\delta(z, x.[\alpha]x, y.[\beta]y))\mathrm{inl}(M)\\
&\quad =_{(name)} \Big(\lambda z.\delta(z, x.[\alpha]x, y.[\beta]y)\Big)((\lambda z'.\mathrm{inl}(z'))M)\\
&\quad =_{(comp)} \Big(\lambda z'.(\lambda z.\delta(z, x.[\alpha]x, y.[\beta]y))\mathrm{inl}(z')\Big)M\\
&\quad =_{(\beta_\supset)} (\lambda z'.\delta(\mathrm{inl}(z'), x.[\alpha]x, y.[\beta]y))M\\
&\quad =_{(\beta_\vee)} (\lambda z'.[\alpha]z')M =_{(name)} [\alpha]M
\end{aligned}$$

Case of (ζ)-rule : to show this case, we introduce evaluation singular context E_w and singular statement context D_w of the $\lambda\mu_v^{wad-}$-calculus.

$$\begin{aligned}
E_w &::= \{-\}N \mid V\{-\} \mid \langle V, \{-\}\rangle \mid \langle\{-\}, M\rangle \mid \mathrm{fst}(\{-\}) \mid \mathrm{snd}(\{-\})\\
D_w &::= [\alpha]\{-\} \mid [\alpha,\beta]\{-\} \mid \{-\}M \mid V\{-\}
\end{aligned}$$

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

$$[\![E_w\{\mu\alpha.S\}]\!] =_v [\![\mu\beta.S\,[^{[\beta]E_w}/_{[\alpha]\{-\}}]]\!] \text{ and } [\![D_w\{\mu\alpha.S\}]\!] =_v [\![S\,[^{D_w\{-\}}/_{[\alpha]\{-\}}]]\!],$$

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 $x\{-\}$:

$$\begin{aligned}
[\![x\mu\alpha.S]\!] \equiv x\mu\alpha.[\![S]\!] =_{(\eta_\supset)} (\lambda z.xz)\mu\alpha.[\![S]\!] =_{(\zeta)} \mu\beta.[\![S]\!][^{[\beta](\lambda z.xz)\{-\}}/_{[\alpha]\{-\}}]\\
=_{(\eta_\supset)} \mu\beta.[\![S]\!][^{[\beta]x\{-\}}/_{[\alpha]\{-\}}] \overset{(*)}{\equiv} \mu\beta.[\![S\,[^{[\beta]x\{-\}}/_{[\alpha]\{-\}}]]\!]
\end{aligned}$$

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

$$\begin{aligned}
&[\![[\alpha]M]\!][^{[\beta]x\{-\}}/_{[\alpha]\{-\}}] \equiv ([\alpha][\![M]\!])[^{[\beta]x\{-\}}/_{[\alpha]\{-\}}] \equiv [\beta]x([\![M]\!][^{[\beta]x\{-\}}/_{[\alpha]\{-\}}])\\
&\overset{I.H.}{\equiv} [\beta]x([\![M[^{[\beta]x\{-\}}/_{[\alpha]\{-\}}]]\!] \equiv [\![([\alpha]M)[^{[\beta]x\{-\}}/_{[\alpha]\{-\}}]]\!]
\end{aligned}$$

- D_w is $x\{-\}$: this can be shown in a way similar to the above case.
- D_w is $[\alpha,\beta]\{-\}$:

$$\begin{aligned}
[\![[\alpha,\beta]\mu\gamma.S]\!] \equiv \delta([\![\mu\gamma.S]\!], x.[\alpha]x, y.[\beta]y) \equiv \delta(\mu\gamma.[\![S]\!], x.[\alpha]x, y.[\beta]y)\\
=_{(\zeta)} [\![S]\!][^{\delta(\{-\},x.[\alpha]x,y.[\beta]y)}/_{[\gamma]\{-\}}] \overset{(*)}{\equiv} [\![S\,[^{[\alpha,\beta]\{-\}}/_{[\gamma]\{-\}}]]\!]
\end{aligned}$$

$(*)$ is already shown in the proof of Proposition 2.2.
Case of $(\eta_\vee)$-rule :

$$\begin{aligned}
[\![\mu(\alpha,\beta).[\alpha,\beta]M]\!] &\equiv \mu\gamma.[\![[\alpha,\beta]M]\!][{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}] \\
&\equiv \mu\gamma.\delta([\![M]\!], x.[\alpha]x, y.[\beta]y)[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}] \\
&\equiv \mu\gamma.\delta([\![M]\!], x.[\gamma]\mathrm{inl}(x), y.[\gamma]\mathrm{inr}(y)) \\
&=_{(\pi)} \mu\gamma.[\gamma]\delta([\![M]\!], x.\mathrm{inl}(x), y.\mathrm{inr}(y)) \\
&=_{(\eta_\mu)} \delta([\![M]\!], x.\mathrm{inl}(x), y.\mathrm{inr}(y)) \\
&=_{(\eta_\vee)} [\![M]\!]
\end{aligned}$$

Case of $(name)$-rule : Let D be a statement context of the $\lambda\mu_v^{wad-}$-calculus, then we can obtain $(name)$-rule of the $\lambda\mu_v^{wad-}$-calculus from $\lambda\mu_v^{wad-} \vdash (\lambda x.[\alpha]x)M =_v [\alpha]M$ with (η_μ), and (ζ)-rule in the following way.

$$D\{M\} =_{(\eta_\mu)} D\{\mu\alpha.[\alpha]M\} =_v D\{\mu\alpha.(\lambda x.[\alpha]x)M\} =_{(\zeta)} (\lambda x.D\{x\})M$$

Therefore, it is sufficient to prove $(\lambda x.[\alpha]x)M]\!] =_v [\![[\alpha]M]\!]$ in our call-by-value $\lambda\mu$-calculus.

$$[\![(\lambda x.[\alpha]x)M]\!] \equiv (\lambda x.[\alpha]x)[\![M]\!] =_{(name)} [\![[\alpha]M]\!]$$

Case of $(comp)$-rule : Let D be a statement context of the $\lambda\mu_v^{wad-}$-calculus, then we can obtain $(comp)$-rule of the $\lambda\mu_v^{wad-}$-calculus from $[\alpha]((\lambda x.M)N) =_v (\lambda x.[\alpha]M)N$, (η_μ), and (ζ)-rule in the following way.

$$D\{(\lambda x.M)N\} =_{(\eta_\mu)} D\{\mu\alpha.[\alpha](\lambda x.M)N\} =_v D\{\mu\alpha.(\lambda x.[\alpha]M)N\} =_{(\zeta)} \lambda x.D\{M\})N$$

Therefore, it is sufficient to prove $[\![[\alpha]((\lambda x.M)N)]\!] =_v [\![\lambda x.[\alpha]N]\!]$.

$$[\![[\alpha]((\lambda x.M)N)]\!] \equiv [\alpha]((\lambda x.[\![M]\!])[\![N]\!]) =_{(comp)} (\lambda x.[\alpha][\![M]\!])[\![N]\!] \equiv [\![(\lambda x.[\alpha]M)N]\!]$$

(3) We can show this by induction on term M and statement S of the $\lambda\mu_v^\eta$-calculus. We consider $\mathrm{inl}(M)$, $\mathrm{inr}(M)$, $\delta(O, x.M, y.N)$ and $\delta(O, x.S, y.T)$.
Case of $\mathrm{inl}(M)$:

$$\begin{aligned}
[\![\langle\!\langle \mathrm{inl}(M)\rangle\!\rangle]\!] \equiv [\![\mu(\alpha,\beta).[\alpha]\langle\!\langle M\rangle\!\rangle]\!] &\equiv \mu\gamma.\Big([\alpha][\![\langle\!\langle M\rangle\!\rangle]\!]\Big)[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}] \\
&\equiv \mu\gamma.[\gamma]\mathrm{inl}\Big([\![\langle\!\langle M\rangle\!\rangle]\!]\Big) \overset{I.H.}{=_v} \mu\gamma.[\gamma]\mathrm{inl}(M) =_{\eta_\mu} \mathrm{inl}(M)
\end{aligned}$$

$\mathrm{inr}(M)$ can be shown similarly.

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

$$\begin{aligned}
\llbracket\langle\langle\delta(O, x.M, y.N)\rangle\rangle\rrbracket &\equiv \llbracket\mu\gamma.\big(\lambda y.[\gamma]\langle\langle N\rangle\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\langle M\rangle\rangle)(\mu\alpha.[\alpha,\beta]\langle\langle O\rangle\rangle)\big)\rangle\rangle \\
&\equiv \mu\gamma.\big(\lambda y.[\gamma]\llbracket\langle\langle N\rangle\rangle\rrbracket\big)\big(\mu\beta.(\lambda x.[\gamma]\llbracket\langle\langle M\rangle\rangle\rrbracket)(\mu\alpha.\llbracket\, [\alpha,\beta]\langle\langle O\rangle\rangle\, \rrbracket)\big) \\
&\equiv \mu\gamma.\big(\lambda y.[\gamma]\llbracket\langle\langle N\rangle\rangle\rrbracket\big)\big(\mu\beta.(\lambda x.[\gamma]\llbracket\langle\langle M\rangle\rangle\rrbracket)(\mu\alpha.\delta(\llbracket\langle\langle O\rangle\rangle\rrbracket, x.[\alpha]x, y.[\beta]y))\big) \\
&\overset{I.H.}{=}_{v} \mu\gamma.\big(\lambda y.[\gamma]N\big)\big(\mu\beta.(\lambda x.[\gamma]M)(\mu\alpha.\delta(O, x.[\alpha]x, y.[\beta]y))\big) \\
&=_{(\zeta)} \mu\gamma.\delta(O, x.(\lambda x.[\gamma]M)x, y.(\lambda y.[\gamma]N)y) \\
&=_{(\beta_\neg)} \mu\gamma.\delta(O, x.[\gamma]M, y.[\gamma]N) \\
&=_{(\pi)} \mu\gamma.[\gamma]\delta(O, x.M, y.N) \\
&=_{(\eta_\mu)} \delta(O, x.M, y.N)
\end{aligned}$$

The other 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 $\lambda\mu_v^{wad-}$-calculus. We consider only $\mu(\alpha,\beta).S$ and $[\alpha,\beta]M$.
Case of $\mu(\alpha,\beta).S$:

$$\begin{aligned}
\langle\langle\llbracket\mu(\alpha,\beta).S\rrbracket\rangle\rangle &\equiv \langle\langle\mu\gamma.\llbracket S\rrbracket[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}]\rangle\rangle \\
&\equiv \mu\gamma.\langle\langle\llbracket S\rrbracket[{}^{[\gamma]\mathrm{inl}\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}\{-\}}/_{[\beta]\{-\}}]\rangle\rangle \\
&\equiv \mu\gamma.\langle\langle\llbracket S\rrbracket\rangle\rangle[{}^{[\gamma]\llbracket\mathrm{inl}\{-\}\rrbracket}/_{[\alpha]\{-\}}, {}^{[\gamma]\llbracket\mathrm{inr}\{-\}\rrbracket}/_{[\beta]\{-\}}] \\
&\equiv \mu\gamma.\langle\langle\llbracket S\rrbracket\rangle\rangle[{}^{[\gamma]\mu(\alpha',\beta').[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mu(\alpha'',\beta'').[\beta'']\{-\}}/_{[\beta]\{-\}}] \\
&=_{(\eta_\vee)} \mu(\alpha_1,\beta_1).[\alpha_1,\beta_1]\mu\gamma.\langle\langle\llbracket S\rrbracket\rangle\rangle[{}^{[\gamma]\mu(\alpha',\beta').[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\gamma]\mu(\alpha'',\beta'').[\beta'']\{-\}}/_{[\beta]\{-\}}] \\
&=_{(\zeta_\vee)} \mu(\alpha_1,\beta_1).\langle\langle\llbracket S\rrbracket\rangle\rangle[{}^{[\alpha_1,\beta_1]\mu(\alpha',\beta').[\alpha']\{-\}}/_{[\alpha]\{-\}}, {}^{[\alpha_1,\beta_1]\mu(\alpha'',\beta'').[\beta'']\{-\}}/_{[\beta]\{-\}}] \\
&=_{(\beta_\vee)} \mu(\alpha,\beta).\langle\langle\llbracket S\rrbracket\rangle\rangle[{}^{[\alpha_1]\{-\}}/_{[\alpha]\{-\}}, {}^{[\beta_1]\{-\}}/_{[\beta]\{-\}}] \\
&\equiv \mu(\alpha,\beta).\langle\langle\llbracket S\rrbracket\rangle\rangle \\
&\overset{I.H.}{=}_{v} \mu(\alpha,\beta).S
\end{aligned}$$

Case of $[\alpha,\beta]M$:

$$\begin{aligned}
\langle\langle\llbracket[\alpha,\beta]M\rrbracket\rangle\rangle &\equiv \langle\langle\delta(\llbracket M\rrbracket, x.[\alpha]x, y.[\beta]y)\rangle\rangle \\
&\equiv \big((\lambda y.[\beta]y)(\mu\beta'.(\lambda x.[\alpha]x)(\mu\alpha'.[\alpha',\beta']\langle\langle\llbracket M\rrbracket\rangle\rangle))\big) \\
&=_{(name)} [\beta]\mu\beta'.[\alpha]\mu\alpha'.[\alpha',\beta']\langle\langle\llbracket M\rrbracket\rangle\rangle \\
&=_{(\beta_\mu)} [\alpha,\beta]\langle\langle\llbracket M\rrbracket\rangle\rangle
\end{aligned}$$

$$\overset{I.H.}{=}_v [\alpha,\beta]M$$

□

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

Lemma 2.4 (Substitution lemma for the $\lambda\mu$-calculus)
Let M and N be terms, and S and T be statements of the $\lambda\mu$-calculus.

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

(2) Let $\mathcal{T}\{-\}$ be a statement context, that is, a statement with a single hole, and suppose $\Gamma, x : A \mid \mathcal{T}\{x\} \vdash_{\lambda\mu} \Delta$,then
if $\Gamma \vdash_{\lambda\mu} \Delta, \alpha : A \mid N : B$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid N[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}] : B$, and
if $\Gamma, \mid S \vdash_{\lambda\mu} \Delta, \alpha : A$, then $\Gamma \mid S[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}] \vdash_{\lambda\mu} \Delta$.

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 $S \equiv [\alpha]M$. Suppose $\Gamma \mid [\alpha]M \vdash_{\lambda\mu} \Delta, \alpha : A$ is derived. Since the last rule to obtain this sequent is $(pass)$, we obtain $\Gamma \vdash_{\lambda\mu} \Delta, \alpha : A \mid M : A$. Hence we have $\Gamma \vdash_{\lambda\mu} \Delta \mid M[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}] : A$ by the induction hypothesis, and then $\Gamma \mid \mathcal{T}\{M[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}]\} \vdash_{\lambda\mu} \Delta$ by (1). This means $\Gamma \mid ([\alpha]M)[{}^{\mathcal{T}\{-\}}/_{[\alpha]\{-\}}] \vdash_{\lambda\mu} \Delta$. □

Proposition 2.5 (Subject reduction for the $\lambda\mu$-calculus)
Let M and N be terms, and S and T be statements of the $\lambda\mu$-calculus.

(1) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$ and $\lambda\mu \vdash M \longrightarrow_n N$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid N : A$,
If $\Gamma \mid S \vdash_{\lambda\mu} \Delta$ and $\lambda\mu \vdash S \longrightarrow_n T$, then $\Gamma \mid T \vdash_{\lambda\mu} \Delta$.

(2) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$ and $\lambda\mu \vdash M \longrightarrow_v N$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid N : A$,
If $\Gamma \mid S \vdash_{\lambda\mu} \Delta$ and $\lambda\mu \vdash S \longrightarrow_v T$, then $\Gamma \mid T \vdash_{\lambda\mu} \Delta$.

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 $\lambda\mu$-calculi given in this paper are confluent. This is proved as a corollary of the results in Section 2.4 and 2.5 (see Proposition 2.37).

2.3 The dual calculus

The dual calculus was proposed by Wadler [48; 49] as a term calculus that corresponds to the classical sequent calculus LK. Wadler [48] first gave the dual calculus as a reduction system, and introduced it as an equation system in his later paper [49]. Detailed definitions of the later version can be found in appendix. 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 $\lambda\mu$-calculus.

Types of the dual calculus

$$A, B ::= X \mid A \wedge B \mid A \vee B \mid A \supset B \mid \neg A$$

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 $\langle M, N\rangle$, a left injection $\langle M\rangle$inl or a right injection $\langle N\rangle$inr, a complement of a coterm $[K]$not, a function abstraction $\lambda x.M$, with x bound in M, or a covariable abstraction $S.\alpha$ with α bound in S. A coterm is either a covariable α, a case $[K, L]$; a projection from the left of a product fst$[K]$ or a projection from the right of a product snd$[L]$, a complement of a term not$\langle M\rangle$, 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 $M \bullet K$.

Expressions of the dual calculus

$$\begin{aligned} M,\ N &::= x \mid \langle M, N\rangle \mid \langle M\rangle\mathrm{inl} \mid \langle M\rangle\mathrm{inr} \mid [K]\mathrm{not} \mid \lambda x.M \mid S.\alpha && (\textit{terms}) \\ K,\ L &::= \alpha \mid [K, L] \mid \mathrm{fst}[K] \mid \mathrm{snd}[L] \mid \mathrm{not}\langle M\rangle \mid M@K \mid x.S && (\textit{coterms}) \\ S,\ T &::= M \bullet K && (\textit{statements}) \end{aligned}$$

The set of free variables and covariables occurring in M, K, and S are denoted by FV(M), FV(K), and FV(S) respectively. We identify the two expressions in the α-equivalence relation and will use $\equiv$ 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, and S. The expressions $M[{}^{L}/_{\alpha}]$, $K[{}^{L}/_{\alpha}]$ and $S[{}^{L}/_{\alpha}]$ are similarly defined.

$$\frac{}{\Gamma, x : A \vdash_{dc} \Delta \mid x : A}\ \text{AxR} \qquad \frac{}{\alpha : A \mid \Gamma \vdash_{dc} \Delta, \alpha : A}\ \text{AxL}$$

$$\frac{\Gamma \vdash_{dc} \Delta \mid M : A \quad K : A \mid \Gamma \vdash_{dc} \Delta}{\Gamma \mid M \bullet K \vdash_{dc} \Delta}\ \text{Cut}$$

$$\frac{\Gamma \vdash_{dc} \Delta \mid M : A \quad \Gamma \vdash_{dc} \Delta \mid N : B}{\Gamma \vdash_{dc} \Delta \mid \langle M, N\rangle : A \wedge B}\ \wedge R$$

$$\frac{K : A \mid \Gamma \vdash_{dc} \Delta}{\text{fst}[K] : A \wedge B \mid \Gamma \vdash_{dc} \Delta}\ \wedge L \qquad \frac{L : B \mid \Gamma \vdash_{dc} \Delta}{\text{snd}[L] : A \wedge B \mid \Gamma \vdash_{dc} \Delta}\ \wedge L$$

$$\frac{\Gamma \vdash_{dc} \Delta \mid M : A}{\Gamma \vdash_{dc} \Delta \mid \langle M\rangle\text{inl} : A \vee B}\ \vee R \qquad \frac{\Gamma \vdash_{dc} \Delta \mid N : B}{\Gamma \vdash_{dc} \Delta \mid \langle N\rangle\text{inr} : A \vee B}\ \vee R$$

$$\frac{K : A \mid \Gamma \vdash_{dc} \Delta \quad L : B \mid \Gamma \vdash_{dc} \Delta}{[K, L] : A \vee B \mid \Gamma \vdash_{dc} \Delta}\ \vee L$$

$$\frac{\Gamma \vdash_{dc} \Delta \mid M : A}{\text{not}\langle M\rangle : \neg A \mid \Gamma \vdash_{dc} \Delta}\ \neg L \qquad \frac{K : A \mid \Gamma \vdash_{dc} \Delta}{\Gamma \vdash_{dc} \Delta \mid [K]\text{not} : \neg A}\ \neg R$$

$$\frac{\Gamma \vdash_{dc} \Delta \mid M : A \quad K : B \mid \Gamma \vdash_{dc} \Delta}{M@K : A \supset B \mid \Gamma \vdash_{dc} \Delta}\ \supset L \qquad \frac{\Gamma, x : A \vdash_{dc} \Delta \mid M : B}{\Gamma \vdash_{dc} \Delta \mid \lambda x.M : A \supset B}\ \supset R$$

$$\frac{x : A, \Gamma \mid S \vdash_{dc} \Delta}{x.S : A \mid \Gamma \vdash_{dc} \Delta}\ \text{LI} \qquad \frac{\Gamma \mid S \vdash_{dc} \Delta, \alpha : A}{\Gamma \vdash_{dc} \Delta \mid S.\alpha : A}\ \text{RI}$$

Figure 2.4: Typing rules of the dual calculus

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 $\alpha_1 : B_1, \ldots \alpha_m : B_m$). A *typing judgment* of the dual calculus takes either the form $\Gamma \vdash_{dc} \Delta \mid M : A$, the form $K : A \mid \Gamma \vdash_{dc} \Delta$, or the form $\Gamma \mid S \vdash_{dc} \Delta$. We note that $\vdash_{dc}$ is sometimes written as $\vdash$. The typing rules of the dual calculus are shown in figure 2.4. These rules are the same as those in Wadler's later paper [49].

A *value* of the dual calculus, denoted by V, $W \ldots$, is a variable x, a pair of values $\langle V, W\rangle$, an injection of a value $\langle V\rangle$inl or $\langle W\rangle$inr, a complement of a coterm $[K]$not, or a function $\lambda x.M$.

Values of the dual calculus

$$V, W ::= x \mid \langle V, W\rangle \mid \langle V\rangle\text{inl} \mid \langle W\rangle\text{inr} \mid [K]\text{not} \mid \lambda x.M$$

A *covalue* of the dual calculus is denoted by P, $Q \ldots$. A covalue is a covariable α, a case over a pair of covalues $[K, L]$, a projection of a covalue fst$[P]$ or snd$[Q]$, a complement of a

term $\mathrm{not}\langle M\rangle$, or a function application over a covalue $M@Q$.
Covalues of the dual calculus

$$P, Q ::= \alpha \mid [P, Q] \mid \mathrm{fst}[P] \mid \mathrm{snd}[Q] \mid \mathrm{not}\langle M\rangle \mid M@Q$$

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 $(\langle x, y\rangle \bullet \mathrm{fst}[\alpha]).\alpha$ 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 $F\{K\}$.
Term contexts and coterm contexts

$$E ::= \langle\{-\}, N\rangle \mid \langle V, \{-\}\rangle \mid \langle\{-\}\rangle\mathrm{inl} \mid \langle\{-\}\rangle\mathrm{inr}$$

$$F ::= [K, \{-\}] \mid [\{-\}, P] \mid \mathrm{fst}[\{-\}] \mid \mathrm{snd}[\{-\}] \mid M@\{-\}$$

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

$$N \bullet (M@K) \longrightarrow^{n} (N \bullet (M@\alpha)).\alpha \bullet K$$

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 2.7).

The *call-by-name reduction relation* $\longrightarrow^{n}$ and the *call-by-value reduction relation* $\longrightarrow^{v}$ of the dual calculus are defined to be the compatible closure of rules in figure 2.5. In the sequel, we use $\longrightarrow^{n*}$, $\longrightarrow^{n+}$, and $=^{n}$ as the reflexive transitive closure, the transitive closure, and the symmetric reflexive transitive closure of $\longrightarrow^{n}$ respectively. $\longrightarrow^{v*}$, $\longrightarrow^{v+}$, and $=^{v}$ are defined similarly.

Some of our reduction rules are slightly different from those in Wadler (2003), but the differences are not essential. ($\beta_{\supset}$)-rules given here are justified in Proposition 2.6. (*name*)-rules correspond to (ς)-rules of the dual calculus in Wadler (2003), though these rules are

	Call-by-name reduction	**Call-by-value reduction**
$(\beta_\wedge)$	$\langle M, N\rangle \bullet \text{fst}[P] \longrightarrow^n M \bullet P$	$\langle V, W\rangle \bullet \text{fst}[K] \longrightarrow^v V \bullet K$
	$\langle M, N\rangle \bullet \text{snd}[Q] \longrightarrow^n N \bullet Q$	$\langle V, W\rangle \bullet \text{snd}[L] \longrightarrow^v W \bullet L$
$(\beta_\vee)$	$\langle M\rangle\text{inl} \bullet [P, Q] \longrightarrow^n M \bullet P$	$\langle V\rangle\text{inl} \bullet [K, L] \longrightarrow^v V \bullet K$
	$\langle N\rangle\text{inr} \bullet [P, Q] \longrightarrow^n N \bullet Q$	$\langle W\rangle\text{inr} \bullet [K, L] \longrightarrow^v W \bullet L$
$(\beta_\neg)$	$[K]\text{not} \bullet \text{not}\langle M\rangle \longrightarrow^n M \bullet K$	$[K]\text{not} \bullet \text{not}\langle M\rangle \longrightarrow^v M \bullet K$
$(\beta_\supset)$	$\lambda x.M \bullet (N@P) \longrightarrow^n M[^N/_x] \bullet P$	$\lambda x.M \bullet (N@K) \longrightarrow^v N \bullet x.(M \bullet K)$
(β_L)	$M \bullet x.S \longrightarrow^n S[^M/_x]$	$V \bullet x.S \longrightarrow^v S[^V/_x]$
(β_R)	$S.\alpha \bullet P \longrightarrow^n S[^P/_\alpha]$	$S.\alpha \bullet K \longrightarrow^v S[^K/_\alpha]$
(η_R)	$M \longrightarrow^n (M \bullet \alpha).\alpha$	$M \longrightarrow^v (M \bullet \alpha).\alpha$
(η_L)	$K \longrightarrow^n x.(x \bullet K)$	$K \longrightarrow^v x.(x \bullet K)$
$(name)$	$M \bullet F\{K\} \longrightarrow^n (M \bullet F\{\alpha\}).\alpha \bullet K$	$E\{M\} \bullet K \longrightarrow^v M \bullet z.(E\{z\} \bullet K)$

Figure 2.5: Reduction rules of the call-by-value and call-by-name dual calculus

not included in his system. Indeed, we can easily show (*name*)-rules from (ς)-rules using (β_L) and (β_R)-rules in both the call-by-name and call-by-value systems. Conversely, we can obtain (ς)-rules from (*name*)-rules using (η_L) and (η_R)-rules.

When a term M of the dual calculus reduces a term N by the one-step call-by-name reduction, we write $DC \vdash M \longrightarrow^n N$. We also write $DC \vdash K \longrightarrow^n L$, $DC \vdash S \longrightarrow^n T$, $DC \vdash M \longrightarrow^{n*} N$, $DC \vdash K \longrightarrow^{n*} L$, $DC \vdash S \longrightarrow^{n*} T$, $DC \vdash M =^n N$, $DC \vdash K =^n L$, and $DC \vdash S =^n T$. 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 2.6
Under call-by-value, an implication can be defined by

$$\begin{array}{rcl} A \supset B & \equiv & \neg(A \wedge \neg B) \\ \lambda x.M & \equiv & [z.(z \bullet \text{fst}[x.(z \bullet \text{snd}[\text{not}\langle M\rangle])])]\text{not} \\ N@K & \equiv & \text{not}\langle\, \langle N, [K]\text{not}\rangle\, \rangle\,. \end{array}$$

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 ($\beta_\supset$)-rule is validated as follows.

(a) If N is a value V, then

$$\begin{aligned}
&(\lambda x.M) \bullet (V@K) \equiv [z.(z \bullet \mathrm{fst}[x.(z \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])])]\mathrm{not} \bullet \mathrm{not}\langle\, \langle V, [K]\mathrm{not}\rangle\, \rangle\\
&\quad \longrightarrow^{v}_{(\beta_\neg)} \langle V, [K]\mathrm{not}\rangle \bullet z.(z \bullet \mathrm{fst}[x.(z \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])])\\
&\quad \longrightarrow^{v}_{(\beta_L)} \langle V, [K]\mathrm{not}\rangle \bullet \mathrm{fst}[x.(\langle V, [K]\mathrm{not}\rangle \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])]\\
&\quad \longrightarrow^{v*}_{(\beta_\wedge)} V \bullet x.([K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle)\\
&\quad \longrightarrow^{v}_{(\beta_\neg)} V \bullet x.(M \bullet K).
\end{aligned}$$

(b) If N is not a value, we need (*name*)-rule:

$$\begin{aligned}
&(\lambda x.M) \bullet (N@K) \equiv [z.(z \bullet \mathrm{fst}[x.(z \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])])]\mathrm{not} \bullet \mathrm{not}\langle\, \langle N, [K]\mathrm{not}\rangle\, \rangle\\
&\quad \longrightarrow^{v}_{(\beta_\neg)} \langle N, [K]\mathrm{not}\rangle \bullet z.(z \bullet \mathrm{fst}[x.(z \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])])\\
&\quad \longrightarrow^{v}_{(name)} N \bullet y.\big(\langle y, [K]\mathrm{not}\rangle \bullet z.(z \bullet \mathrm{fst}[x.(z \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])])\big)\\
&\quad \longrightarrow^{v}_{(\beta_L)} N \bullet y.(\langle y, [K]\mathrm{not}\rangle \bullet \mathrm{fst}[x.(\langle y, [K]\mathrm{not}\rangle \bullet \mathrm{snd}[\mathrm{not}\langle M\rangle])])\\
&\quad \longrightarrow^{v}_{(\beta_\wedge)} N \bullet y.(y \bullet x.([K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle))\\
&\quad \longrightarrow^{v}_{(\beta_L)} N \bullet x.([K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle) \longrightarrow^{v}_{(\beta_\neg)} N \bullet x.(M \bullet K).
\end{aligned}$$

□

Proposition 2.7
Under call-by-name, an implication can be defined by

$$\begin{aligned}
A \supset B &\equiv \neg A \vee B\\
\lambda x.M &\equiv (\langle [x.(\langle M\rangle\mathrm{inr} \bullet \gamma)]\mathrm{not}\rangle\mathrm{inl} \bullet \gamma).\gamma\\
N@K &\equiv [\mathrm{not}\langle N\rangle, L]\ .
\end{aligned}$$

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 ($\beta_\supset$) and (*name*)-rules are validated as follows.

$$\begin{aligned}
(\lambda x.M) \bullet (N@P) &\equiv (\langle [x.(\langle M\rangle\mathrm{inr} \bullet \gamma)]\mathrm{not}\rangle\mathrm{inl} \bullet \gamma).\gamma \bullet [\mathrm{not}\langle N\rangle, P]\\
&\longrightarrow^{n}_{(\beta_R)} (\langle [x.(\langle M\rangle\mathrm{inr} \bullet [\mathrm{not}\langle N\rangle, P])]\mathrm{not}\rangle\mathrm{inl} \bullet [\mathrm{not}\langle N\rangle, P]\\
&\longrightarrow^{n*}_{\beta_\vee} [x.(M \bullet P)]\mathrm{not} \bullet \mathrm{not}\langle N\rangle\\
&\longrightarrow^{n}_{(\beta_\neg)} N \bullet x.(M \bullet P)\\
&\longrightarrow^{n}_{(\beta_L)} M[{}^{N}/{}_{x}] \bullet P
\end{aligned}$$

$$\begin{aligned} N \bullet (M@K) &\equiv N \bullet [\mathrm{not}\langle M\rangle, K] \\ &\longrightarrow^{n}_{(name)} (N \bullet [\mathrm{not}\langle M\rangle, \alpha]).\alpha \bullet K \\ &\equiv (N \bullet (M@\alpha)).\alpha \bullet K \end{aligned}$$

□

We now mention some basic properties of the dual calculus.

Lemma 2.8 (Substitution lemma for the dual calculus)
Let M and N be terms, K and L be coterms, and S and T be statements of the dual calculus.

(1) Suppose $\Gamma \vdash \Delta \mid M : A$, then
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 $L[^M/_x] : B \mid \Gamma \vdash \Delta$; and
if $\Gamma, x : A \mid S \vdash \Delta$, then $\Gamma \mid S[^M/_x] \vdash \Delta$.

(2) Suppose $K : A \mid \Gamma \vdash \Delta$, then
if $\Gamma \vdash \Delta, \alpha : A \mid N : B$, then $\Gamma \vdash \Delta \mid N[^K/_\alpha] : B$;
if $L : B \mid \Gamma \vdash \Delta, \alpha : A$, then $L[^K/_\alpha] : B \mid \Gamma \vdash \Delta$; and
if $\Gamma \mid S \vdash \Delta, \alpha : A$, then $\Gamma \mid S[^K/_\alpha] \vdash \Delta$.

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

Proposition 2.9 (Subject reduction for the dual calculus)
Let M and N be terms, K and L be coterms, and S and T be statements of the dual calculus.

(1) If $\Gamma \vdash \Delta \mid M : A$ and $DC \vdash M \longrightarrow^{n} N$, then $\Gamma \vdash \Delta \mid N : A$,
If $K : A \mid \Gamma \vdash \Delta$ and $DC \vdash K \longrightarrow^{n} L$, then $L : A \mid \Gamma \vdash \Delta$,
If $\Gamma \mid S \vdash \Delta$ and $DC \vdash S \longrightarrow^{n} T$, then $\Gamma \mid T \vdash \Delta$.

(2) If $\Gamma \vdash \Delta \mid M : A$ and $DC \vdash M \longrightarrow^{v} N$, then $\Gamma \vdash \Delta \mid N : A$,
If $K : A \mid \Gamma \vdash \Delta$ and $DC \vdash K \longrightarrow^{v} L$, then $L : A \mid \Gamma \vdash \Delta$,
If $\Gamma \mid S \vdash \Delta$ and $DC \vdash S \longrightarrow^{v} T$, then $\Gamma \mid T \vdash \Delta$.

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

As Wadler mentioned in his paper, the reductions of his dual calculus are confluent. Moreover, if (η_L), (η_R), and (ς)-rules are omitted, then the remaining reductions are strongly normalizing for typed terms. Our systems enjoy similar properties. However, since (η_L) and (η_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 for typed terms. For example, $\langle x, y\rangle \bullet \alpha$ is a typable statement, and this statement loops in the call-by-value calculus.

$$\begin{aligned}\langle x,y\rangle \bullet \alpha &\longrightarrow^{v}_{(\eta_R)} \langle (x \bullet \beta).\beta, y\rangle \bullet \alpha \longrightarrow^{v}_{(name)} (x \bullet \beta).\beta \bullet z.(\langle z,y\rangle \bullet \alpha)\\ &\longrightarrow^{v}_{(\beta_R)} x \bullet z.(\langle z,y\rangle \bullet \alpha) \longrightarrow^{v}_{(\beta_L)} \langle x,y\rangle \bullet \alpha\end{aligned}$$

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 $DC_n^{\eta=}$ as the call-by-name system and $DC_v^{\eta=}$ 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 (η)-rules related to implications, negations, pairs, and sums while the second one does contain them, and
- the second version contains terms of the forms $(V \bullet \mathrm{fst}[\alpha]).\alpha$ and $(V \bullet \mathrm{snd}[\beta]).\beta$ as values and coterms of the forms $x.(\langle x\rangle\mathrm{inl} \bullet P)$ and $y.(\langle y\rangle\mathrm{inr} \bullet Q)$ as covalues.

2.4 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.

2.4.1 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 2.1 (The naive translation from $\lambda\mu$ into DC)
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° and a statement S° of the dual calculus respectively.

$$\begin{array}{ll}
(x)^\circ \equiv x & (\langle M, N\rangle)^\circ \equiv \langle M^\circ, N^\circ\rangle \\
\multicolumn{2}{l}{(\delta(O, x.M, y.N))^\circ \equiv (O^\circ \bullet [x.(M^\circ \bullet \alpha), y.(N^\circ \bullet \alpha)]).\alpha} \\
\multicolumn{2}{l}{(\delta(O, x.S, y.T))^\circ \equiv O^\circ \bullet [x.S^\circ, y.T^\circ]} \\
(\mathrm{fst}(O))^\circ \equiv (O^\circ \bullet \mathrm{fst}[\alpha]).\alpha & (\mathrm{inl}(O))^\circ \equiv \langle O^\circ\rangle\mathrm{inl} \\
(\mathrm{snd}(O))^\circ \equiv (O^\circ \bullet \mathrm{snd}[\alpha]).\alpha & (\mathrm{inr}(O))^\circ \equiv \langle O^\circ\rangle\mathrm{inr} \\
(\lambda x.S)^\circ \equiv [x.S^\circ]\mathrm{not} & (OM)^\circ \equiv O^\circ \bullet \mathrm{not}\langle M^\circ\rangle \\
(\mu\alpha.S)^\circ \equiv S^\circ.\alpha & ([\alpha]M)^\circ \equiv M^\circ \bullet \alpha \\
(\lambda x.M)^\circ \equiv \lambda x.M^\circ & (OM)^\circ \equiv (O^\circ \bullet (M^\circ @ \alpha)).\alpha
\end{array}$$

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

Lemma 2.10
Let M be a term, and S be a statement of our $\lambda\mu$-calculus, then

(1) $DC^{\eta=} \vdash \langle\!\langle M\rangle\!\rangle^* =^n M^\circ$ and $DC^{\eta=} \vdash \langle\!\langle S\rangle\!\rangle^* =^n S^\circ$;

(2) $DC^{\eta=} \vdash \langle\!\langle M\rangle\!\rangle^* =^v M^\circ$ and $DC^{\eta=} \vdash \langle\!\langle S\rangle\!\rangle^* =^v S^\circ$ hold.

Proof. (1) is proved by induction on M and S. We give the sums, *i.e.*, $\mathrm{inl}(O)$, $\mathrm{inr}(O)$, $\delta(O, x.M, y.N)$, and $\delta(O, x.S, y.T)$.
Case of $\mathrm{inl}(O)$:

$$\begin{aligned}
\langle\!\langle \mathrm{inl}(O)\rangle\!\rangle^* &\equiv (\mu(\alpha,\beta).[\alpha]\langle\!\langle O\rangle\!\rangle)^* \equiv \Big(\Big\langle(\langle((\langle\!\langle O\rangle\!\rangle^* \bullet \alpha).\beta\rangle\mathrm{inr} \bullet \gamma).\alpha\Big\rangle\mathrm{inr} \bullet \gamma\Big).\gamma \\
&\overset{I.H.}{=^n} \Big(\Big\langle(\langle(O^\circ \bullet \alpha).\beta\rangle\mathrm{inr} \bullet \gamma).\alpha\Big\rangle\mathrm{inr} \bullet \gamma\Big).\gamma \\
&=^n_{(\eta_\vee)} \Big(\Big\langle(\langle(O^\circ \bullet \alpha).\beta\rangle\mathrm{inr} \bullet \gamma).\alpha\Big\rangle\mathrm{inr} \bullet [x.(\langle x\rangle\mathrm{inl} \bullet \gamma), y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]\Big).\gamma \\
&=^n_{(\beta_\vee)} \Big(\langle((O^\circ \bullet \alpha).\beta\rangle\mathrm{inr} \bullet \gamma).\alpha \bullet y.(\langle y\rangle\mathrm{inr} \bullet \gamma)\Big).\gamma \\
&=^n_{(\eta_\vee)} \Big(\langle((O^\circ \bullet \alpha).\beta\rangle\mathrm{inr} \bullet [x.(\langle x\rangle\mathrm{inl} \bullet \gamma), y.(\langle y\rangle\mathrm{inr} \bullet \gamma)]).\alpha \bullet y.(\langle y\rangle\mathrm{inr} \bullet \gamma)\Big).\gamma \\
&=^n_{(\beta_\vee)} \Big((O^\circ \bullet \alpha).\beta \bullet x.(\langle x\rangle\mathrm{inl} \bullet \gamma)).\alpha \bullet y.(\langle y\rangle\mathrm{inr} \bullet \gamma)\Big).\gamma \\
&=^n_{(\beta_R)} \Big(O^\circ \bullet y.(\langle y\rangle\mathrm{inr} \bullet \gamma)\Big).\gamma \\
&=^n_{(\beta_L)} (\langle O^\circ\rangle\mathrm{inr} \bullet \gamma).\gamma
\end{aligned}$$

$$=^{n}_{(\eta_L)} \langle O^\circ \rangle \mathrm{inr}$$

Note that these equations are the $DC^{\eta=}$ equation, that is, Wadler's system (2005).
Case of $\mathrm{inr}(O)$: this case is proved in a way similar to the above case.
Case of $\delta(O, x.M, y.N)$:

$$\begin{aligned}
\langle\!\langle \delta(O, x.M, y.N) \rangle\!\rangle^* &\equiv \Big(\mu\gamma.\big(\lambda y.[\gamma]\langle\!\langle N \rangle\!\rangle\big)\big(\mu\beta.(\lambda x.[\gamma]\langle\!\langle M \rangle\!\rangle)\mu\alpha.[\alpha,\beta]\langle\!\langle O \rangle\!\rangle\big)\Big)^* \\
&\equiv \Big(\big[y.(\langle\!\langle N \rangle\!\rangle^* \bullet \gamma)\big]\mathrm{not} \bullet \mathrm{not}\Big\langle\Big([x.(\langle\!\langle M \rangle\!\rangle^* \bullet \gamma)]\mathrm{not} \bullet \mathrm{not}\big\langle(\langle\!\langle O \rangle\!\rangle^* \bullet [\alpha,\beta]).\alpha\big\rangle\Big).\beta\Big\rangle\Big).\gamma \\
&\overset{I.H.}{=^{n}} \Big(\big[y.(N^\circ \bullet \gamma)\big]\mathrm{not} \bullet \mathrm{not}\Big\langle\Big([x.(M^\circ \bullet \gamma)]\mathrm{not} \bullet \mathrm{not}\big\langle(O^\circ \bullet [\alpha,\beta]).\alpha\big\rangle\Big).\beta\Big\rangle\Big).\gamma \\
&=^{n}_{\beta_\neg} \Big(\big[y.(N^\circ \bullet \gamma)\big]\mathrm{not} \bullet \mathrm{not}\Big\langle\Big((O^\circ \bullet [\alpha,\beta]).\alpha \bullet x.(M^\circ \bullet \gamma)\Big).\beta\Big\rangle\Big).\gamma \\
&=^{n}_{\beta_\neg} \Big(\Big((O^\circ \bullet [\alpha,\beta]).\alpha \bullet x.(M^\circ \bullet \gamma)\Big).\beta \bullet y.(N^\circ \bullet \gamma)\Big).\gamma \\
&=^{n}_{(name)} \Big(\Big(O^\circ \bullet \big[x.(M^\circ \bullet \gamma), \beta\big]\Big).\beta \bullet y.(N^\circ \bullet \gamma)\Big).\gamma \\
&=^{n}_{(name)} \Big(O^\circ \bullet \big[x.(M^\circ \bullet \gamma), y.(N^\circ \bullet \gamma)\big]\Big).\gamma \\
&\equiv (\delta(O, x.M, y.N))^\circ
\end{aligned}$$

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

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

The naive translation preserves typing rules and equalities.

Proposition 2.11

(1) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$, then $\Gamma \vdash \Delta \mid M^\circ : A$.
If $\Gamma \mid S \vdash_{\lambda\mu} \Delta$, then $\Gamma \mid S^\circ \vdash \Delta$.

(2) If $\lambda\mu \vdash M =_n N$, then $DC \vdash M^\circ =^n N^\circ$.
If $\lambda\mu \vdash S =_n T$, then $DC \vdash S^\circ =^n T^\circ$.

(3) If $\lambda\mu \vdash M =_v N$, then $DC \vdash M^\circ =^v N^\circ$.
If $\lambda\mu \vdash S =_v T$, then $DC \vdash S^\circ =^v T^\circ$.

Proof. (1) We can prove this claim by a straight forward induction on $\vdash_{\lambda\mu}$.
(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 2.16 using Lemma 2.12.
(3) As with (2), we can show this claim directly, or as a corollary of Theorem 2.21 using

Lemma 2.17. $\square$

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

$$\begin{aligned}
\Big((\mu\alpha.[\beta]\lambda x.[\alpha]x)y\Big)^\circ &\equiv \Big(([x.(x \bullet \alpha)]\mathrm{not} \bullet \beta).\alpha \bullet (y@\gamma)\Big).\gamma \\
&\longrightarrow^n_{(\beta_R)} \Big([x.(x \bullet (y@\gamma))]\mathrm{not} \bullet \beta\Big).\gamma \\
&\longleftarrow^n_{(\beta_R)} \Big([x.((x \bullet (y@\gamma')).\gamma' \bullet \gamma)]\mathrm{not} \bullet \beta\Big).\gamma \\
&\equiv \Big(\mu\gamma.[\beta]\lambda x.[\gamma](xy)\Big)^\circ
\end{aligned}$$

To solve this problem, we modify the naive translation. The idea of modification is similar to the modified CPS translation by de Groote [13; 14]. 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.

2.4.2 The translation from CBN $\lambda\mu$-calculus into CBN dual calculus

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

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

Definition 2.2 ($(-)^\sharp$: CBN $\lambda\mu$-calculus $\longrightarrow$ CBN dual calculus)
Let M be a term of the $\lambda\mu$-calculus, $(M)^\sharp$ is defined as

$$(M)^\sharp \equiv (M :_n \alpha).\alpha \qquad \text{(where } \alpha \text{ is a fresh covariable)}$$

and let S be a statement of the $\lambda\mu$-calculus, $(S)^\sharp$ is defined as

$$\begin{aligned}
([\alpha]M)^\sharp &\equiv M :_n \alpha \\
(MN)^\sharp &\equiv M :_n \mathrm{not}\langle N^\sharp\rangle && \text{(where } M \text{ is not a } \lambda\text{-abstraction)} \\
((\lambda x.S)N)^\sharp &\equiv N^\sharp \bullet x.S^\sharp \\
(\delta(O, x.S, y.T))^\sharp &\equiv O^\sharp \bullet [x.S^\sharp, y.T^\sharp]
\end{aligned}$$

where the infix operator "$:_n$" translates a pair of a term M of the $\lambda\mu$-calculus and a coterm K of the dual calculus into a statement $M :_n K$ of the dual calculus. This operator is defined as follows:

$$
\begin{array}{rlrl}
x :_n K \equiv x \bullet K & & \langle M, N\rangle :_n K \equiv \langle M^\sharp, N^\sharp\rangle \bullet K \\
\mathrm{fst}(M) :_n K \equiv M :_n \mathrm{fst}[K] & & \mathrm{inl}(M) :_n K \equiv \langle M^\sharp\rangle\mathrm{inl} \bullet K \\
\mathrm{snd}(M) :_n K \equiv M :_n \mathrm{snd}[K] & & \mathrm{inr}(M) :_n K \equiv \langle M^\sharp\rangle\mathrm{inr} \bullet K \\
(\lambda x.S) :_n K \equiv [x.S^\sharp]\mathrm{not} \bullet K & & \mu\alpha.S :_n K \equiv S^\sharp.\overline{\alpha} \bullet K \\
(\lambda x.M) :_n K \equiv (\lambda x.M^\sharp) \bullet K & & \\
(MN) :_n K \equiv M :_n (N^\sharp @ K) & & \text{(where } M \text{ is not a } \lambda\text{-abstraction)} \\
((\lambda x.M)N) :_n K \equiv (N^\sharp \bullet x.(M :_n \alpha)).\overline{\alpha} \bullet K & & \\
\delta(O, x.M, y.N) :_n K \equiv \big(O^\sharp \bullet [x.(M :_n \alpha), y.(N :_n \alpha)]\big).\overline{\alpha} \bullet K & &
\end{array}
$$

where $S.\overline{\alpha} \bullet K$ 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 2.12
Let M and S be a term and a statement of the $\lambda\mu$-calculus, then

(1) $DC \vdash M^\circ \bullet P \longrightarrow^{n*} M :_n P$ for any covalue P,

(2) $DC \vdash M^\circ \longrightarrow^{n*} M^\sharp$, and

(3) $DC \vdash S^\circ \longrightarrow^{n*} S^\sharp$.

Proof. 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

$$M^\circ \longrightarrow^n_{\eta_R} (M^\circ \bullet \alpha).\alpha \overset{(1)}{\longrightarrow}{}^{n*} (M :_n \alpha).\alpha \equiv M^\sharp.$$

Therefore we prove (1) and (3).
Case of x : this case is immediate.
Case of $\lambda x.M$, $\langle M, N\rangle$, $\mathrm{inl}(O)$, and $\mathrm{inr}(O)$: these cases are easily shown by the induction hypothesis of (2).
Case of MN (MN is a term, M is not a λ-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 λ-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).
Case of $(\lambda x.M)N$:

$$\begin{aligned}((\lambda x.M)N)^{\circ} \bullet P &\equiv ((\lambda x.M^{\circ}) \bullet (N^{\circ}@\alpha)).\alpha \bullet P \longrightarrow^{n}_{(\beta_L)} (\lambda x.M^{\circ}) \bullet (N^{\circ}@P) \\ &\longrightarrow^{n}_{(\beta_{\supset})} N^{\circ} \bullet x.(M^{\circ} \bullet P) \overset{I.H.(1)}{\longrightarrow^{n*}} N^{\sharp} \bullet x.(M^{\circ} \bullet P) \\ &\overset{I.H.(2)}{\longrightarrow^{n*}} N^{\sharp} \bullet x.(M :_n P) \equiv ((\lambda x.M)N) :_n P\end{aligned}$$

Case of fst(O) :

$$\mathrm{fst}(O)^{\circ} \bullet P \equiv (O^{\circ} \bullet \mathrm{fst}[\alpha]).\alpha \bullet P \longrightarrow^{n}_{(\beta_L)} O^{\circ} \bullet \mathrm{fst}[P] \overset{I.H.(1)}{\longrightarrow^{n*}} O :_n \mathrm{fst}[P] \equiv \mathrm{fst}(O) :_n P$$

Case of snd(O) : this case is shown in a way similar to the above case.
Case of $\delta(O, x.M, y.N)$:

$$\begin{aligned}&\delta(O, x.M, y.N)^{\circ} \bullet P \equiv (O^{\circ} \bullet [x.(M^{\circ} \bullet \alpha), y.(N^{\circ} \bullet \alpha)]).\alpha \bullet P \\ &\longrightarrow^{n}_{(\beta_L)} O^{\circ} \bullet [x.(M^{\circ} \bullet P), y.(N^{\circ} \bullet P)] \overset{I.H.(1)}{\longrightarrow^{n*}} O^{\circ} \bullet [x.(M :_n P), y.(N :_n P)] \\ &\overset{I.H.(2)}{\longrightarrow^{n*}} O^{\sharp} \bullet [x.(M :_n P), y.(N :_n P)] \equiv \delta(O, x.M, y.N) :_n P\end{aligned}$$

Case of $\mu\alpha.S$:

$$(\mu\alpha.S)^{\circ} \bullet P \equiv S^{\circ}.\alpha \bullet P \overset{I.H.(3)}{\longrightarrow^{n*}} S^{\sharp}.\alpha \bullet P \longrightarrow^{n}_{(\beta_L)} S^{\sharp}[{}^{P}/{}_{\alpha}] \equiv \mu\alpha.S :_n P$$

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

$$\begin{aligned}((\lambda x.S)N)^{\circ} &\equiv [x.S^{\circ}]\mathrm{not} \bullet \mathrm{not}\langle N^{\circ}\rangle \longrightarrow^{n}_{(\beta_{\neg})} N^{\circ} \bullet x.S^{\circ} \\ &\overset{I.H.(2),(3)}{\longrightarrow^{n*}} N^{\sharp} \bullet x.S^{\sharp} \equiv ((\lambda x.S)N)^{\sharp}\end{aligned}$$

□

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

Proposition 2.13

(1) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$, then $\Gamma \vdash_{dc} \Delta \mid M^{\sharp} : A$.

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

Proof. This proposition can be shown by the subject reduction property for the dual calculus using Proposition 2.11 and Lemma 2.12. □

Lemma 2.14

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

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

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

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

Proof. (1) we can prove this claim by a straightforward induction on M and S. The key case is :

$$(x :_n P)[^{N^\sharp}/_x] \equiv (x \bullet P)[^{N^\sharp}/_x] \equiv N^\sharp \bullet P[^{N^\sharp}/_x] \equiv (N :_n \alpha).\alpha \bullet P[^{N^\sharp}/_x] \longrightarrow^n N :_n P[^{N^\sharp}/_x]$$

(2) $M \bullet [x.(x \bullet P), y.(y \bullet Q)] \longrightarrow^n_{(name)} (M \bullet [x.(x \bullet P), \beta]).\beta \bullet y.(y \bullet Q)$

$$\longrightarrow^n_{(\beta_L)} (M \bullet [x.(x \bullet P), \beta]).\beta \bullet Q \longrightarrow^n_{(\beta_R)} M \bullet [x.(x \bullet P), Q]$$
$$\longrightarrow^n_{(name)} (M \bullet [\alpha, Q]).\alpha \bullet x.(x \bullet P) \longrightarrow^n_{(\beta_L)} (M \bullet [\alpha, Q]).\alpha \bullet P \longrightarrow^n_{(\beta_R)} M \bullet [P, Q]$$

(3) this claim can be shown by induction on term O.
Cases of x, $\lambda x.M$, $\lambda x.S$, $\mu\alpha.S$, $\langle M, N\rangle$, $\mathrm{inl}(M)$, and $\mathrm{inr}(M)$: these are easily shown by the definition.
Case of MN (MN is a term, M is not a λ-abstraction) :

$$MN :_n K \equiv M :_n N^\sharp @ K \overset{I.H.}{\longrightarrow^{n*}} M^\sharp \bullet (N^\sharp @ K) \longrightarrow^n_{(name)} (M^\sharp \bullet (N^\sharp @ \alpha)).\alpha \bullet K$$
$$\longrightarrow^n (M :_n N^\sharp @ \alpha).\alpha \bullet K \equiv (MN :_n \alpha).\alpha \bullet K \equiv (MN)^\sharp \bullet K$$

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

$$(\lambda x.M)N :_n K \equiv (N^\sharp \bullet x.(M :_n \alpha)).\alpha \bullet K \equiv ((\lambda x.M)N :_n \alpha).\alpha \bullet K \equiv ((\lambda x.M)N)^\sharp \bullet K$$

Case of fst(O) :

$$\begin{aligned}\mathrm{fst}(O) :_n K \equiv O :_n \mathrm{fst}[K] \overset{I.H.}{\longrightarrow^{n*}} O^\sharp \bullet \mathrm{fst}[K] \longrightarrow^n_{(name)} (O^\sharp \bullet \mathrm{fst}[\alpha]).\alpha \bullet K \\ \longrightarrow^n (M :_n \mathrm{fst}[\alpha]).\alpha \bullet K \equiv (\mathrm{fst}(O) :_n \alpha).\alpha \bullet K \equiv \mathrm{fst}(O)^\sharp \bullet K\end{aligned}$$

Case of snd(O) : this case is shown in a way similar to the above case.
Case of $\delta(O, x.M, y.N)$:

$$\begin{aligned}\delta(O, x.M, y.N) :_n K \equiv (O^\sharp, [x.(M :_n \alpha), y.(N :_n \alpha)]).\alpha \bullet K \\ \equiv (\delta(O, x.M, y.N) :_n \alpha).\alpha \bullet K \equiv \delta(O, x.M, y.N)^\sharp \bullet K\end{aligned}$$

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

$$\begin{aligned}(\lambda x.M) : (N^\sharp @P) \equiv (\lambda x.M^\sharp) \bullet (N^\sharp @P) \longrightarrow^n_{\beta_\supset} N^\sharp \bullet x.(M^\sharp \bullet P) \\ \longrightarrow^n_{\beta_L} N^\sharp \bullet x.(M :_n P) \equiv (\lambda x.M)N :_n P \\ (\lambda x.S) : \mathrm{not}\langle N^\sharp\rangle \equiv [x.S^\sharp]\mathrm{not} \bullet \mathrm{not}\langle N^\sharp\rangle \longrightarrow^n_{\beta_\neg} N^\sharp \bullet x.S^\sharp \equiv ((\lambda x.S)N)^\sharp\end{aligned}$$

□

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:

$$\begin{array}{ll}\Phi(\{-\}, P) \equiv P & \Phi(E_n N, P) \equiv \Phi(E_n, N^\sharp @P) \\ \Phi(\mathrm{fst}(E_n), P) \equiv \Phi(E_n, \mathrm{fst}[P]) & \Phi(\mathrm{snd}(E_n), P) \equiv \Phi(E_n, \mathrm{snd}[P]) \\ \multicolumn{2}{l}{\Phi(\delta(E_n, x.E'_n\{x\}, y.E''_n\{y\}), P) \equiv \Phi(E_n, [\Phi(E'_n, P), \Phi(E''_n, P)])} \\ \Phi([\alpha]E_n) \equiv \Phi(E_n, \alpha) & \Phi(E_n N) \equiv \Phi(E_n, \mathrm{not}\langle N^\sharp\rangle) \\ \multicolumn{2}{l}{\Phi(\delta(E_n, x.D_n\{x\}, y.D'_n\{y\}), P) \equiv \Phi(E_n, [\Phi(D_n), \Phi(D'_n)])}\end{array}$$

Then the following properties hold.

Lemma 2.15

(1) If M is not a λ-abstraction, then $DC \vdash (E_n\{M\} :_n P) \longrightarrow^{n*} (M :_n \Phi(E_n, P))$ and $DC \vdash (D_n\{M\})^\sharp \longrightarrow^{n*} (M :_n \Phi(D_n))$ hold.

(2) For any term M of the $\lambda\mu$-calculus, $DC \vdash (M :_n \Phi(E_n, P)) \longrightarrow^{n*} (E_n\{M\} :_n P)$ and $DC \vdash (M :_n \Phi(D_n)) \longrightarrow^{n*} (D_n\{M\})^\sharp$ hold.

(3) Let M be a term and S be a statement of the $\lambda\mu$-calculus, then
$DC \vdash (M :_n P)[^{\Phi(D_n)}/_{\alpha}] \longrightarrow^{n*} M[^{D_n\{-\}}/_{[\alpha]\{-\}}] :_n P[^{\Phi(D_n)}/_{\alpha}]$, and
$DC \vdash S^{\sharp}[^{\Phi(D_n)}/_{\alpha}] \longrightarrow^{n*} (S[^{D_n\{-\}}/_{[\alpha]\{-\}}])^{\sharp}$ hold for any covalue P.

Proof. (1) is proved by induction on E_n and D_n. For E_nN, we can prove the claim by the induction hypothesis, since $E_n\{M\}$ is not a λ-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 $\delta(E_n, x.E'_n\{x\}, y.E''_n\{y\})$:

$$\begin{aligned}
&\delta(E_n\{M\}, x.E'_n\{x\}, y.E''_n\{y\}) :_n P \equiv E_n\{M\}^{\sharp} \bullet [x.(E'_n\{x\} :_n P), y.(E''_n\{y\} :_n P)] \\
&\overset{I.H.}{\longrightarrow^{n*}} E_n\{M\}^{\sharp} \bullet [x.(x \bullet \Phi(E'_n, P)), y.(y \bullet \Phi(E''_n, P))] \\
&\overset{Lem\ 2.14(2)}{\longrightarrow^{n*}} E_n\{M\}^{\sharp} \bullet [\Phi(E'_n, P), \Phi(E''_n, P)] \\
&\equiv (E_n\{M\} :_n \alpha).\alpha \bullet [\Phi(E'_n, P), \Phi(E''_n, P)] \\
&\longrightarrow^{n} (E_n\{M\} :_n [\Phi(E'_n, P), \Phi(E''_n, P)] \\
&\overset{I.H.}{\longrightarrow^{n*}} M :_n \Phi(E_n, [\Phi(E'_n, P), \Phi(E''_n, P)]) \\
&\equiv M :_n \Phi(\delta(E_n, x.E'_n\{x\}, y.E''_n\{y\}), P)
\end{aligned}$$

Case of $\delta(E_n, x.D_n\{x\}, y.D'_n\{y\})$: 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 induction on E_n and D_n. If E_n is E_nN, then

$$\begin{aligned}
M :_n \Phi(E_nN, P) \equiv M :_n \Phi(E_n, N^{\sharp}@P) &\overset{I.H.}{\longrightarrow^{n*}} E_n\{M\} :_n (N^{\sharp}@P) \\
&\overset{Lem\ 2.14(4)}{\longrightarrow^{n*}} E_n\{M\}N :_n P
\end{aligned}$$

If D_n is E_nN, this case is also proved by the induction hypothesis and Lemma 2.14(4). If E_n is $\delta(E_n, x.E'_n\{x\}, y.E''_n\{y\})$, then

$$\begin{aligned}
M :_n &\Phi(\delta(E_n, x.E'_n\{x\}, y.E''_n\{y\}), P) \equiv M :_n \Phi(E_n, [\Phi(E'_n, P), \Phi(E''_n, P)]) \\
&\overset{I.H.}{\longrightarrow^{n*}} E_n\{M\} :_n [\Phi(E'_n, P), \Phi(E''_n, P)] \\
&\longrightarrow^{n*}_{\eta_L} E_n\{M\} :_n [x.(x \bullet \Phi(E'_n, P)), y.(y \bullet \Phi(E''_n, P))] \\
&\overset{I.H.}{\longrightarrow^{n*}} E_n\{M\} :_n [x.(E'_n\{x\} :_n P), y.(E''_n\{y\} :_n P)] \\
&\overset{Lem\ 2.14(3)}{\longrightarrow^{n*}} (E_n\{M\})^{\sharp} \bullet [x.(E'_n\{x\} :_n P), y.(E''_n\{y\} :_n P)] \\
&\equiv \delta(E_n\{M\}, x.E'_n\{x\}, y.E''_n\{y\}) :_n P
\end{aligned}$$

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. The remaining cases are proved easily using the the induction hypothesis.
(3) is proved by induction on M and S. The key case is $[\alpha]M$.

$$
\begin{aligned}
([\alpha]M)^\sharp[{}^{\Phi(D_n)}/_\alpha] \equiv (M :_n \alpha)[{}^{\Phi(D_n)}/_\alpha] &\overset{I.H.}{\longrightarrow^{n*}} (M[{}^{D_n\{-\}}/_{[\alpha]\{-\}}] :_n \Phi(D_n)) \\
&\overset{(2)}{\longrightarrow^{n*}} \left(D_n\{M[{}^{D_n\{-\}}/_{[\alpha]\{-\}}]\}\right)^\sharp \equiv \left(([\alpha]M)[{}^{D_n\{-\}}/_{[\alpha]\{-\}}]\right)^\sharp
\end{aligned}
$$

The other cases are proved easily using the the induction hypothesis. □

Then, we prove that the call-by-name modified translation $(-)^\sharp$ preserves reductions.

Theorem 2.16 (Soundness of $(-)^\sharp$)
(1) If $\lambda\mu \vdash M \longrightarrow_n N$, then $DC \vdash (M :_n P) \longrightarrow^{n*} (N :_n P)$ for any covalue P, especially $DC \vdash M^\sharp \longrightarrow^{n*} N^\sharp$.
(2) If $\lambda\mu \vdash S \longrightarrow_n T$, then $DC \vdash S^\sharp \longrightarrow^{n*} T^\sharp$.
Moreover, if $\longrightarrow_n$ is $(\beta_\supset)$, $(\beta_\wedge)$, $(\beta_\vee)$ or $(\beta_\neg)$, then $\longrightarrow^{n*}$ can be replaced by $\longrightarrow^{n+}$.

Proof. The claims are proved by simultaneous induction on $\longrightarrow_n$.
- Base cases are shown as follows.
Case of $(\beta_\supset)$:

$$((\lambda x.M)N :_n P) \equiv N^\sharp \bullet x.(M :_n P) \longrightarrow^n_{(\beta_L)} (M :_n P)[{}^{N^\sharp}/_x] \overset{Lem\ 2.14(1)}{\longrightarrow^{n*}} (M[{}^N/_x] :_n P)$$

Case of $(\beta_\wedge)$:

$$
\begin{aligned}
(\mathrm{fst}\langle M, N\rangle :_n P) &\equiv \langle M, N\rangle :_n \mathrm{fst}[P] \equiv \langle M^\sharp, N^\sharp\rangle \bullet \mathrm{fst}[P] \\
&\longrightarrow^n_{(\beta_\vee)} M^\sharp \bullet P \longrightarrow^n_{(\beta_R)} (M :_n P)
\end{aligned}
$$

The other rule of $(\beta_\wedge)$ can be proved similarly.
Case of $(\beta_\vee)$:

$$
\begin{aligned}
&(\delta(\mathrm{inl}(O), x.E_n\{x\}, y.E'_n\{y\}) :_n P) \overset{Lem\ 2.15(1)}{\longrightarrow^{n*}} (\mathrm{inl}(O) :_n \Phi(\delta(\{-\}, x.E_n\{x\}, y.E'_n\{y\}), P)) \\
&\equiv (\mathrm{inl}(O) :_n \Phi(\{-\}, [\Phi(E_n, P), \Phi(E'_n, P)])) \equiv (\mathrm{inl}(O) :_n [\Phi(E_n, P), \Phi(E'_n, P)]) \\
&\equiv (\langle O^\sharp\rangle\mathrm{inl} \bullet [\Phi(E_n, P), \Phi(E'_n, P)]) \longrightarrow^n_{(\beta_\vee)} O^\sharp \bullet \Phi(E_n, P) \longrightarrow^n_{(\beta_R)} O :_n \Phi(E_n, P) \\
&\overset{Lem\ 2.15(2)}{\longrightarrow^{n*}} (E_n\{O\} :_n P)
\end{aligned}
$$

The other rules of $(\beta_\vee)$ can be proved similarly.

Case of $(\beta_\neg)$:

$$((\lambda x.S)N)^\sharp \equiv N^\sharp \bullet x.S^\sharp \longrightarrow^n_{(\beta_L)} S^\sharp[{}^{N^\sharp}/{}_x] \overset{Lem\ 2.14(1)}{\longrightarrow^{n*}} (S[{}^N/{}_x])^\sharp$$

Case of (ζ) :

$$\begin{aligned}(E_n\{\mu\alpha.S\} :_n P) &\overset{Lem\ 2.15(1)}{\longrightarrow^{n*}} (\mu\alpha.S :_n \Phi(E_n,P)) \equiv S^\sharp[{}^{\Phi(E_n,P)}/{}_\alpha] \equiv S^\sharp[{}^{\Phi(E_n,\beta)}/{}_\alpha][{}^P/{}_\beta]\\ &\equiv S^\sharp[{}^{\Phi([\beta]E_n)}/{}_\alpha][{}^P/{}_\beta] \overset{Lem\ 2.15(3)}{\longrightarrow^{n*}} (S[{}^{[\beta]E_n\{-\}}/{}_{[\alpha]\{-\}}])^\sharp[{}^P/{}_\beta]\\ &\equiv (\mu\beta.S[{}^{[\beta]E_n\{-\}}/{}_{[\alpha]\{-\}}] :_n P)\end{aligned}$$

The other rule of (ζ) can be proved similarly.

Case of (η_μ) :

$$(\mu\alpha.[\alpha]M :_n P) \equiv ([\alpha]M)^\sharp[{}^P/{}_\alpha] \equiv (M :_n \alpha)[{}^P/{}_\alpha] \equiv M :_n P$$

Case of (π) :

$$\begin{aligned}(E_n\{\delta(O,x.M,y.N)\} :_n P) &\overset{Lem\ 2.15(1)}{\longrightarrow^{n*}} (\delta(O,x.M,y.N) :_n \Phi(E_n,P))\\ &\equiv (O^\sharp \bullet [x.(M :_n \Phi(E_n,P)), y.(N :_n \Phi(E_n,P))])\\ &\overset{Lem\ 2.15(2)}{\longrightarrow^{n*}} (O^\sharp \bullet [x.(E_n\{M\} :_n P), y.(E_n\{N\} :_n P)])\\ &\equiv (\delta(O,x.E_n\{M\},y.E_n\{N\}) :_n P)\end{aligned}$$

The other rules of (π) can be proved similarly.

Case of (ν) : Let T not be a simple form. Then

$$\begin{aligned}&(\delta(O,x.S,y.T))^\sharp \equiv O^\sharp \bullet [x.S^\sharp, y.T^\sharp] \longrightarrow^n_{(name)} (O^\sharp \bullet [x.S^\sharp, \beta]).\beta \bullet y.T^\sharp\\ &\longrightarrow^n_{(\eta_L)} (O^\sharp \bullet [x.S^\sharp, y'.(y' \bullet \beta)]).\beta \bullet y.T^\sharp \equiv \delta(O,x.S,y'.[\beta]y')^\sharp.\beta \bullet y.T^\sharp\\ &\equiv ((\lambda y.T)\mu\beta.\delta(O,x.S,y'.[\beta]y'))^\sharp.\end{aligned}$$

The other rule of (ν) can be proved similarly.

- Induction cases of (1) and (2).

We can easily show these cases by the induction hypothesis. We consider the less than obvious case: $ON \longrightarrow_n (\lambda x.M)N$ is obtained from $O \longrightarrow_n \lambda x.M$ and O is not a λ-abstraction.

$$\begin{aligned}(ON :_n P) &\equiv (O :_n (N^\sharp @ P)) \overset{I.H.}{\longrightarrow^{n*}} (\lambda x.M :_n (N^\sharp @ P)) \equiv \lambda x.M^\sharp \bullet (N^\sharp @ P)\\ &\longrightarrow^n_{(\beta_\supset)} N^\sharp \bullet x.(M^\sharp \bullet P) \longrightarrow^n_{(\beta_R)} N^\sharp \bullet x.(M :_n P) \equiv ((\lambda x.M)N :_n P)\end{aligned}$$

□

2.4.3 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 $(-)^\circ$. The call-by-name translation also consists of the two translations: $(-)^\dagger$ and $(-) :_v K$.

Definition 2.3 ($(-)^\dagger$: CBV $\lambda\mu$-calculus $\longrightarrow$ CBV dual calculus)
Let M be a term of the $\lambda\mu$-calculus, $(M)^\dagger$ is defined as

$$(M)^\dagger \equiv (M :_v \alpha).\alpha \qquad\qquad \text{(where } \alpha \text{ is a fresh covariable)}$$

and let S be a statement of the $\lambda\mu$-calculus, $(S)^\dagger$ is defined as

$$\begin{aligned}
([\alpha]M)^\dagger &\equiv M :_v \alpha \\
(MN)^\dagger &\equiv M :_v \mathrm{not}\langle N^\dagger\rangle \qquad\qquad \text{(where } M \text{ is not a } \lambda\text{-abstraction)} \\
((\lambda x.S)N)^\dagger &\equiv N :_v x.S^\dagger \\
(\delta(O, x.S, y.T))^\dagger &\equiv M :_v [x.S^\dagger, y.T^\dagger]
\end{aligned}$$

where the infix operator "$:_v$" translates a pair of terms M of the $\lambda\mu$-calculus and a coterm K of the dual calculus into a statement $M :_v K$ of the dual calculus. This operator is defined as follows:

$$\begin{aligned}
x :_v K &\equiv x \bullet K & \langle M, N\rangle :_v K &\equiv \langle M^\dagger, N^\dagger\rangle \bullet K \\
\mathrm{fst}(M) :_v K &\equiv M :_v \mathrm{fst}[K] & \mathrm{inl}(M) :_v K &\equiv \langle M^\dagger\rangle\mathrm{inl} \bullet K \\
\mathrm{snd}(M) :_v K &\equiv M :_v \mathrm{snd}[K] & \mathrm{inr}(M) :_v K &\equiv \langle M^\dagger\rangle\mathrm{inr} \bullet K \\
(\lambda x.S) :_v K &\equiv [x.S^\dagger]\mathrm{not} \bullet K & \mu\alpha.S :_v K &\equiv S^\dagger[{}^K/_\alpha] \\
(\lambda x.M) :_v K &\equiv (\lambda x.M^\dagger) \bullet K \\
(MN) :_v K &\equiv M :_v (N^\dagger @ K) & &\text{(where } M \text{ is not a } \lambda\text{-abstraction)} \\
((\lambda x.M)N) :_v K &\equiv N :_v x.(M :_v K) \\
\delta(O, x.M, y.N) :_v K &\equiv O :_v [x.(M :_v K), y.(N :_v K)]
\end{aligned}$$

When we compare the CBV translation given here with the CBN translation, the definitions of $(\mu\alpha.S :_v K)$, $((\lambda x.M)N :_v K)$, $(\delta(O, x.M, y.N) :_v K)$, $((\lambda x.S)N)^\dagger$, and $\delta(O, x.S, y.T)^\dagger$ are different. This is because the administrative redexes differ according to the difference of (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 2.17
Let M and S be a term and a statement of the $\lambda\mu$-calculus, then

(1) $DC \vdash M^\circ \bullet K \longrightarrow^{v*} M :_v K$ for any coterm K,

(2) $DC \vdash M^\circ \longrightarrow^{v*} M^\dagger$, and

(3) $DC \vdash S^\circ \longrightarrow^{v*} S^\dagger$.

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

$$((\lambda x.M)N)^\circ \bullet K \equiv ((\lambda x.M^\circ) \bullet (N^\circ @\alpha)).\alpha \bullet K \longrightarrow^v_{(\beta_L)} (\lambda x.M^\circ) \bullet (N^\circ @K)$$
$$\longrightarrow^v_{(\beta_\supset)} N^\circ \bullet x.(M^\circ \bullet K) \overset{I.H.(1)}{\longrightarrow^{v*}} N :_v x.(M :_v K) \equiv ((\lambda x.M)N) :_v K$$

□

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

Proposition 2.18

(1) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$, then $\Gamma \vdash_{dc} \Delta \mid M^\dagger : A$.

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

Proof. This proposition can be shown by the subject reduction property for the dual calculus using Proposition 2.11 and Lemma 2.17. □

Definition 2.4

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

$$\begin{array}{rlrl} (x)^v & \equiv x, & (\langle V, W\rangle)^v & \equiv \langle V^v, W^v\rangle \\ (\mathrm{inl}(V))^v & \equiv \langle V^v\rangle\mathrm{inl} & (\lambda x.M)^v & \equiv \lambda x.M^\dagger \\ (\mathrm{inr}(W))^v & \equiv \langle W^v\rangle\mathrm{inr} & (\lambda x.S)^v & \equiv [x.S^\dagger]\mathrm{not} \end{array}$$

Using this notation, we can show the following lemma.

Lemma 2.19

(1) $DC \vdash V^v \bullet K \longrightarrow^{v*} (V :_v K)$ for any coterm K, especially $V^v \longrightarrow^{v*} V^\dagger$.

(2) $DC \vdash (V :_v K) \longrightarrow^{v*} V^v \bullet K$ for any coterm K. That is, the statement $(V :_v K)$ 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
$DC \vdash (M :_v K)[^{V^v}/_x] \longrightarrow^{v*} (M[^V/_x] :_v K[^{V^v}/_x])$, and
$DC \vdash S^\dagger[^{V^v}/_x] \longrightarrow^{v*} (S[^V/_x])^\dagger$ for any coterm K.

Proof. (1) is proved by a straightforward induction on V. For example, we consider the case of $\langle V, W\rangle$: $(\langle V, W\rangle^v \bullet K) \equiv \langle V^v, W^v\rangle \bullet K \overset{I.H.}{\longrightarrow^{v*}} \langle V^\dagger, W^\dagger\rangle \bullet K \equiv \langle V, W\rangle^\dagger \bullet K$
(2) is also proved by an induction on V. We consider the key cases.
Case of $\langle V, W\rangle$:

$$\begin{aligned}
&(\langle V, W\rangle :_v K) \equiv \langle V^\dagger, W^\dagger\rangle \bullet K \longrightarrow^v_{(name)} V^\dagger \bullet x.(\langle x, W^\dagger\rangle \bullet K)\\
&\longrightarrow^v_{(\beta_R)} V :_v x.(\langle x, W^\dagger\rangle \bullet K) \overset{I.H.}{\longrightarrow^{v*}} V^v \bullet x.(\langle x, W^\dagger\rangle \bullet K) \longrightarrow^v_{(\beta_L)} \langle V^v, W^\dagger\rangle \bullet K\\
&\longrightarrow^v_{(name)} W^\dagger \bullet y.(\langle V^v, y\rangle \bullet K) \longrightarrow^v_{(\beta_R)} W :_v y.(\langle V^v, y\rangle \bullet K) \overset{I.H.}{\longrightarrow^{v*}} W^v \bullet y.(\langle V^v, y\rangle \bullet K)\\
&\longrightarrow^v_{(\beta_L)} \langle V^v, W^v\rangle \bullet K \equiv \langle V, W\rangle^v \bullet K
\end{aligned}$$

Cases of $\langle V\rangle$inl and $\langle W\rangle$inr: this case can be proved in a way similar to the above case using the induction hypothesis and (*name*)-rule.
(3) is proved by a straightforward induction on M and S. The key case is:

$$(x :_v K)[^{V^v}/_x] \equiv (x \bullet K)[^{V^v}/_x] \equiv V^v \bullet K[^{V^v}/_x] \overset{(1)}{\longrightarrow^{v*}} (V :_v K[^{V^v}/_x])$$

□

Definition 2.5
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.

$$\begin{array}{ll}
\Psi(\{-\}N, K) \equiv N^\dagger @ K & \Psi((\lambda x.M)\{-\}, K) \equiv x.(M :_v K)\\
\Psi(\mathrm{fst}(-), K) \equiv \mathrm{fst}[K] & \Psi(\mathrm{inl}(-), K) \equiv x.(\langle x\rangle\mathrm{inl} \bullet K)\\
\Psi(\mathrm{snd}\{-\}, K) \equiv \mathrm{snd}[K] & \Psi(\mathrm{inr}(-), K) \equiv y.(\langle y\rangle\mathrm{inr} \bullet K)\\
\Psi(\langle\{-\}, M\rangle, K) \equiv x.(\langle x, M^\dagger\rangle \bullet K) & \Psi(\langle V, \{-\}\rangle, K) \equiv y.(\langle V^v, y\rangle \bullet K)\\
\multicolumn{2}{l}{\Psi(\delta(\{-\}, x.M, y.N), K) \equiv [x.(M :_v K), y.(N :_v K)]}
\end{array}$$

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

$$\begin{array}{ll}
\Psi([\alpha]\{-\}) \equiv \alpha & \Psi(\delta(\{-\}, x.S, y.T)) \equiv [x.S^\dagger, y.T^\dagger]\\
\Psi(\{-\}N) \equiv \mathrm{not}\langle N^\dagger\rangle & \Psi((\lambda x.S)\{-\}) \equiv x.S^\dagger
\end{array}$$

About this notation, the following properties hold.

Lemma 2.20

(1) Let M not be a λ-abstraction, then $DC \vdash (E_v\{M\} :_v K) \longrightarrow^{v*} (M :_v \Psi(E_v, K))$ and $DC \vdash (D_v\{M\})^\dagger \longrightarrow^{v*} (M :_v \Psi(D_v))$ hold.

(2) Let E be an elimination context or $(\lambda x.M)\{-\}$, and D_v be an evaluation singular statement context. Then $DC \vdash (M :_v \Psi(E, K)) \longrightarrow^{v*} (E\{M\} :_v K)$ and $DC \vdash (M :_v \Psi(D_v)) \longrightarrow^{v*} (D_v\{M\})^\dagger$ hold for any term M of the $\lambda\mu$-calculus.

(3) Let E_i be an introduction context of the $\lambda\mu$-calculus. Then $DC \vdash (M :_v \Psi(E_i, K)) \longrightarrow^{v*} ((\lambda x.E_i\{x\})M :_v K)$ for any term M.

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

(5) Let D_v be an evaluation singular statement context of the call-by-value $\lambda\mu$-calculus. Then $DC \vdash (M :_v K)[{}^{\Psi(D_v)}/_{\alpha}] \longrightarrow^{v*} M[{}^{D_v\{-\}}/_{[\alpha]\{-\}}] :_v K[{}^{\Psi(D_v)}/_{\alpha}]$, and $DC \vdash S^\dagger[{}^{\Psi(D_v)}/_{\alpha}] \longrightarrow^{v*} (S\,[{}^{D_v\{-\}}/_{[\alpha]\{-\}}])^\dagger$ hold for any coterm K.

Proof. (1) Since M is not a λ-abstraction, we can immediately show $(D_v\{M\} :_v K) \equiv (M :_v \Psi(D_v))$ by the definition. If E_v is an elimination context or $(\lambda x.M)\{-\}$, then we have $(E_v\{M\} :_v K) \equiv (M :_v \Psi(E_v, K))$. For the introduction contexts, the claim is proved by a case analysis of E_v.

Case of $\langle\{-\}, N\rangle$:

$$\begin{aligned}\langle M, N\rangle :_v K \equiv \langle M^\dagger, N^\dagger\rangle \bullet K &\longrightarrow^{v}_{(name)} M^\dagger \bullet x.(\langle x, N^\dagger\rangle \bullet K)\\ &\longrightarrow^{v}_{(\beta_R)} (M :_v x.(\langle x, N^\dagger\rangle \bullet K)) \equiv (M :_v \Psi(\langle\{-\}, N\rangle, K))\end{aligned}$$

Case of $\langle V, \{-\}\rangle$:

$$\begin{aligned}&\langle V, M\rangle :_v K \equiv \langle V^\dagger, M^\dagger\rangle \bullet K \longrightarrow^{v}_{(name)} V^\dagger \bullet x.(\langle x, M^\dagger\rangle \bullet K) \longrightarrow^{v}_{(\beta_R)} V :_v x.(\langle x, M^\dagger\rangle \bullet K)\\ &\quad \overset{Lem\ 2.19(2)}{\longrightarrow^{v}} V^v \bullet x.(\langle x, M^\dagger\rangle \bullet K) \longrightarrow^{v}_{(\beta_L)} \langle V^v, M^\dagger\rangle \bullet K \longrightarrow^{v}_{(name)} M^\dagger \bullet y.(\langle V^v, y\rangle \bullet K)\\ &\quad \longrightarrow^{v}_{(\beta_R)} (M :_v y.(\langle V^v, y\rangle \bullet K)) \equiv (M :_v \Psi(\langle V, \{-\}\rangle, K))\end{aligned}$$

Case of $\mathrm{inl}(-)$:

$$\begin{aligned}\mathrm{inl}(M) :_v K \equiv \langle M^\dagger\rangle\mathrm{inl} \bullet K &\longrightarrow^{v}_{(name)} M^\dagger \bullet x.(\langle x\rangle\mathrm{inl} \bullet K)\\ &\longrightarrow^{v}_{(\beta_R)} M :_v x.(\langle x\rangle\mathrm{inl} \bullet K) \equiv M :_v \Psi(\mathrm{inl}(-), K)\end{aligned}$$

Case of 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:

$$\begin{aligned}\lambda x.S :_v \Psi(\{-\}N) &\equiv \lambda x.S :_v \mathrm{not}\langle N^\dagger\rangle \equiv [x.S^\dagger]\mathrm{not} \bullet \mathrm{not}\langle N^\dagger\rangle \\ &\longrightarrow^v_{(\beta_\neg)} N^\dagger \bullet x.S^\dagger \longrightarrow^v_{(\beta_R)} N :_v x.S^\dagger \equiv ((\lambda x.S)N)^\dagger\end{aligned}$$

For E, the key case is: M is $\lambda x.M$ and E is $\{-\}N$. 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 $\langle\{-\}, N\rangle$:

$$\begin{aligned}(M :_v \Psi(\langle\{-\}, N\rangle, K)) &\equiv M :_v x.(\langle x, N^\dagger\rangle \bullet K) \longrightarrow^v_{(\eta_R)} M :_v x.(\langle x^\dagger, N^\dagger\rangle \bullet K) \\ &\equiv M :_v x.(\langle x, N\rangle :_v K) \equiv (\lambda x.\langle x, N\rangle)M :_v K\end{aligned}$$

Case of $\langle V, \{-\}\rangle$:

$$\begin{aligned}M :_v \Psi(\langle V, \{-\}\rangle, K) &\equiv (M :_v y.(\langle V^v, y\rangle \bullet K)) \overset{Lem\ 2.19(1)}{\longrightarrow^v} (M :_v y.(\langle V^\dagger, y\rangle \bullet K)) \\ &\longrightarrow^v_{(\eta_R)} (M :_v y.(\langle V^\dagger, y^\dagger\rangle \bullet K)) \equiv (M :_v y.(\langle V, y\rangle :_v K)) \equiv (\lambda y.\langle V, y\rangle)M :_v K\end{aligned}$$

Case of inl(−):

$$M :_v \Psi(\mathrm{inl}(-), K) \equiv M :_v x.(\langle x\rangle\mathrm{inl} \bullet K) \equiv M :_v x.(\mathrm{inl}(x) :_v K) \equiv (\lambda x.\mathrm{inl}(x))M :_v K$$

Case of inr(−): this case is proved in a way similar to the above case.

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

$$\begin{aligned}([\alpha]M)[{}^{\Psi(E,\beta)}/_{\alpha}] &\equiv (M :_v \alpha)[{}^{\Psi(E,\beta)}/_{\alpha}] \overset{I.H.}{\longrightarrow^{v*}} M[{}^{[\beta]E\{-\}}/_{[\alpha]\{-\}}] :_v \Psi(E, \beta) \\ &\overset{(2)}{\longrightarrow^{v*}} E\{M[{}^{[\beta]E\{-\}}/_{[\alpha]\{-\}}]\} :_v \beta \equiv ([\beta]E\{M[{}^{[\beta]E\{-\}}/_{[\alpha]\{-\}}]\})^\dagger \\ &\equiv (([\alpha]M)[{}^{[\beta]E\{-\}}/_{[\alpha]\{-\}}])^\dagger\end{aligned}$$

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

$$\begin{aligned}([\alpha]M)[{}^{\Psi(D_v)}/_{\alpha}] &\equiv (M :_v \alpha)[{}^{\Psi(D_v)}/_{\alpha}] \overset{I.H.}{\longrightarrow^{v*}} M[{}^{D_v\{-\}}/_{[\alpha]\{-\}}] :_v \Psi(D_v) \\ &\overset{(2)}{\longrightarrow^{v*}} (D_v\{M[{}^{D_v\{-\}}/_{[\alpha]\{-\}}]\})^\dagger \equiv (([\alpha]M)[{}^{D_v\{-\}}/_{[\alpha]\{-\}}])^\dagger\end{aligned}$$

□

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

Theorem 2.21 (Soundness of $(-)^{\dagger}$)
(1) If $\lambda\mu \vdash M \longrightarrow_v N$, then $DC \vdash (M :_v K) \longrightarrow^{v*} (N :_v K)$ for any coterm K, especially $DC \vdash M^{\dagger} \longrightarrow^{v*} N^{\dagger}$.
(2) If $\lambda\mu \vdash S \longrightarrow_v T$, then $DC \vdash S^{\dagger} \longrightarrow^{v*} T^{\dagger}$.
Moreover, if $\longrightarrow_v$ is $(\beta_{\supset})$, $(\beta_{\wedge})$, $(\beta_{\vee})$ or $(\beta_{\neg})$, then $\longrightarrow^{v*}$ can be replaced by $\longrightarrow^{v+}$.

Proof. (1) and (2) are proved by simultaneous induction on $\longrightarrow_v$. Base cases are shown as follows.
Case of $(\beta_{\supset})$:

$$\begin{aligned}(\lambda x.M)V :_v K \equiv V :_v x.(M :_v K) &\overset{\textit{Lem } 2.19(2)}{\longrightarrow^{v*}} V^v \bullet x.(M :_v K) \\ &\longrightarrow^{v}_{(\beta_L)} (M :_v K)[^{V^v}/_x] \overset{\textit{Lem } 2.19(3)}{\longrightarrow^{v*}} M[^V/_x] :_v K\end{aligned}$$

Case of $(\beta_{\wedge})$:

$$\begin{aligned}\mathrm{fst}\langle V, W\rangle :_v K \equiv \langle V, W\rangle :_v \mathrm{fst}[K] &\overset{\textit{Lem } 2.19(2)}{\longrightarrow^{v*}} \langle V, W\rangle^v \bullet \mathrm{fst}[K] \\ &\equiv \langle V^v, W^v\rangle \bullet \mathrm{fst}[K] \longrightarrow^{v}_{(\beta_{\wedge})} V^v \bullet K \overset{\textit{Lem } 2.19(1)}{\longrightarrow^{v*}} V :_v K\end{aligned}$$

The other rule of $(\beta_{\wedge})$ is proved similarly.
Case of $(\beta_{\vee})$:

$$\begin{aligned}\delta(\mathrm{inl}(V), x.M, y.N) :_v K &\equiv \mathrm{inl}(V) :_v [x.(M :_v K), y.(N :_v K)] \\ &\overset{\textit{Lem } 2.19(2)}{\longrightarrow^{v*}} \langle V^v\rangle\mathrm{inl} \bullet [x.(M :_v K), y.(N :_v K)] \\ &\longrightarrow^{v}_{(\beta_{\vee})} V^v \bullet x.(M :_v K) \longrightarrow^{v}_{(\beta_L)} (M :_v K)[^{V^v}/_x] \\ &\overset{\textit{Lem } 2.19(3)}{\longrightarrow^{v*}} M[^V/_x] :_v K\end{aligned}$$

The other rules of $(\beta_{\vee})$ are proved similarly.
Case of $(\beta_{\neg})$:

$$((\lambda x.S)V)^{\dagger} \equiv V :_v x.S^{\dagger} \overset{\textit{Lem } 2.19(2)}{\longrightarrow^{v*}} V^v \bullet_v x.S^{\dagger} \longrightarrow^{v}_{(\beta_L)} S^{\dagger}[^{V^v}/_x] \overset{\textit{Lem } 2.19(3)}{\longrightarrow^{v*}} (S[^V/_x])^{\dagger}$$

Case of (ζ): Let $E_{e\lambda}$ be an elimination context of $(\lambda x.M)\{-\}$, then

$$\begin{aligned}E_{e\lambda}\{\mu\alpha.S\} :_v K &\overset{\textit{Lem } 2.20(1)}{\longrightarrow^{v*}} \mu\alpha.S :_v \Psi(E_{e\lambda}, K) \equiv S^{\dagger}[^{\Psi(E_{e\lambda},K)}/_{\alpha}] \equiv S^{\dagger}[^{\Psi(E_{e\lambda},\beta)}/_{\alpha}][^K/_{\beta}] \\ &\overset{\textit{Lem } 2.20(4)}{\longrightarrow^{v*}} (S[^{[\beta]E_{e\lambda}\{-\}}/_{[\alpha]\{-\}}])^{\dagger}[^K/_{\beta}] \equiv \mu\beta.S[^{[\beta]E_{e\lambda}\{-\}}/_{[\alpha]\{-\}}] :_V K\end{aligned}$$

The other rule of (ζ) is proved similarly.
Case of $(comp)$:

$$\begin{aligned} E_{e\lambda}\{(\lambda x.M)N\} :_v K &\overset{Lem\ 2.20(1)}{\longrightarrow^{v*}} (\lambda x.M)N :_v \Psi(E_{e\lambda}, K) \equiv N :_v x.(M :_v \Psi(E_{e\lambda}, K)) \\ &\overset{Lem\ 2.20(2)}{\longrightarrow^{v*}} N :_v x.(E_{e\lambda}\{M\} :_v K) \equiv (\lambda x.E_{e\lambda}\{M\})N :_v K \end{aligned}$$

The other rule of (ζ) is proved similarly.
Case of (π):

$$\begin{aligned} E_{e\lambda}\{\delta(O, x.M, y.N)\} :_v K &\overset{Lem\ 2.20(1)}{\longrightarrow^{v*}} \delta(O, x.M, y.N) :_v \Psi(E_{e\lambda}, K) \\ &\equiv O :_v [x.(M :_v \Psi(E_{e\lambda}, K)), y.(N :_v \Psi(E_{e\lambda}, K))] \\ &\overset{Lem\ 2.20(2)}{\longrightarrow^{v*}} O :_v [x.(E_{e\lambda}\{M\} :_v K), y.(E_{e\lambda}\{N\} :_v K)] \\ &\equiv \delta(O, x.E_{e\lambda}\{M\}, y.E_{e\lambda}\{N\}) :_v K \end{aligned}$$

The other rule of (ζ) is proved similarly.
Case of (η_μ):

$$\mu\alpha.[\alpha]M :_v K \equiv ([\alpha]M)^\dagger[{}^K/_\alpha] \equiv (M :_v \alpha)[{}^K/_\alpha] \equiv M :_v K$$

Case of $(name)$: Let O not be a value. Then

$$\begin{aligned} E_i\{O\} :_v K &\overset{Lem\ 2.20(1)}{\longrightarrow^{v*}} O :_v \Psi(E_i, K) \overset{Lem\ 2.20(3)}{\longrightarrow^{v*}} (\lambda x.E_i\{x\})O :_v K, \text{ and} \\ E_e\{O\} :_v K &\overset{Lem\ 2.20(1)}{\longrightarrow^{v*}} O :_v \Psi(E_e, K) \longrightarrow^{v}_{\eta_L} O :_v x.(x \bullet \Psi(E_e, K)) \\ &\overset{Lem\ 2.20(2)}{\longrightarrow^{v*}} O :_v x.(E_e\{x\} :_v K) \equiv (\lambda x.E_e\{x\})O :_v K\,. \end{aligned}$$

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

2.5 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.

2.5.1 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 2.6 (The naive translation from DC into $\lambda\mu$)
The naive translation from the dual calculus into the $\lambda\mu$-calculus is defined as follows. This translation $(-)_\circ$ maps a term M and a statement S of the dual calculus to a term $M_\circ$ and a statement $S_\circ$ of the $\lambda\mu$-calculus respectively, and maps a coterm K with a term O of the $\lambda\mu$-calculus to a statement $K_\circ\{O\}$ of the $\lambda\mu$-calculus.

$$\begin{array}{rlrl}
(x)_\circ \equiv x & & \alpha_\circ\{O\} \equiv [\alpha]O \\
(\langle M, N\rangle)_\circ \equiv \langle M_\circ, N_\circ\rangle & & [K, L]_\circ\{O\} \equiv \delta(O, x.K_\circ\{x\}, y.L_\circ\{y\}) \\
(\langle M\rangle\mathrm{inl})_\circ \equiv \mathrm{inl}(M_\circ) & & (\mathrm{fst}[K])_\circ\{O\} \equiv K_\circ\{\mathrm{fst}(O)\} \\
(\langle N\rangle\mathrm{inr})_\circ \equiv \mathrm{inr}(N_\circ) & & (\mathrm{snd}[L])_\circ\{O\} \equiv L_\circ\{\mathrm{snd}(O)\} \\
([K]\mathrm{not})_\circ \equiv \lambda x.K_\circ\{x\} & & \mathrm{not}\langle M\rangle_\circ\{O\} \equiv OM_\circ \\
(\lambda x.M)_\circ \equiv \lambda x.M_\circ & & (M@K)_\circ\{O\} \equiv K_\circ\{OM_\circ\} \\
(S.\alpha)_\circ \equiv \mu\alpha.S_\circ & & (x.S)_\circ\{O\} \equiv (\lambda x.S_\circ)O
\end{array}$$
$$(M \bullet K)_\circ \equiv K_\circ\{M_\circ\}$$

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

Lemma 2.22
Let M be a term, K be a coterm, S be a statement of the dual calculus, and O be a term of the $\lambda\mu^{wad}$-calculus. Then

(1) $\lambda\mu \vdash [\![M_*]\!] =^n M_\circ$, $\lambda\mu \vdash [\![K_*\{O\}]\!] =^n K_\circ\{[\![O]\!]\}$, and $\lambda\mu \vdash [\![S_*]\!] =^n S_\circ$;

(2) $\lambda\mu \vdash [\![M_*]\!] =^v M_\circ$, $\lambda\mu \vdash [\![K_*\{O\}]\!] =^v K_\circ\{[\![O]\!]\}$, and $\lambda\mu \vdash [\![S_*]\!] =^v S_\circ$.

Proof. (1) is proved by induction on M, K, and S. We give the cases of sums, *i.e.*, $\langle M\rangle$inl, $\langle N\rangle$inl, and $[K, L]$.
Case of $\langle M\rangle$inl :

$$[\![\langle M\rangle\mathrm{inl}_*]\!] \equiv [\![\mu(\alpha,\beta).[\alpha]M_*]\!] \equiv \mu\gamma.([\alpha][\![M_*]\!])[{}^{[\gamma]\mathrm{inl}(-)}/_{[\alpha]\{-\}}, {}^{[\gamma]\mathrm{inr}(-)}/_{[\beta]\{-\}}]$$
$$\equiv \mu\gamma.[\gamma]\mathrm{inl}([\![M_*]\!]) \overset{I.H.}{=}_n \mu\gamma.[\gamma]\mathrm{inl}(M_\circ) =_n \mathrm{inl}(M_\circ) \equiv (\langle M\rangle\mathrm{inl})_\circ$$

Case of $\langle N\rangle\mathrm{inr}$: this case is proved in a way similar to the above case.
Case of $[K, L]$:

$$\begin{aligned}
&[\![[K, L]_*\{O\}]\!] \equiv [\![L_*\{\mu\beta.K_*\{\mu\alpha.[\alpha,\beta]O\}\}]\!]\\
&\overset{I.H.}{=_n} L_\circ\{[\![\mu\beta.K_*\{\mu\alpha.[\alpha,\beta]O\}]\!]\} \equiv L_\circ\{\mu\beta.[\![K_*\{\mu\alpha.[\alpha,\beta]O\}]\!]\}\\
&\overset{I.H.}{=_n} L_\circ\{\mu\beta.K_\circ\{[\![\mu\alpha.[\alpha,\beta]O]\!]\}\} \equiv L_\circ\{\mu\beta.K_\circ\{\mu\alpha.\delta([\![O]\!], x.[\alpha]x, y.[\beta]y)\}\}\\
&\overset{(*)}{=_n} \delta([\![O]\!], x.K_\circ\{x\}, y.L_\circ\{y\}) \equiv [K, L]_\circ\{[\![O]\!]\}
\end{aligned}$$

$(*)$ comes from the claim: $K_\circ\{\mu\alpha.S\} =_n S[{}^{K_\circ\{-\}}/_{[\alpha]\{-\}}]$. This claim is proved by a straightforward induction on K.

(2) is proved by 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). □

The naive translation preserves typing rules and equalities.

Proposition 2.23

(1) If $\Gamma \vdash \Delta \mid M : A$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid M_\circ : A$;
if $K : A \mid \Gamma \vdash \Delta$ and $\Gamma \vdash_{\lambda\mu} \Delta \mid O : A$, then $\Gamma \mid K_\circ\{O\} \vdash_{\lambda\mu} \Delta$; and
if $\Gamma \mid S \vdash \Delta$, then $\Gamma \mid S_\circ \vdash_{\lambda\mu} \Delta$.

(2) If $DC \vdash M =^n N$, then $\lambda\mu \vdash M_\circ =_n N_\circ$;
if $DC \vdash K =^n L$, then $\lambda\mu \vdash K_\circ\{O\} =_n L_\circ\{O\}$; and
if $DC \vdash S =^n T$, then $\lambda\mu \vdash S_\circ =_n T_\circ$.

(3) If $DC \vdash M =^v N$, then $\lambda\mu \vdash M_\circ =_v N_\circ$;
if $DC \vdash K =^v L$, then $\lambda\mu \vdash K_\circ\{O\} =_v L_\circ\{O\}$; and
if $DC \vdash S =^v T$, then $\lambda\mu \vdash S_\circ =_v T_\circ$.

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 2.28 using Lemma 2.24.
(3) As in (2), we can show this claim directly, or as a corollary of Theorem 2.34 using Lemma 2.30. □

In general, this naive translation does not preserve reductions as well as Wadler's translation. (η_L)-rule is a counter-example for the call-by-name system:

$$\alpha_\circ\{O\} \equiv [\alpha]O \longleftarrow_n (\lambda x.[\alpha]x)O \equiv (x.(x \bullet \alpha))_\circ\{O\}$$

On the other hand, $(\beta_\neg)$-rule is a counter-example for the call-by-value system:

$$
\begin{gathered}
([\alpha]\mathrm{not} \bullet \mathrm{not}\langle(x \bullet \beta).\gamma\rangle)_\circ \equiv (\mathrm{not}\langle(x \bullet \beta).\gamma\rangle)_\circ\{[\alpha]\mathrm{not}_\circ\} \equiv (\lambda x.[\alpha]x)\mu\gamma.[\beta]x \\
\longleftarrow_{(name)} [\alpha]\mu\gamma.[\beta]x \equiv \alpha_\circ\{((x \bullet \beta).\gamma)\circ\} \equiv ((x \bullet \beta).\gamma \bullet \alpha)_\circ
\end{gathered}
$$

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.

2.5.2 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 2.7 ($(-)_\sharp$: CBN dual calculus $\longrightarrow$ CBN $\lambda\mu$-calculus)
We introduce the new translation $(-)_\sharp$ by modifying the definition of $(-)_\circ$ as follows.

$$
\begin{array}{rlrl}
(x)_\sharp & \equiv x & \alpha_\sharp\{O\} & \equiv [\alpha]O \\
(\langle M, N\rangle)_\sharp & \equiv \langle M_\sharp, N_\sharp\rangle & [K, L]_\sharp\{O\} & \equiv \delta(O, x.K_\sharp\{x\}, y.L_\sharp\{y\}) \\
(\langle M\rangle\mathrm{inl})_\sharp & \equiv \mathrm{inl}(M_\sharp) & (\mathrm{fst}[K])_\sharp\{O\} & \equiv K_\sharp\{\mathrm{fst}(O)\} \\
(\langle N\rangle\mathrm{inr})_\sharp & \equiv \mathrm{inr}(N_\sharp) & (\mathrm{snd}[L])_\sharp\{O\} & \equiv L_\sharp\{\mathrm{snd}(O)\} \\
([K]\mathrm{not})_\sharp & \equiv \lambda x.K_\sharp\{x\} & \mathrm{not}\langle M\rangle_\sharp\{O\} & \equiv OM_\sharp \\
(\lambda x.M)_\sharp & \equiv \lambda x.M_\sharp & (M@K)_\sharp\{O\} & \equiv K_\sharp\{OM_\sharp\} \\
(S.\alpha)_\sharp & \equiv \mu\alpha.S_\sharp & (x.S)_\sharp\{O\} & \equiv S_\sharp[{}^O/_x]
\end{array}
$$
$$
(M \bullet K)_\sharp \equiv K_\sharp\{M_\sharp\}
$$

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

Lemma 2.24
Let M be a term, K be a coterm, and S be a statement of the dual calculus. Then

(1) $\lambda\mu \vdash M_\circ \longrightarrow_n^* M_\sharp$;

(2) $\lambda\mu \vdash K_\circ\{O\} \longrightarrow_n^* K_\sharp\{O\}$ for any term O of the $\lambda\mu$-calculus; and

(3) $\lambda\mu \vdash S_\circ \longrightarrow_n^* S_\sharp$.

Proposition 2.25
(1) If $\Gamma \vdash_{dc} \Delta \mid M : A$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid M_\sharp : A$.
(2) If $K : A \mid \Gamma \vdash_{dc} \Delta$ and $\Gamma \vdash_{\lambda\mu} \Delta \mid O : A$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid K_\sharp\{O\}$.
(3) If $\Gamma \mid S \vdash_{dc} \Delta$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid S_\sharp$.

Proof. These claims can be shown by the subject reduction property for the $\lambda\mu$-calculus using Proposition 2.23 (1) and Lemma 2.24. $\square$

Lemma 2.26

(1) If $\lambda\mu \vdash O \longrightarrow_n O'$, then $\lambda\mu \vdash K_\sharp\{O\} \longrightarrow_n^* K_\sharp\{O'\}$ for any coterm K.

(2) $M_\sharp[{}^{N_\sharp}/_x] \equiv (M[{}^N/_x])_\sharp$, $K_\sharp\{O\}[{}^{N_\sharp}/_x] \equiv (K[{}^N/_x])_\sharp\{O[{}^{N_\sharp}/_x]\}$ 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 $\lambda\mu$-calculus.

(4) $M_\sharp[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}] \equiv (M[{}^L/_\alpha])_\sharp$, $(K_\sharp\{O\})[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}] \equiv (K[{}^L/_\alpha])_\sharp\{O[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}]\}$, and $S_\sharp[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}] \equiv (S[{}^L/_\alpha])_\sharp$ 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:

$$(\alpha_\sharp\{O\})[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}] \equiv ([\alpha]O)[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}] \equiv L_\sharp\{O[{}^{L_\sharp\{-\}}/_{[\alpha]\{-\}}]\}$$

$\square$

Let F be a coterm context of the dual calculus, and O be a term of the $\lambda\mu$-calculus. Then we define term $F_\sharp\{O\}$ of the $\lambda\mu$-calculus as follows:

$$\begin{array}{ll}
(\{-\})_\sharp\{O\} \equiv O & (M@\{-\})_\sharp\{O\} \equiv OM_\sharp \\
(\mathrm{fst}[-])_\sharp\{O\} \equiv \mathrm{fst}(O) & ([-,P])_\sharp\{O\} \equiv \mu\alpha.\delta(O, x.[\alpha]x, y.P_\sharp\{y\}) \\
(\mathrm{snd}[-])_\sharp\{O\} \equiv \mathrm{snd}(O) & ([K,-])_\sharp\{O\} \equiv \mu\beta.\delta(O, x.K_\sharp\{x\}, y.[\beta]y)
\end{array}$$

This notation satisfies the following property.

Lemma 2.27

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

(1) $\lambda\mu \vdash F\{L\}_\sharp\{O\} \longrightarrow_n^* L_\sharp\{F_\sharp\{O\}\}$, and

(2) $\lambda\mu \vdash P_\sharp\{F_\sharp\{O\}\} \longrightarrow_n^* F\{P\}_\sharp\{O\}$

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 $[-, P]$:

$$[L, P]_\sharp\{O\} \equiv \delta(O, x.L_\sharp\{x\}, y.P_\sharp\{y\}) \longrightarrow_{(\nu)} (\lambda x.L_\sharp\{x\})\mu\alpha.\delta(O, x.[\alpha]x, y.P_\sharp\{y\})$$
$$\longrightarrow_{(\beta_\neg)} L_\sharp\{\,\mu\alpha.\delta(O, x.[\alpha]x, y.P_\sharp\{y\})\,\} \equiv L_\sharp\{\,[-, P]_\sharp\{O\}\,\}$$

Case of $[K, -]$:

$$[K, L]_\sharp\{O\} \equiv \delta(O, x.K_\sharp\{x\}, y.L_\sharp\{y\}) \longrightarrow_{(\nu)} (\lambda y.L_\sharp\{y\})\mu\beta.\delta(O, x.K_\sharp\{x\}, y.[\beta]y)$$
$$\longrightarrow_{(\beta_\neg)} L_\sharp\{\,\mu\beta.\delta(O, x.K_\sharp\{x\}, y.[\beta]y)\,\} \equiv L_\sharp\{\,[K, -]_\sharp\{O\}\,\}$$

(2) is also proved by a case analysis of F. We give the key cases.
Case of $[-, Q]$:

$$P_\sharp\{\,[-, Q]_\sharp\{O\}\,\} \equiv P_\sharp\{\,\mu\alpha.\delta(O, x.[\alpha]x, y.Q_\sharp\{y\})\,\}$$
$$\longrightarrow_{(\zeta)} \delta(O, x.P_\sharp\{x\}, y.Q_\sharp\{y\}) \equiv [P, Q]_\sharp\{O\}$$

Case of $[K, -]$:

$$P_\sharp\{\,[K, -]_\sharp\{O\}\,\} \equiv P_\sharp\{\,\mu\beta.\delta(O, x.K_\sharp\{x\}, y.[\beta]y)\,\}$$
$$\longrightarrow_{(\zeta)} \delta(O, x.K_\sharp\{x\}, y.P_\sharp\{y\}) \equiv [K, P]_\sharp\{O\}$$

□

We now prove that the call-by-name translation $(-)_\sharp$ preserves reductions.

Theorem 2.28 (Soundness of $(-)_\sharp$)
(1) If $DC \vdash M \longrightarrow^n N$, then $\lambda\mu \vdash M_\sharp \longrightarrow_n^* N_\sharp$.
(2) If $DC \vdash K \longrightarrow^n L$, then $\lambda\mu \vdash K_\sharp\{O\} \longrightarrow_n^* L_\sharp\{O\}$.
(3) If $DC \vdash S \longrightarrow^n T$, then $\lambda\mu \vdash S_\sharp \longrightarrow_n^* T_\sharp$.
Moreover, in (1), (2) and (3), if $\longrightarrow^n$ is $(\beta_\supset)$, $(\beta_\wedge)$, $(\beta_\vee)$ or $(\beta_\neg)$, then $\longrightarrow_n^*$ can be replaced by $\longrightarrow_n^+$.

Proof. The claims are proved by simultaneous induction on the reduction relation $\longrightarrow^n$. Base cases are shown as follows.
Case of $(\beta_\supset)$:

$$((\lambda x.M) \bullet (N@P))_\sharp \equiv (N@P)_\sharp\{\lambda x.M_\sharp\} \equiv P_\sharp\{(\lambda x.M_\sharp)N_\sharp\}$$

$$\longrightarrow_{(\beta_\supset)} P_\sharp\{M_\sharp[{}^{N_\sharp}/_x]\} \equiv (P_\sharp\{M_\sharp\})[{}^{N_\sharp}/_x] \equiv (M \bullet P)_\sharp[{}^{N_\sharp}/_x]$$
$$\equiv (x.(M \bullet P))_\sharp\{N_\sharp\} \equiv (N \bullet x.(M \bullet P))_\sharp$$

Case of $(\beta_\wedge)$:

$$(\langle M, N\rangle \bullet \mathrm{fst}[P])_\sharp \equiv \mathrm{fst}[P]_\sharp\{\langle M_\sharp, N_\sharp\rangle\} \equiv P_\sharp\{\mathrm{fst}\langle M_\sharp, N_\sharp\rangle\}$$
$$\longrightarrow_{(\beta_\wedge)} P_\sharp\{M_\sharp\} \equiv (M \bullet P)_\sharp$$

The other $(\beta_\wedge)$ case can be shown similarly.

Case of $(\beta_\vee)$:

$$(\langle M\rangle \mathrm{inl} \bullet [P, Q])_\sharp \equiv [P, Q]_\sharp\{\mathrm{inl}(M_\sharp)\}a \equiv \delta(\mathrm{inl}(M_\sharp), x.P_\sharp\{x\}, y.Q_\sharp\{y\})$$
$$\longrightarrow_{(\beta_\vee)} P_\sharp\{M_\sharp\} \equiv (M \bullet P)_\sharp$$

The other $(\beta_\wedge)$ case can be shown similarly.

Case of $(\beta_\neg)$:

$$([K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle)_\sharp \equiv (\mathrm{not}\langle M\rangle)_\sharp\{\lambda x.K_\sharp\{x\}\} \equiv (\lambda x.K_\sharp\{x\})M_\sharp$$
$$\longrightarrow_{(\beta_\neg)} K_\sharp\{M_\sharp\} \equiv (M \bullet K)_\sharp$$

Case of (β_R):

$$(S.\alpha \bullet P)_\sharp \equiv P_\sharp\{\,\mu\alpha.S_\sharp\,\} \longrightarrow_{(\zeta)} S_\sharp[{}^{P_\sharp\{-\}}/_{[\alpha](-)}] \overset{Lem\ 2.26(4)}{\equiv} (S[{}^{P}/_\alpha])_\sharp$$

Case of (β_L):

$$(M \bullet x.S)_\sharp \equiv (x.S)_\sharp\{M_\sharp\} \equiv S_\sharp[{}^{M_\sharp}/_x] \overset{Lem\ 2.26(2)}{\equiv} (S[{}^{M}/_x])_\sharp$$

Case of (η_R):

$$M_\sharp \longrightarrow_{(\eta_\mu)} \mu\alpha.[\alpha]M_\sharp \equiv (\,(M \bullet \alpha).\alpha\,)_\sharp$$

Case of (η_L):

$$K_\sharp\{O\} \equiv (K_\sharp\{x\})[{}^{O}/_x] \equiv (x \bullet K)_\sharp[{}^{O}/_x] \equiv (x.(x \bullet K))_\sharp\{O\}$$

Case of (*name*): Note that K is not a covalue.

$$(\,M \bullet F\{K\}\,)_\sharp \equiv F\{K\}_\sharp\{M_\sharp\} \overset{Lem\ 2.27(1)}{\longrightarrow_n^*} K_\sharp\{F_\sharp\{M_\sharp\}\} \longrightarrow_{(\eta_\mu)} K_\sharp\{\mu\alpha.[\alpha]F_\sharp\{M_\sharp\}\}$$
$$\equiv K_\sharp\{\mu\alpha.\alpha_\sharp\{\,F_\sharp\{M_\sharp\}\,\}\} \overset{Lem\ 2.27(2)}{\longrightarrow_n^*} K_\sharp\{\mu\alpha.F\{\alpha\}_\sharp\{M_\sharp\}\}$$
$$\equiv ((M \bullet F\{\alpha\}).\alpha \bullet K)_\sharp$$

Induction cases can be shown easily. □

2.5.3 The translation from CBV dual calculus into CBV $\lambda\mu$-calculus

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

Definition 2.8 ($(-)_\dagger$: CBV dual calculus $\longrightarrow$ CBV $\lambda\mu$-calculus)
We define the translation $(-)_\dagger$ as follows.

$$
\begin{aligned}
(x)_\dagger &\equiv x & \alpha_\dagger\{O\} &\equiv [\alpha]O \\
(\langle M, N\rangle)_\dagger &\equiv \langle M_\dagger, N_\dagger\rangle & [K, L]_\dagger\{O\} &\equiv \delta(O, x.K_\dagger\{x\}, y.L_\dagger\{y\}) \\
(\langle M\rangle\mathrm{inl})_\dagger &\equiv \mathrm{inl}(M_\dagger) & (\mathrm{fst}[K])_\dagger\{O\} &\equiv K_\dagger[\mathrm{fst}(O)] \\
(\langle N\rangle\mathrm{inr})_\dagger &\equiv \mathrm{inr}(N_\dagger) & (\mathrm{snd}[L])_\dagger\{O\} &\equiv L_\dagger[\mathrm{snd}(O)] \\
([K]\mathrm{not})_\dagger &\equiv \lambda x.K_\dagger\{x\} & \mathrm{not}\langle M\rangle_\dagger\{O\} &\equiv OM_\dagger \\
(\lambda x.M)_\dagger &\equiv \lambda x.M_\dagger & (M@K)_\dagger\{O\} &\equiv K_\dagger[OM_\dagger] \\
(S.\alpha)_\dagger &\equiv \mu\alpha.S_\dagger & (x.S)_\dagger\{O\} &\equiv (\overline{\lambda}_v x.S_\dagger)O \\
(M \bullet K)_\dagger &\equiv K_\dagger[M_\dagger] & K_\dagger[O] &\equiv (\overline{\lambda}_v x.K_\dagger\{x\})O
\end{aligned}
$$

For the call-by-value translation, we use the two notations $K_\dagger\{O\}$ and $K_\dagger[O]$. The relation between these two notations is as follows.

Lemma 2.29
Let O be a term of the $\lambda\mu$-calculus, and K be a coterm, then

(1) $K_\dagger\{V\} \equiv K_\dagger[V]$ for any value V; and

(2) $\lambda\mu \vdash K_\dagger\{O\} \longrightarrow_v^* K_\dagger[O]$.

Proof. (1) is immediately shown. We show (2) when O is not a value by a case analysis of K.
Case of α: $\alpha_\dagger\{O\} \equiv [\alpha]O \longrightarrow_{(name)} (\lambda z.[\alpha]z)O \equiv \alpha_\dagger[O]$
Case of $[K, L]$:

$$[K, L]_\dagger\{O\} \equiv \delta(O, x.K_\dagger\{x\}, y.L_\dagger\{y\}) \longrightarrow_{(name)} (\lambda z.\delta(z, x.K_\dagger\{x\}, y.L_\dagger\{y\}))O \equiv [K, L]_\dagger[O]$$

Case of $\mathrm{fst}[K]$:

$$
\begin{aligned}
\mathrm{fst}[K]_\dagger\{O\} \equiv K_\dagger[\mathrm{fst}(O)] \equiv (\lambda z.K_\dagger\{z\})\mathrm{fst}(O) &\longrightarrow_{(name)} (\lambda z.K_\dagger\{z\})((\lambda x.\mathrm{fst}(x))O) \\
&\longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})\mathrm{fst}(x))O \equiv (\lambda x.(K_\dagger[\mathrm{fst}(x)])O
\end{aligned}
$$

$$\equiv (\lambda x.(\mathrm{fst}[K]_\dagger\{x\})O \equiv \mathrm{fst}[K]_\dagger[O]$$

Case of $\mathrm{snd}[K]$: This case can be shown in a way similar to the above case.
Case of $\mathrm{not}\langle M\rangle$:

$$\mathrm{not}\langle M\rangle_\dagger\{O\} \equiv OM_\dagger \longrightarrow_{(name)} (\lambda z.zM_\dagger)O \equiv (\lambda z.\mathrm{not}\langle M\rangle_\dagger\{z\})O \equiv \mathrm{not}\langle M\rangle_\dagger[O]$$

Case of $M@K$:

$$\begin{aligned}(M@K)_\dagger\{O\} \equiv K_\dagger[OM_\dagger] \equiv (\lambda z.K_\dagger\{z\})OM_\dagger &\longrightarrow_{(name)} (\lambda z.K_\dagger\{z\})((\lambda x.xM_\dagger)O)\\ &\longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})(xM_\dagger))O \equiv (\lambda x.(K_\dagger[xM_\dagger])O\\ &\equiv (\lambda x.(M@K)_\dagger\{x\})O \equiv (M@K)_\dagger[O]\end{aligned}$$

Case of $x.S$:

$$(x.S)_\dagger\{O\} \equiv (\lambda x.S_\dagger)O \equiv (\lambda z.S_\dagger[^z/_x])O \equiv (\lambda z.(x.S)_\dagger\{z\})O \equiv (x.S)_\dagger[O]$$

□

The call-by-value translation, $(-)_\dagger$, is consistent with the naive translation as well as the call-by-name translation.

Lemma 2.30
Let M be a term, K be a coterm, and S be a statement of the dual calculus, then

(1) $\lambda\mu \vdash M_\circ \longrightarrow_v^* M_\dagger$,

(2) $\lambda\mu \vdash K_\circ\{O\} \longrightarrow_v^* K_\dagger\{O\}$ for any term O of the $\lambda\mu$-calculus , and

(3) $\lambda\mu \vdash S_\circ \longrightarrow_v^* S_\dagger$.

Proof. The claims are shown by a simultaneous induction on M, K, and S. We consider the key cases.
Case of $\mathrm{fst}[K]$:

$$\mathrm{fst}[K]_\circ\{O\} \equiv K_\circ\{\mathrm{fst}(O)\} \overset{I.H.}{\longrightarrow_v^*} K_\dagger\{\mathrm{fst}(O)\} \overset{Lem\ 2.29}{\longrightarrow_v^*} K_\dagger[\mathrm{fst}(O)] \equiv \mathrm{fst}[K]_\dagger[O]$$

Case of $\mathrm{snd}[K]$: This case is proved similarly.
Case of $M@K$:

$$(M@K)_\circ\{O\} \equiv K_\circ\{OM_\circ\} \overset{I.H.}{\longrightarrow_v^*} K_\dagger\{OM_\dagger\} \overset{Lem\ 2.29}{\longrightarrow_v^*} K_\dagger[OM_\dagger] \equiv (M@K)_\dagger\{O\}$$

Case of $x.S$: $(x.S)_\circ\{O\} \equiv (\lambda x.S_\circ)O \overset{I.H.}{\longrightarrow_v^*} (\lambda x.S_\dagger)O \longrightarrow_v^* (\overline{\lambda}_v x.S_\dagger)O \equiv (x.S)_\dagger\{O\}$

Case of $M \bullet K$: $(M \bullet K)_\circ \equiv K_\circ\{M_\circ\} \overset{I.H.}{\longrightarrow_v^*} K_\dagger\{M_\dagger\} \overset{Lem\ 2.29}{\longrightarrow_v^*} K_\dagger[M_\dagger] \equiv (M \bullet K)_\dagger$ □

The translation $(-)_\dagger$ is compatible with the type system.

Proposition 2.31

(1) If $\Gamma \vdash_{dc} \Delta \mid M : A$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid M_\dagger : A$.

(2) If $K : A \mid \Gamma \vdash_{dc} \Delta$ and $\Gamma \vdash_{\lambda\mu} \Delta \mid O : A$, then $\Gamma \vdash_{\lambda\mu} \Delta \mid K_\dagger\{O\}$.

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

Proof. These claims can be shown by the subject reduction property for the $\lambda\mu$-calculus using Proposition 2.23 (1) and Lemma 2.30. □

Lemma 2.32

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 $\lambda\mu \vdash O \longrightarrow_v O'$, then $\lambda\mu \vdash K_\dagger\{O\} \longrightarrow_v^* K_\dagger\{O'\}$ and $\lambda\mu \vdash K_\dagger[O] \longrightarrow_v^* K_\dagger[O']$.

(2) $M_\dagger[{}^{V_\dagger}/_x] \equiv (M[{}^{V}/_x])_\dagger$, $(K_\dagger\{O\})[{}^{V_\dagger}/_x] \equiv (K[{}^{V}/_x])_\dagger\{O[{}^{V_\dagger}/_x]\}$,
$(K_\dagger[O])[{}^{V_\dagger}/_x] \equiv (K[{}^{V}/_x])_\dagger[\,O[{}^{V_\dagger}/_x]\,]$, and $S_\dagger[{}^{V_\dagger}/_x] \equiv (S[{}^{V}/_x])_\dagger$.

Proof. (1) The claim is proved by induction on K.

(2) We claim that if we have $(K_\dagger\{O\})[{}^{V_\dagger}/_x] \equiv (K[{}^{V}/_x])_\dagger\{O[{}^{V_\dagger}/_x]\}$, then we can derive $(K_\dagger[O])[{}^{V_\dagger}/_x] \equiv (K[{}^{V}/_x])_\dagger[\,O[{}^{V_\dagger}/_x]\,]$. We show this claim. Assume O is a value, then $O[{}^{V}/_x]$ is also a value. Hence we have

$$(K_\dagger[O])[{}^{V_\dagger}/_x] \equiv (K_\dagger\{O\})[{}^{V_\dagger}/_x] \equiv (K[{}^{V}/_x])_\dagger\{O[{}^{V_\dagger}/_x]\} \equiv (K[{}^{V}/_x])_\dagger[\,O[{}^{V_\dagger}/_x]\,].$$

If O is not a value, then $O[{}^{V}/_x]$ is also not a value, so we have

$$\begin{aligned}(K_\dagger[O])[{}^{V_\dagger}/_x] &\equiv ((\lambda z.K_\dagger\{z\})O)[{}^{V_\dagger}/_x] \equiv (\lambda z.(K_\dagger\{z\})[{}^{V_\dagger}/_x])(O[{}^{V_\dagger}/_x])) \\ &\equiv (\lambda z.(K[{}^{V}/_x])_\dagger\{z\})(O[{}^{V_\dagger}/_x])) \equiv (K[{}^{V}/_x])_\dagger[\,O[{}^{V_\dagger}/_x]\,].\end{aligned}$$

Therefore we show the other claims by induction on M, K, and S. We give the key case: $x_\dagger[{}^{V_\dagger}/_x] \equiv x[{}^{V_\dagger}/_x] \equiv V_\dagger$. □

Lemma 2.33
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

$$\begin{aligned}
&\lambda\mu \vdash M_\dagger[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}] \longrightarrow_v^* (M[{}^L/_\alpha])_\dagger,\\
&\lambda\mu \vdash (K_\dagger\{O\})[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}] \longrightarrow_v^* (K[{}^L/_\alpha])_\dagger\Big[O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]\Big],\\
&\lambda\mu \vdash (K_\dagger[O])[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}] \longrightarrow_v^* (K[{}^L/_\alpha])_\dagger\Big[\, O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]\,\Big], \text{ and}\\
&\lambda\mu \vdash S_\dagger[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}] \longrightarrow_v^* (S[{}^L/_\alpha])_\dagger.
\end{aligned}$$

Proof. We can prove this lemma by a simultaneous induction on M, K, and S. We give two cases.
Case of α:

$$\begin{aligned}
(\alpha_\dagger\{O\})[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}] &\equiv ([\alpha]O)[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]\\
&\equiv (\lambda y.L_\dagger\{y\})(O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}])\\
&\longrightarrow_v^* (\overline{\lambda}_v y.L_\dagger\{y\})(O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]) \equiv L_\dagger[O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]]
\end{aligned}$$

Case of $\mathrm{fst}[K]$:

$$\begin{aligned}
((\mathrm{fst}[K])_\dagger\{O\})[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}] &\equiv (K_\dagger[\mathrm{fst}(O)])[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]\\
&\overset{I.H.}{\longrightarrow_v^*} (K[{}^L/_\alpha])_\dagger\Big[\mathrm{fst}(O)[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}]\Big]\\
&\equiv (K[{}^L/_\alpha])_\dagger\Big[\mathrm{fst}(O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}])\Big]\\
&\equiv (\mathrm{fst}(K[{}^L/_\alpha]))_\dagger\Big\{\mathrm{fst}(O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}])\Big\}\\
&\overset{Lem\ 2.29(2)}{\longrightarrow_v^*} (\mathrm{fst}(K[{}^L/_\alpha]))_\dagger\Big[\mathrm{fst}(O[{}^{(\lambda y.L_\dagger\{y\})\{-\}}/_{[\alpha]\{-\}}])\Big]
\end{aligned}$$

□

We now prove that the call-by-name translation $(-)_\dagger$ preserves reductions.

Theorem 2.34 (Soundness of $(-)_\dagger$)
(1) If $DC \vdash M \longrightarrow^v N$, then $\lambda\mu \vdash M_\dagger \longrightarrow_v^* N_\dagger$.
(2) If $DC \vdash K \longrightarrow^v L$, then $\lambda\mu \vdash K_\dagger\{O\} \longrightarrow_v^* L_\dagger\{O\}$ and $\lambda\mu \vdash K_\dagger[O] \longrightarrow_v^* L_\dagger[O]$.
(3) If $DC \vdash S \longrightarrow^v T$, then $\lambda\mu \vdash S_\dagger \longrightarrow_v^* T_\dagger$.
Moreover, in (1), (2), and (3), if $\longrightarrow^v$ is $(\beta_\supset)$, $(\beta_\wedge)$, $(\beta_\vee)$ or $(\beta_\neg)$, then $\longrightarrow_v^*$ can be replaced by $\longrightarrow_v^+$.

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

We often use the following shortcuts:

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

(b) $(\lambda x.K_\dagger\{x\})M_\dagger \longrightarrow_v^* (\overline{\lambda}_v x.K_\dagger\{x\})M_\dagger \equiv K_\dagger[M_\dagger] \equiv (M \bullet K)_\dagger$

(c) $(\lambda x.S_\dagger)M_\dagger \longrightarrow_v^* (\overline{\lambda}_v x.S_\dagger)M_\dagger \equiv (x.S)_\dagger\{M_\dagger\} \overset{\mathit{Lem}\ 2.29}{\longrightarrow_v^*} (x.S)_\dagger[M_\dagger] \equiv (M \bullet x.S)_\dagger$

Base cases are shown as follows.

Case of $(\beta_\supset)$:

$$\begin{aligned}((\lambda x.M) \bullet (N@K))_\dagger &\overset{(a)}{\equiv} (N@K)_\dagger\{(\lambda x.M)_\dagger\} \equiv K_\dagger[(\lambda x.M_\dagger)N_\dagger]\\ &\equiv (\lambda z.K_\dagger\{z\})((\lambda x.M_\dagger)N_\dagger) \longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})M_\dagger)N_\dagger\\ &\overset{(b)}{\longrightarrow_v^*} (\lambda x.(M \bullet K)_\dagger)N_\dagger \overset{(c)}{\longrightarrow_v^*} (N \bullet x.(M \bullet K))_\dagger\end{aligned}$$

Case of $(\beta_\wedge)$:

$$(\langle V, W\rangle \bullet \mathrm{fst}[K])_\dagger \overset{(a)}{\equiv} \mathrm{fst}[K]_\dagger\{(\langle V, W\rangle)_\dagger\} \equiv K_\dagger[\,\mathrm{fst}\langle V_\dagger, W_\dagger\rangle\,] \longrightarrow_{(\beta_\wedge)} K_\dagger[V_\dagger] \equiv (V \bullet K)_\dagger$$

The other case of $(\beta_\wedge)$ can be shown similarly.

Case of $(\beta_\vee)$:

$$\begin{aligned}(\langle V\rangle\mathrm{inl} \bullet [K, L])_\dagger &\overset{(a)}{\equiv} [K, L]_\dagger\{\,\langle V\rangle\mathrm{inl}_\dagger\,\} \equiv \delta(\mathrm{inl}(V_\dagger), x.K_\dagger\{x\}, y.L_\dagger\{y\})\\ &\longrightarrow_{(\beta_\vee)} K_\dagger\{V_\dagger\} \overset{(a)}{\equiv} (V \bullet K)_\dagger\end{aligned}$$

The other case of $(\beta_\wedge)$ can be shown similarly.

Case of $(\beta_\neg)$:

$$([K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle)_\dagger \overset{(a)}{\equiv} (\mathrm{not}\langle M\rangle)_\dagger\{\,[K]\mathrm{not}_\dagger\,\} \equiv (\lambda x.K_\dagger\{x\})M_\dagger \overset{(b)}{\longrightarrow_v^*} (M \bullet K)_\dagger$$

Case of (β_R):

$$\begin{aligned}(S.\alpha \bullet K)_\dagger &\equiv K_\dagger[\,(S.\alpha)_\dagger\,] \equiv (\lambda x.K_\dagger\{x\})\mu\alpha.S_\dagger\\ &\longrightarrow_{(\zeta)} S_\dagger[{}^{(\lambda x.K_\dagger\{x\})\{-\}}/_{[\alpha](-)}] \overset{\mathit{Lem}\ 2.33}{\equiv} (S[{}^{K}/_{\alpha}])_\dagger\end{aligned}$$

Case of (β_L):

$$(V \bullet x.S)_\dagger \overset{(a)}{\equiv} (x.S)_\dagger\{V_\dagger\} \equiv S_\dagger[{}^{V_\dagger}/_{x}] \overset{\mathit{Lem}\ 2.32(2)}{\equiv} (S[{}^{V}/_{x}])_\dagger$$

Case of (η_R):

$$M_\dagger \longrightarrow_{(\eta_\mu)} \mu\alpha.[\alpha]M_\dagger \equiv \mu\alpha.\alpha_\dagger\{M\} \overset{Lem\ 2.29(2)}{\longrightarrow_v^*} \mu\alpha.\alpha_\dagger[M]$$
$$\equiv \mu\alpha.(M \bullet \alpha)_\dagger \equiv (\,(M \bullet \alpha).\alpha\,)_\dagger$$

Case of (η_L): If O is a value V, then

$$K_\dagger\{V\} \equiv (K_\dagger\{x\})[^V/_x] \overset{(a)}{\equiv} (x \bullet K)_\dagger[^V/_x] \equiv (x.(x \bullet K))_\dagger\{V\}\,.$$

If O is not a value, then

$$K_\dagger\{O\} \overset{Lem\ 2.29(2)}{\longrightarrow_v^*} K_\dagger[O] \equiv (\lambda x.K_\dagger\{x\})O \overset{(a)}{\equiv} (\lambda x.(x \bullet K)_\dagger)O \equiv (x.(x \bullet K))_\dagger\{O\}\,.$$

Case of (*name*): Note that $M_\dagger$ is not a value because M is not a value.

$$(\,\langle M, N\rangle \bullet K\,)_\dagger \equiv K_\dagger[\,\langle M, N\rangle_\dagger\,] \equiv (\lambda z.K_\dagger\{z\})\langle M_\dagger, N_\dagger\rangle$$
$$\longrightarrow_{(name)} (\lambda z.K_\dagger\{z\})((\lambda x.\langle x, N_\dagger\rangle)M_\dagger) \longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})\langle x, N_\dagger\rangle)M_\dagger$$
$$\overset{(b)}{\longrightarrow_v^*} (\lambda x.(\langle x, N\rangle \bullet K)_\dagger)M_\dagger \overset{(c)}{\longrightarrow_v^*} (M \bullet x.(\langle x, N\rangle \bullet K))_\dagger$$

$$(\,\langle V, M\rangle \bullet K\,)_\dagger \equiv K_\dagger[\,\langle V, M\rangle_\dagger\,] \equiv (\lambda z.K_\dagger\{z\})\langle V_\dagger, M_\dagger\rangle$$
$$\longrightarrow_{(name)} (\lambda z.K_\dagger\{z\})((\lambda x.\langle V_\dagger, x\rangle)M_\dagger) \longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})\langle V_\dagger, x\rangle)M_\dagger$$
$$\overset{(b)}{\longrightarrow_v^*} (\lambda x.(\langle V, x\rangle \bullet K)_\dagger)M_\dagger \overset{(c)}{\longrightarrow_v^*} (M \bullet x.(\langle V, x\rangle \bullet K))_\dagger$$

$$(\,\langle M\rangle\mathrm{inl} \bullet K\,)_\dagger \equiv K_\dagger[\,\langle M\rangle\mathrm{inl}_\dagger\,] \equiv (\lambda z.K_\dagger\{z\})\mathrm{inl}(M_\dagger)$$
$$\longrightarrow_{(name)} (\lambda z.K_\dagger\{z\})((\lambda x.\mathrm{inl}(x))M_\dagger) \longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})\mathrm{inl}(x))M_\dagger$$
$$\overset{(b)}{\longrightarrow_v^*} (\lambda x.(\langle x\rangle\mathrm{inl} \bullet K)_\dagger)M_\dagger \overset{(c)}{\longrightarrow_v^*} (M \bullet x.(\langle x\rangle\mathrm{inl} \bullet K))_\dagger$$

The last case, $\langle M\rangle\mathrm{inr} \bullet K \longrightarrow^v M \bullet x.(\langle x\rangle\mathrm{inr} \bullet K)$, is also shown similarly.

Induction cases can be easily shown. □

2.5.4 Reloading property

Wadler (2005) showed that the compositions of his translations $\lambda\mu \to dual \to \lambda\mu$ and $dual \to \lambda\mu \to dual$ 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 obtain corresponding results as follows:

- a reloaded term by the call-by-name modified translations is reduced from the original term by the call-by-name reductions (Proposition 2.35 (1), 2.36 (1)), and

- a reloaded term by the call-by-value modified translations is reduced from the original term by the call-by-value reductions (Proposition 2.35 (2), 2.36 (2)).

Proposition 2.35 (Reloading property: $\lambda\mu \to dual \to \lambda\mu$)
Let O be a term and S be a statement of the $\lambda\mu$-calculus. Then

(1) $\lambda\mu \vdash S \longrightarrow_n^* (S^\sharp)_\sharp$,
$\lambda\mu \vdash P_\sharp\{O\} \longrightarrow_n^* (O :_n P)_\sharp$ for any covalue P, especially $\lambda\mu \vdash O \longrightarrow_n^* (O^\sharp)_\sharp$.

(2) $\lambda\mu \vdash S \longrightarrow_v^* (S^\dagger)_\dagger$,
$\lambda\mu \vdash K_\dagger[O] \longrightarrow_v^* (O :_v K)_\dagger$ for any coterm K, especially $\lambda\mu \vdash O \longrightarrow_v^* (O^\dagger)_\dagger$.

Proof. (1) If we have $P_\sharp\{O\} \longrightarrow_n^* (O :_n P)_\sharp$ for any covalue P, then we can obtain $O \longrightarrow_{(\eta_\mu)} \mu\alpha.[\alpha]O \equiv \mu\alpha.\alpha_\sharp\{O\} \longrightarrow_n^* \mu\alpha.(O :_n \alpha)_\sharp \equiv ((O :_n \alpha).\alpha)_\sharp \equiv (O^\sharp)_\sharp$. We prove the rest of (1) by a simultaneous induction on O and S.
Case of x: $P_\sharp\{x\} \equiv (x \bullet P)_\sharp \equiv (x :_n P)_\sharp$
Case of $\langle M, N\rangle$: $P_\sharp\{\langle M, N\rangle\} \overset{I.H.}{\longrightarrow_n^*} P_\sharp\{\langle (M^\sharp)_\sharp, (N^\sharp)_\sharp\rangle\} \equiv P_\sharp\{\langle M^\sharp, N^\sharp\rangle_\sharp\}$
$\equiv (\langle M, N\rangle^\sharp \bullet P)_\sharp \longrightarrow_n^* (\langle M, N\rangle :_n P)_\sharp$
Case of $\mathrm{fst}(O)$: $P_\sharp\{\mathrm{fst}(O)\} \equiv \mathrm{fst}[P]_\sharp\{O\} \overset{I.H.}{\longrightarrow_n^*} \mathrm{fst}[P]_\sharp\{(O^\sharp)_\sharp\} \equiv (O^\sharp \bullet \mathrm{fst}[P])_\sharp$
$\longrightarrow_n^* (O :_n \mathrm{fst}[P])_\sharp \equiv (\mathrm{fst}(O) :_n P)_\sharp$
The case of $\mathrm{snd}(O)$ is shown similarly.
Case of $\mathrm{inl}(O)$:

$$P_\sharp\{\mathrm{inl}(O)\} \overset{I.H.}{\longrightarrow_n^*} P_\sharp\{\mathrm{inl}((O^\sharp)_\sharp)\} \equiv P_\sharp\{\langle O^\sharp\rangle\mathrm{inl}_\sharp\} \equiv (\langle O^\sharp\rangle\mathrm{inl} \bullet P)_\sharp \equiv (\mathrm{inl}(O) :_n P)_\sharp$$

The case of $\mathrm{inr}(O)$ is shown similarly.
Case of $\delta(O, x.M, y.N)$:

$$\begin{aligned}
&P_\sharp\{\delta(O, x.M, y.N)\} \longrightarrow_{(\pi)} \delta(O, x.P_\sharp\{M\}, y.P_\sharp\{N\}) \\
&\quad \overset{I.H.}{\longrightarrow_n^*} \delta\big(O, x.(M :_n P), y.(N :_n P)\big) \equiv \delta\big(O, x'.(x.(M :_n P))_\sharp\{x'\}, y'.(y.(N :_n P))_\sharp\{y'\}\big) \\
&\quad \equiv [x.(M :_n P), y.(N :_n P)]_\sharp\{O\} \overset{I.H.}{\longrightarrow_n^*} [x.(M :_n P), y.(N :_n P)]_\sharp\{(O^\sharp)_\sharp\} \\
&\quad \equiv \big(O^\sharp \bullet [x.(M :_n P), y.(N :_n P)]\big)_\sharp \equiv (\delta(O, x.M, y.N) :_n P)_\sharp
\end{aligned}$$

Case of $\lambda x.M$: $P_\sharp\{\lambda x.M\} \overset{I.H.}{\longrightarrow_n^*} P_\sharp\{\lambda x.(M^\sharp)_\sharp\} \equiv ((\lambda x.M^\sharp) \bullet P)_\sharp \equiv (\lambda x.M :_n P)_\sharp$
Case of MN (MN is a term, M is not a λ-abstraction):

$$P_\sharp\{\, MN\,\} \overset{I.H.}{\longrightarrow_n^*} P_\sharp\{\, M(N^\sharp)_\sharp\,\} \equiv (N^\sharp @P)_\sharp\{\, M\,\} \overset{I.H.}{\longrightarrow_n^*} (M^\sharp :_n (N^\sharp @P))_\sharp \equiv (MN :_n P)_\sharp$$

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

$$P_\sharp\{\,(\lambda x.M)N\,\} \longrightarrow_{(\beta_\supset)} P_\sharp\{\, M[{}^N/_x]\,\} \equiv (x.(P_\sharp\{M\}))_\sharp\{N\} \overset{I.H.}{\longrightarrow_n^*} (x.(M :_n P))_\sharp\{N\}$$
$$\overset{I.H.}{\longrightarrow_n^*} (x.(M :_n P))_\sharp\{\,(N^\sharp)_\sharp\,\} \equiv (N^\sharp \bullet x.(M :_n P))_\sharp \equiv (\,(\lambda x.M)N :_n P\,)_\sharp$$

Case of $\lambda x.S$:

$$P_\sharp\{\lambda x.S\,\} \overset{I.H.}{\longrightarrow_n^*} P_\sharp\{\lambda x.(S^\sharp)_\sharp\} \equiv P_\sharp\big\{\lambda y.(x.S^\sharp)_\sharp\{y\}\big\} \equiv \big([x.S^\sharp]\text{not} \bullet P\big)_\sharp \equiv (\,\lambda x.S \;:_n P\,)_\sharp$$

Case of $\mu\alpha.S$:

$$P_\sharp\{\mu\alpha.S\,\} \overset{I.H.}{\longrightarrow_n^*} P_\sharp\{\mu\alpha.(S^\sharp)_\sharp\} \longrightarrow_{(\zeta)} (S^\sharp)_\sharp[{}^{P_\sharp\{-\}}/_{[\alpha]\{-\}}] \longrightarrow_n^* (S^\sharp[{}^P/_\alpha])_\sharp \equiv (\,\mu\alpha.S \;:_n P\,)_\sharp$$

Case of $[\alpha]O$: $[\alpha]O \equiv \alpha_\sharp\{O\} \overset{I.H.}{\longrightarrow_n^*} (O :_n \alpha)_\sharp \equiv (([\alpha]O)^\sharp)_\sharp$
Case of $\delta(O, x.S, y.T)$:

$$\delta(O, x.S, y.T) \overset{I.H.}{\longrightarrow_n^*} \delta\big(O, x.(S^\sharp)_\sharp, y.(T^\sharp)_\sharp\big) \equiv \delta\big(O, x'.((x.S^\sharp)_\sharp\{x'\}), y'.((y.T^\sharp)_\sharp\{y'\})\big)$$
$$\equiv [x.S^\sharp, y.T^\sharp]_\sharp\{O\} \overset{I.H.}{\longrightarrow_n^*} [x.S^\sharp, y.T^\sharp]_\sharp\{(O^\sharp)_\sharp\} \equiv \big(O^\sharp \bullet [x.S^\sharp, y.T^\sharp]\big)_\sharp \equiv (\delta(O, x.S, y.T)^\sharp)_\sharp$$

Case of MN (MN is a statement, M is not a λ-abstraction):

$$MN \overset{I.H.}{\longrightarrow_n^*} M(N^\sharp)_\sharp \equiv \text{not}\langle N^\sharp\rangle_\sharp\{M\} \overset{I.H.}{\longrightarrow_n^*} (M :_n \text{not}\langle N^\sharp\rangle)_\sharp \equiv (\,(MN)^\sharp\,)_\sharp$$

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

$$(\lambda x.S\,)N \overset{I.H.}{\longrightarrow_n^*} (\lambda x.(S^\sharp)_\sharp)N \longrightarrow_{(\beta_\neg)} (S^\sharp)_\sharp[{}^N/_x] \equiv (x.S^\sharp)_\sharp\{N\}$$
$$\overset{I.H.}{\longrightarrow_n^*} (x.S^\sharp)_\sharp\{(N^\sharp)_\sharp\} \equiv \big(N^\sharp \bullet x.S^\sharp\big)_\sharp \equiv \big(((\lambda x.S\,)N)^\sharp\big)_\sharp$$

(2) We can easily show $O \longrightarrow_v^* (O^\dagger)_\dagger$ from $K_\dagger[O] \longrightarrow_v^* (O :_v K)_\dagger$ 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: $K_\dagger[x] \equiv (x \bullet K)_\dagger \equiv (x :_v K)_\dagger$

Case of $\langle M, N\rangle$:

$$\begin{aligned} K_\dagger[\langle M, N\rangle] &\overset{I.H.}{\longrightarrow_v^*} K_\dagger[\langle (M^\dagger)_\dagger, (N^\dagger)_\dagger\rangle] \equiv K_\dagger[\,\langle M^\dagger, N^\dagger\rangle_\dagger\,] \equiv (\langle M, N\rangle^\dagger \bullet K)_\dagger \\ &\longrightarrow_v^* (\langle M, N\rangle :_v K)_\dagger \end{aligned}$$

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

$$\begin{aligned} K_\dagger[\mathrm{fst}(O)] &\equiv \mathrm{fst}[K]_\dagger\{O\} \overset{I.H.}{\longrightarrow_v^*} \mathrm{fst}[K]_\dagger\{(O^\dagger)_\dagger\} \overset{Lem\ 2.29}{\longrightarrow_v^*} \mathrm{fst}[K]_\dagger[(O^\dagger)_\dagger] \equiv (O^\dagger \bullet \mathrm{fst}[K])_\dagger \\ &\longrightarrow_v^* (O :_v \mathrm{fst}[K])_\dagger \equiv (\mathrm{fst}(O) :_v K)_\dagger \end{aligned}$$

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

Case of $\mathrm{inl}(O)$:

$$K_\dagger[\mathrm{inl}(O)] \overset{I.H.}{\longrightarrow_v^*} K_\dagger[\mathrm{inl}((O^\dagger)_\dagger)] \equiv K_\dagger[\langle O^\dagger\rangle \mathrm{inl}_\dagger] \equiv (\langle O^\dagger\rangle \mathrm{inl} \bullet K)_\dagger \equiv (\mathrm{inl}(O) :_v K)_\dagger$$

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

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

$$\begin{aligned} K_\dagger[\delta(O, x.M, y.N)] &\equiv (\lambda z.K_\dagger\{z\})\delta(O, x.M, y.N) \longrightarrow_{(\pi)} \delta\big(O, x.(\lambda z.K_\dagger\{z\})M, y.(\lambda z.K_\dagger\{z\})N\big) \\ &\longrightarrow_v^* \delta\big(O, x.K_\dagger[M], y.K_\dagger[N]\big) \overset{I.H.}{\longrightarrow_v^*} \delta\big(O, x.(M :_v K), y.(N :_v K)\big) \\ &\equiv \delta\big(O, x'.(x.(M :_v K))_\dagger\{x'\}, y'.(y.(N :_v K))_\dagger\{y'\}\big) \equiv [x.(M :_v K), y.(N :_v K)]_\dagger\{O\} \\ &\overset{Lem\ 2.29}{\longrightarrow_v^*} [x.(M :_v K), y.(N :_v K)]_\dagger[O] \overset{I.H.}{\longrightarrow_v^*} \big(O :_v [x.(M :_v K), y.(N :_v K)]\big)_\dagger \\ &\equiv (\,\delta(O, x.M, y.N) :_v K\,)_\dagger \end{aligned}$$

Case of $\lambda x.M$: $K_\dagger[\lambda x.M] \overset{I.H.}{\longrightarrow_v^*} K_\dagger[\lambda x.(M^\dagger)_\dagger] \equiv ((\lambda x.M^\dagger) \bullet K)_\dagger \equiv (\lambda x.M :_v K)_\dagger$

Case of MN (MN is a term, M is not a λ-abstraction):

$$K_\dagger[\,MN\,] \overset{I.H.}{\longrightarrow_v^*} K_\dagger[\,M(N^\dagger)_\dagger\,] \equiv (N^\dagger @ K)_\dagger\{\,M\,\} \overset{Lem\ 2.29}{\longrightarrow_v^*} (N^\dagger @ K)_\dagger[\,M\,]$$

$$\overset{I.H.}{\longrightarrow}{}^*_v (M :_v (N^\dagger @K))_\dagger \equiv (MN :_v K)_\dagger$$

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

$$\begin{aligned}K_\dagger[\,(\lambda x.M)N\,] &\equiv (\lambda z.K_\dagger\{z\})((\lambda x.M)N) \longrightarrow_{(comp)} (\lambda x.(\lambda z.K_\dagger\{z\})M)N \\ &\longrightarrow^*_v (\lambda x.K_\dagger[M])N \overset{I.H.}{\longrightarrow}{}^*_v (\lambda x.(M :_v K)_\dagger)N \longrightarrow^*_v (x.(M :_v K))_\dagger\{N\} \\ &\overset{I.H.}{\longrightarrow}{}^*_v (N :_v x.(M :_v K))_\dagger \equiv (\,(\lambda x.M)N :_v K\,)_\dagger\end{aligned}$$

Case of $\lambda x.S$:

$$K_\dagger[\lambda x.S\,] \overset{I.H.}{\longrightarrow}{}^*_v K_\dagger[\lambda x.(S^\dagger)_\dagger] \equiv K_\dagger\Big[\lambda y.(x.S^\dagger)_\dagger\{y\}\Big] \equiv \Big([x.S^\dagger]\text{not} \bullet K\Big)_\dagger \equiv (\,\lambda x.S\ :_v K\,)_\dagger$$

Case of $\mu\alpha.S$:

$$\begin{aligned}K_\dagger[\mu\alpha.S\,] &\overset{I.H.}{\longrightarrow}{}^*_v K_\dagger[\mu\alpha.(S^\dagger)_\dagger] \equiv (\lambda z.K_\dagger\{z\})\mu\alpha.(S^\dagger)_\dagger \longrightarrow_{(\zeta)} (S^\dagger)_\dagger[{}^{(\lambda z.K_\dagger\{z\})\{-\}}/_{[\alpha]\{-\}}] \\ &\overset{Lem\ 2.33}{\longrightarrow}{}^*_v\ (S^\dagger[{}^K/_\alpha])_\dagger \equiv (\,\mu\alpha.S\ :_v K\,)_\dagger\end{aligned}$$

Case of $[\alpha]O$: $[\alpha]O \equiv \alpha_\dagger\{O\} \overset{Lem\ 2.29}{\longrightarrow}{}^*_v\ \alpha_\dagger[O] \overset{I.H.}{\longrightarrow}{}^*_v (O :_v \alpha)_\dagger \equiv (([\alpha]O)^\dagger)_\dagger$

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

$$\begin{aligned}\delta(O, x.S, y.T) &\overset{I.H.}{\longrightarrow}{}^*_v \delta\big(O, x.(S^\dagger)_\dagger, y.(T^\dagger)_\dagger\big) \equiv \delta\big(O, x'.((x.S^\dagger)_\dagger\{x'\}), y'.((y.T^\dagger)_\dagger\{y'\})\big) \\ &\equiv [x.S^\dagger, y.T^\dagger]_\dagger\{O\} \overset{Lem\ 2.29}{\longrightarrow}{}^*_v\ [x.S^\dagger, y.T^\dagger]_\dagger[O] \overset{I.H.}{\longrightarrow}{}^*_v \big(O :_v [x.S^\dagger, y.T^\dagger]\big)_\dagger \equiv (\delta(O, x.S, y.T)^\dagger)_\dagger\end{aligned}$$

Case of MN (MN is a statement, M is not a λ-abstraction) :

$$MN \overset{I.H.}{\longrightarrow}{}^*_v M(N^\dagger)_\dagger \equiv \text{not}\langle N^\dagger\rangle_\dagger\{M\} \overset{I.H.}{\longrightarrow}{}^*_v (M :_v \text{not}\langle N^\dagger\rangle)_\dagger \equiv (\,(MN)^\dagger\,)_\dagger$$

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

$$(\lambda x.S\,)N \overset{I.H.}{\longrightarrow}{}^*_v (\lambda x.(S^\dagger)_\dagger)N \longrightarrow^*_v (x.S^\dagger)_\dagger[N] \overset{I.H.}{\longrightarrow}{}^*_v (N :_v x.S^\dagger)_\dagger \equiv \Big(((\lambda x.S\,)N)^\dagger\Big)_\dagger$$

□

Proposition 2.36 (Reloading property: *dual* $\to \lambda\mu \to$ *dual*)
Let M be a term, K be a coterm, and S be a statement of the dual calculus. Then

(1) $DC \vdash M \longrightarrow^{n*} (M_\sharp)^\sharp$,
$DC \vdash O^\sharp \bullet K \longrightarrow^{n*} (K_\sharp\{O\})^\sharp$ for any term O of the $\lambda\mu$-calculus, and
$DC \vdash S \longrightarrow^{n*} (S_\sharp)^\sharp$;

(2) $DC \vdash M \longrightarrow^{v*} (M_\dagger)^\dagger$,
$DC \vdash O^\dagger \bullet K \longrightarrow^{v*} (K_\dagger\{O\})^\dagger$ for any term O of the $\lambda\mu$-calculus, and
$DC \vdash S \longrightarrow^{v*} (S_\dagger)^\dagger$.

Proof. (1) If we establish the following claims: (a) $M \bullet Q \longrightarrow^{n*} (M_\sharp :_n Q)$; (b) $O^\sharp \bullet K \longrightarrow^{n*} K_\sharp\{O\}$ if K is not a covalue; (c) $(O :_n P) \longrightarrow^{n*} (P_\sharp\{O\})^\sharp$; and (d) $S \longrightarrow^{n*} (S_\sharp)^\sharp$, 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 $\langle M\rangle$inl: $\langle M\rangle\mathrm{inl} \bullet Q \overset{I.H.(a)}{\longrightarrow^{n*}} \langle (M_\sharp)^\sharp\rangle\mathrm{inl} \bullet Q \equiv \mathrm{inl}(M_\sharp) :_n Q \equiv \langle M\rangle\mathrm{inl}_\sharp :_n Q$
The case of $\langle M\rangle$inr is shown similarly.
Case of $\langle M, N\rangle$: $\langle M, N\rangle \bullet Q \overset{I.H.(a)}{\longrightarrow^{n*}} \langle (M_\sharp)^\sharp, (N_\sharp)^\sharp\rangle \bullet Q \equiv (\langle M_\sharp, N_\sharp\rangle :_n Q) \equiv (\langle M, N\rangle_\sharp :_n Q)$
Case of $[K]$not:

$$[K]\mathrm{not} \bullet Q \longrightarrow^{n}_{(\eta_L)} [x.(x \bullet K)]\mathrm{not} \bullet Q \longrightarrow^{n}_{(\eta_R)} [x.(x^\sharp \bullet K)]\mathrm{not} \bullet Q$$
$$\overset{I.H.(b)}{\longrightarrow^{n*}} [x.(K_\sharp\{x\})^\sharp]\mathrm{not} \bullet Q \equiv (\lambda x.(K_\sharp\{x\}) :_n Q) \equiv ([K]\mathrm{not}_\sharp :_n Q)$$

Case of $\lambda x.M$: $(\lambda x.M) \bullet P \overset{I.H.(a)}{\longrightarrow^{n*}} (\lambda x.(M_\sharp)^\sharp) \bullet P \equiv (\lambda x.M_\sharp :_n P) \equiv ((\lambda x.M)_\sharp :_n P)$
Case of $S.\alpha$:

$$S.\alpha \bullet Q \overset{I.H.(d)}{\longrightarrow^{n*}} (S_\sharp)^\sharp.\alpha \bullet Q \longrightarrow^{n}_{(\beta_R)} (S_\sharp)^\sharp[{}^{Q}/_{\alpha}] \equiv (\mu\alpha.S_\sharp :_n Q) \equiv ((S.\alpha)_\sharp :_n Q)$$

Case of α: $(O :_n \alpha) \equiv ([\alpha]O)^\sharp \equiv (\alpha_\sharp\{O\})^\sharp$
Case of $[P, Q]$:

$$(O :_n [P, Q]) \longrightarrow^{n*}_{(\eta_L)} (O :_n [x.(x \bullet P), y.(y \bullet Q)]) \overset{I.H.(c)}{\longrightarrow^{n*}} (O :_n [x.(P_\sharp\{x\})^\sharp, y.(Q_\sharp\{y\})^\sharp])$$
$$\longrightarrow^{n*} O^\sharp \bullet [x.(P_\sharp\{x\})^\sharp, y.(Q_\sharp\{y\})^\sharp] \equiv \delta(O, x.P_\sharp\{x\}, y.Q_\sharp\{y\})^\sharp \equiv ([P, Q]_\sharp\{O\})^\sharp$$

Case of fst$[P]$: $(O :_n \text{fst}[P]) \equiv (\text{fst}(O) :_n P) \overset{I.H.(c)}{\longrightarrow^{n*}} (P_\sharp\{\text{fst}(O)\})^\sharp \equiv (\text{fst}[P]_\sharp\{O\})^\sharp$
The case of snd$[P]$ can be shown similarly.
Case of not$\langle M\rangle$:

$$(O :_n \text{not}\langle M\rangle) \overset{I.H.(a)}{\longrightarrow^{n*}} (O :_n \text{not}\langle (M_\sharp)^\sharp\rangle) \longrightarrow^{n*} (OM_\sharp)^\sharp \equiv (\text{not}\langle M\rangle_\sharp\{O\})^\sharp$$

Case of $M@P$:

$$\begin{aligned}(O :_n (M@P)) &\overset{I.H.(a)}{\longrightarrow^{n*}} \big(O :_n ((M_\sharp)^\sharp @P)\big) \longrightarrow^{n*} (OM_\sharp :_n P) \overset{I.H.(c)}{\longrightarrow^{n*}} (P_\sharp\{OM_\sharp\})^\sharp \\ &\equiv (\,(M@P)_\sharp\{O\}\,)^\sharp\end{aligned}$$

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

$$\begin{aligned}(O^\sharp \bullet [K, L]) &\longrightarrow^{n*}_{(\eta_L)} O^\sharp \bullet [x.(x \bullet K), y.(y \bullet L)] \overset{I.H.(b)}{\longrightarrow^{n*}} O^\sharp \bullet [x.(K_\sharp\{x\})^\sharp, y.(L_\sharp\{y\})^\sharp] \\ &\equiv \delta(O, x.K_\sharp\{x\}, y.L_\sharp\{y\}) \equiv ([K, L]_\sharp\{O\})^\sharp\end{aligned}$$

Case of fst$[K]$ (where K is not a covalue):

$$\begin{aligned}(O^\sharp \bullet \text{fst}[K]) &\longrightarrow^{n}_{(name)} (O^\sharp \bullet \text{fst}[\alpha]).\alpha \bullet K \longrightarrow^{n*} (O :_n \text{fst}[\alpha]).\alpha \bullet K \\ &\equiv (\text{fst}(O) :_n \alpha).\alpha \bullet K \equiv (\text{fst}(O))^\sharp \bullet K \\ &\overset{I.H.(b)}{\longrightarrow^{n*}} \big(K_\sharp\{\text{fst}(O)\}\big)^\sharp \equiv \big(\text{fst}[K]_\sharp\{O\}\big)^\sharp\end{aligned}$$

The case of snd$[K]$ can be shown similarly.
Case of $M@K$ (where K is not a covalue):

$$\begin{aligned}(O^\sharp \bullet (M@K)) &\longrightarrow^{n}_{(name)} (O^\sharp \bullet (M@\alpha)).\alpha \bullet K \longrightarrow^{n*} (\,O :_n (M@\alpha)\,).\alpha \bullet K \\ &\overset{I.H.(a)}{\longrightarrow^{n*}} \big(\,O :_n ((M_\sharp)^\sharp @\alpha)\,\big).\alpha \bullet K \longrightarrow^{n*} (\,OM_\sharp :_n \alpha\,).\alpha \bullet K \\ &\equiv (OM_\sharp)^\sharp \bullet K \overset{I.H.(b)}{\longrightarrow^{n*}} (\,K_\sharp\{OM_\sharp\}\,)^\sharp \equiv (\,(M@K)_\sharp\{O\}\,)^\sharp\end{aligned}$$

Case of $x.S$:

$$(O^\sharp \bullet x.S) \overset{I.H.(d)}{\longrightarrow^{n*}} (O^\sharp \bullet x.(S_\sharp)^\sharp) \longrightarrow^{n}_{(\beta_L)} (S_\sharp)^\sharp[{}^{O^\sharp}/_x] \overset{Lem\ 2.14(1)}{\longrightarrow^{n*}} (S_\sharp[{}^{O}/_x])^\sharp \equiv ((x.S)_\sharp\{O\})^\sharp$$

Case of $M \bullet K$: $(M \bullet K) \overset{I.H.(a)}{\longrightarrow^{n*}} (M_\sharp)^\sharp \bullet K \overset{I.H.}{\longrightarrow^{n*}} (K_\sharp\{M_\sharp\})^\sharp \equiv ((M \bullet K)_\sharp)^\sharp$

(2) We show these claims by a simultaneous induction on M, K, and S. If we establish $(O :_v K) \longrightarrow^{v*} (K_\dagger\{O\})^\dagger$, then we can easily obtain $(O^\dagger \bullet K) \longrightarrow^{v*} (K_\dagger\{O\})^\dagger$. Therefore, we show this claim instead of the second clause of (2).

Case of x: $x \longrightarrow^{v}_{(\eta_R)} (x \bullet \alpha).\alpha \equiv (x :_v \alpha).\alpha \equiv x^\dagger \equiv (x_\dagger)^\dagger$

Case of $\langle M\rangle$inl:

$$\begin{aligned}\langle M\rangle\mathrm{inl} &\overset{I.H.}{\longrightarrow^{v*}} \langle (M_\dagger)^\dagger\rangle\mathrm{inl} \longrightarrow^{v}_{(\eta_R)} (\langle (M_\dagger)^\dagger\rangle\mathrm{inl} \bullet \alpha).\alpha \equiv (\mathrm{inl}(M_\dagger) :_v \alpha).\alpha \\ &\equiv (\mathrm{inl}(M_\dagger))^\dagger \equiv (\langle M\rangle\mathrm{inl}_\dagger)^\dagger\end{aligned}$$

$\langle M\rangle$inr is shown similarly.

Case of $\langle M, N\rangle$:

$$\begin{aligned}\langle M, N\rangle &\overset{I.H.}{\longrightarrow^{v*}} \langle (M_\dagger)^\dagger, (N_\dagger)^\dagger\rangle \longrightarrow^{v}_{(\eta_R)} (\langle (M_\dagger)^\dagger, (N_\dagger)^\dagger\rangle \bullet \alpha).\alpha \\ &\equiv (\langle M_\dagger, N_\dagger\rangle :_v \alpha).\alpha \equiv \langle M_\dagger, N_\dagger\rangle^\dagger \equiv (\langle M, N\rangle_\dagger)^\dagger\end{aligned}$$

Case of $[K]$not:

$$\begin{aligned}[K]\mathrm{not} &\longrightarrow^{v}_{(\eta_L)} [x.(x \bullet K)]\mathrm{not} \overset{I.H.}{\longrightarrow^{v*}} [x.(K_\dagger\{x\})^\dagger]\mathrm{not} \longrightarrow^{v}_{(\eta_R)} \big([x.(K_\dagger\{x\})^\dagger]\mathrm{not} \bullet \alpha\big).\alpha \\ &\equiv (\lambda x.K_\dagger\{x\} :_v \alpha).\alpha \equiv (\lambda x.K_\dagger\{x\})^\dagger \equiv ([K]\mathrm{not}_\dagger)^\dagger\end{aligned}$$

Case of $\lambda x.M$:

$$\begin{aligned}\lambda x.M &\overset{I.H.}{\longrightarrow^{v*}} \lambda x.(M_\dagger)^\dagger \longrightarrow^{v}_{(\eta_R)} (\lambda x.(M_\dagger)^\dagger \bullet \alpha).\alpha \equiv (\lambda x.M_\dagger :_v \alpha).\alpha \\ &\equiv (\lambda x.M_\dagger)^\dagger \equiv ((\lambda x.M)_\dagger)^\dagger\end{aligned}$$

Case of $S.\alpha$:

$$S.\alpha \overset{I.H.}{\longrightarrow^{v*}} (S_\dagger)^\dagger.\alpha \equiv (S_\dagger)^\dagger[{}^\beta/_\alpha].\beta \equiv (\mu\alpha.S_\dagger :_v \beta).\beta \equiv (\mu\alpha.S_\dagger)^\dagger \equiv ((S.\alpha)_\dagger)^\dagger$$

Case of α: $(O :_v \alpha) \equiv ([\alpha]O)^\dagger \equiv (\alpha_\dagger\{O\})^\dagger$

Case of $[K, L]$:

$$(O :_v [K, L]) \longrightarrow^{v*}_{(\eta_L)} (O :_v [x.(x \bullet K), y.(y \bullet L)]) \overset{I.H.}{\longrightarrow^{v*}} (O :_v [x.(K_\dagger\{x\})^\dagger, y.(L_\dagger\{y\})^\dagger])$$

$$\equiv \delta(O, x.K_{\dagger}\{x\}, y.L_{\dagger}\{y\})^{\dagger} \equiv ([K, L]_{\dagger}\{O\})^{\dagger}$$

Case of fst$[K]$:

$$(O :_v \mathrm{fst}[K]) \equiv (\mathrm{fst}(O) :_v K) \overset{I.H.}{\longrightarrow^{v*}} (K_{\dagger}\{\mathrm{fst}(O)\})^{\dagger} \overset{Lem\ 2.29}{\longrightarrow^{v*}} (K_{\dagger}[\mathrm{fst}(O)])^{\dagger} \equiv (\mathrm{fst}[K]_{\dagger}\{O\})^{\dagger}$$

snd$[K]$ can be shown similarly.
Case of not$\langle M\rangle$:

$$(O :_v \mathrm{not}\langle M\rangle) \overset{I.H.}{\longrightarrow^{v*}} (O :_v \mathrm{not}\langle (M_{\dagger})^{\dagger}\rangle) \longrightarrow^{v*} (OM_{\dagger})^{\dagger} \equiv (\mathrm{not}\langle M\rangle_{\dagger}\{O\})^{\dagger}$$

Case of $M@K$:

$$(O :_v (M@K)) \overset{I.H.}{\longrightarrow^{v*}} \big(O :_v ((M_{\dagger})^{\dagger}@K)\big) \longrightarrow^{v*} (OM_{\dagger} :_v K) \overset{I.H.}{\longrightarrow^{v*}} (K_{\dagger}\{OM_{\dagger}\})^{\dagger}$$
$$\overset{Lem\ 2.29}{\longrightarrow^{v*}} (K_{\dagger}[OM_{\dagger}])^{\dagger} \equiv ((M@K)_{\dagger}\{O\})^{\dagger}$$

Case of $x.S$: $(O :_v \bullet x.S) \overset{I.H.}{\longrightarrow^{v*}} (O :_v x.(S_{\dagger})^{\dagger}) \equiv ((\lambda x.S_{\dagger})O)^{\dagger} \longrightarrow^{v*} ((x.S)_{\dagger}\{O\})^{\dagger}$
Case of $M \bullet K$:

$$(M \bullet K) \overset{I.H.}{\longrightarrow^{v*}} (M_{\dagger})^{\dagger} \bullet K \longrightarrow^{v}_{(\beta_L)} (M_{\dagger} :_v K) \overset{I.H.}{\longrightarrow^{v*}} (K_{\dagger}\{M_{\dagger}\})^{\dagger}$$
$$\overset{Lem\ 2.29}{\longrightarrow^{v*}} (K_{\dagger}[M_{\dagger}])^{\dagger} \equiv ((M \bullet K)_{\dagger})^{\dagger}$$

□

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 2.4 and 2.5.

Proposition 2.37 (Church-Rosser property for the $\lambda\mu$-calculus)

(1) If $\lambda\mu \vdash O \longrightarrow_n^* M$ and $\lambda\mu \vdash O \longrightarrow_n^* M'$, then there exists a term O' such that $\lambda\mu \vdash M \longrightarrow_n^* O'$ and $\lambda\mu \vdash M' \longrightarrow_n^* O'$.
If $\lambda\mu \vdash S \longrightarrow_n^* T$ and $\lambda\mu \vdash S \longrightarrow_n^* T'$, then there exists a term S' such that $\lambda\mu \vdash T \longrightarrow_n^* S'$ and $\lambda\mu \vdash T' \longrightarrow_n^* S'$.

(2) If $\lambda\mu \vdash O \longrightarrow_v^* M$ and $\lambda\mu \vdash O \longrightarrow_v^* M'$, then there exists a term O' such that $\lambda\mu \vdash M \longrightarrow_v^* O'$ and $\lambda\mu \vdash M' \longrightarrow_v^* O'$.
If $\lambda\mu \vdash S \longrightarrow_v^* T$ and $\lambda\mu \vdash S \longrightarrow_v^* T'$, then there exists a term S' such that $\lambda\mu \vdash T \longrightarrow_v^* S'$ and $\lambda\mu \vdash T' \longrightarrow_v^* S'$.

Proof. (1) We show the first line of (1). Suppose $\lambda\mu \vdash O \longrightarrow_n^* M$ and $\lambda\mu \vdash O \longrightarrow_n^* M'$, then $DC \vdash (O)^\sharp \longrightarrow^{n*} (M)^\sharp$ and $DC \vdash (O)^\sharp \longrightarrow^{n*} (M')^\sharp$ by Theorem 2.16. By the Church-Rosser property of the dual calculus, there is a term N of the dual calculus such that $DC \vdash (M)^\sharp \longrightarrow^{n*} N$ and $DC \vdash (M')^\sharp \longrightarrow^{n*} N$. Hence we have $\lambda\mu \vdash ((M)^\sharp)_\sharp \longrightarrow_n^* (N)_\sharp$ and $\lambda\mu \vdash ((M')^\sharp)_\sharp \longrightarrow_n^* (N)_\sharp$ by Theorem 2.28. Therefore we obtain $\lambda\mu \vdash M \longrightarrow_n^* (N)_\sharp$ and $\lambda\mu \vdash M' \longrightarrow_n^* (N)_\sharp$ by Proposition 2.35. The second line of (1) is shown similarly. (2) is also shown in a way similar to (1). □

2.6 Duality of call-by-name and call-by-value

Duality is the essential feature of the dual calculus. The dual calculus corresponds to Gentzen's sequent calculus and has explicit duality of classical logic at each level.
- 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

$$\begin{array}{cc}
(X)^\circ \equiv X & (\neg A)^\circ \equiv \neg A^\circ \\
(A \wedge B)^\circ \equiv B^\circ \vee A^\circ & (A \vee B)^\circ \equiv B^\circ \wedge A^\circ \\
(x)^\circ \equiv x & (\alpha)^\circ \equiv \alpha \\
(\langle M, N\rangle)^\circ \equiv [N^\circ, M^\circ] & ([K, L])^\circ \equiv \langle L^\circ, K^\circ\rangle \\
(\langle M\rangle\mathrm{inl})^\circ \equiv \mathrm{snd}[M^\circ] & (\mathrm{fst}[K])^\circ \equiv \langle K^\circ\rangle\mathrm{inr} \\
(\langle N\rangle\mathrm{inr})^\circ \equiv \mathrm{fst}[N^\circ] & (\mathrm{snd}[L])^\circ \equiv \langle L^\circ\rangle\mathrm{inl} \\
(S.\alpha)^\circ \equiv x.S^\circ & (x.S)^\circ \equiv S^\circ.x \\
\multicolumn{2}{c}{(M \bullet K)^\circ \equiv K^\circ \bullet M^\circ}
\end{array}$$

Proposition 2.38 (Duality for the dual calculus)
(*Involution*) Duality is an involution, that is,

$$A^{\circ\circ} \equiv A,\ M^{\circ\circ} \equiv M,\ K^{\circ\circ} \equiv K,\ \text{and } S^{\circ\circ} \equiv S.$$

(*Expressions and typing rules*)

(a) For any term M of the dual calculus, M° is a coterm.
If M has type A, then M° also has type A, *i.e.*,

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

where Γ° is $x_m : A_m^\circ, \ldots, x_1 : A_1^\circ$ for $\Gamma \equiv x_1 : A_1, \ldots, x_m : A_m$, and Δ° 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° is a term.
If K has type A, then K° 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° 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 $DC \vdash M^\circ \longrightarrow^v N^\circ$.
If $DC \vdash K \longrightarrow^n L$, then $DC \vdash K^\circ \longrightarrow^v L^\circ$.
If $DC \vdash S \longrightarrow^n T$, then $DC \vdash S^\circ \longrightarrow^v T^\circ$.

(b) If $DC \vdash M \longrightarrow^v N$, then $DC \vdash M^\circ \longrightarrow^n N^\circ$.
If $DC \vdash K \longrightarrow^v L$, then $DC \vdash K^\circ \longrightarrow^n L^\circ$.
If $DC \vdash S \longrightarrow^v T$, then $DC \vdash S^\circ \longrightarrow^n T^\circ$.

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 $\lambda\mu$-calculi.

Definition 2.9 (The translation from the CBN $\lambda\mu$ into the CBV $\lambda\mu$)
Let A be a type, M and O be terms, and S be a statement of the $\lambda\mu$-calculus. Then we define the translation $(-)_\circ$ as follows.

$$(A)_\circ \equiv A^\circ$$
$$M_\circ\{O\} \equiv ((M^\sharp)^\circ)_\dagger\{O\}$$
$$S_\circ \equiv ((S^\sharp)^\circ)_\dagger$$

Definition 2.10 (The translation from the CBV $\lambda\mu$ into the CBN $\lambda\mu$)
Let A be a type, M and O be terms, and S be a statement of the $\lambda\mu$-calculus. Then we define the translation $(-)_\bullet$ as follows.

$$(A)_\bullet \equiv A^\circ$$
$$M_\bullet\{O\} \equiv ((M^\dagger)^\circ)_\sharp\{O\}$$
$$S_\bullet \equiv ((S^\dagger)^\circ)_\sharp$$

The following properties of these translations are easily shown.

Proposition 2.39

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

(2) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$ and $\Delta_\circ \vdash_{\lambda\mu} \Gamma_\circ \mid O : A_\circ$, then $\Gamma \mid M_\circ\{O\} \vdash_{\lambda\mu} \Delta$.
If $\Gamma \mid S \vdash_{\lambda\mu} \Delta$, then $\Gamma \mid S_\circ \vdash_{\lambda\mu} \Delta$.

(3) If $\Gamma \vdash_{\lambda\mu} \Delta \mid M : A$ and $\Delta_\bullet \vdash_{\lambda\mu} \Gamma_\bullet \mid O : A_\bullet$, then $\Gamma \mid M_\bullet\{O\} \vdash_{\lambda\mu} \Delta$.
If $\Gamma \mid S \vdash_{\lambda\mu} \Delta$, then $\Gamma \mid S_\bullet \vdash_{\lambda\mu} \Delta$.

Then, we obtain our final results.

Theorem 2.40
Let M, N, and O be terms, and S and T be statements. Then the following hold.

(1) The translation $(-)_\circ$ preserves reductions.

$$\lambda\mu \vdash M \longrightarrow_n N \text{ implies } \lambda\mu \vdash M_\circ\{O\} \longrightarrow_v N_\circ\{O\}$$
$$\lambda\mu \vdash S \longrightarrow_n T \text{ implies } \lambda\mu \vdash S_\circ \longrightarrow_v T_\circ$$

(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.

$$\lambda\mu \vdash M \longrightarrow_n \mu\alpha.(M_\circ\{\alpha\})_\bullet$$

$$\lambda\mu \vdash O_\bullet\{M\} \longrightarrow_n (M_\circ\{O\})_\bullet$$
$$\lambda\mu \vdash S \longrightarrow_n (S_\circ)_\bullet$$

(4) The composition of translations obtained by applying $(-)_\circ$ after $(-)_\bullet$ is identity up to the call-by-value reductions.

$$\lambda\mu \vdash M \longrightarrow_v \mu\alpha.(M_\bullet\{\alpha\})_\circ$$
$$\lambda\mu \vdash O_\circ\{M\} \longrightarrow_v (M_\bullet\{O\})_\circ$$
$$\lambda\mu \vdash S \longrightarrow_v (S_\bullet)_\circ$$

Proof. (1) is shown by using Theorem 2.16, Theorem 2.34, and Proposition 2.38.
(2) is shown by using Theorem 2.21, Theorem 2.28, and Proposition 2.38.
(3) follows from Proposition 2.35, 2.36, and 2.38. We show the third line first.

$$S \overset{Prop\ 2.35(1)}{\longrightarrow_n^*} (S^\sharp)_\sharp \equiv (S^{\sharp\circ\circ})_\sharp \overset{Prop\ 2.36(2)}{\longrightarrow_n^*} (((S^{\sharp\circ})_\dagger)^{\dagger\circ})_\sharp \equiv (S_\circ)_\bullet$$

The second line is shown as follows.

$$\begin{aligned} O_\bullet\{M\} &\equiv (O^{\dagger\circ})_\sharp\{M\} \overset{Prop\ 2.35(1)}{\longrightarrow_n^*} (O^{\dagger\circ})_\sharp\{(M^\sharp)_\sharp\} \equiv (M^\sharp \bullet O^{\dagger\circ})_\sharp \\ &\equiv (M^{\sharp\circ\circ} \bullet O^{\dagger\circ})_\sharp \equiv ((O^\dagger \bullet M^{\sharp\circ})^\circ)_\sharp \\ &\overset{Prop\ 2.36(2)}{\longrightarrow_n^*} (((M^{\sharp\circ})_\dagger\{O\})^{\dagger\circ})_\sharp \equiv (M_\circ\{O\})_\bullet \end{aligned}$$

The first line follows from the second line.

$$M \longrightarrow_{(\eta_\mu)} \mu\alpha.[\alpha]M \overset{(*)}{\equiv} \mu\alpha.\alpha_\bullet\{M\} \longrightarrow_n^* \mu\alpha.(M_\circ\{\alpha\})_\bullet$$

$(*)$ is shown by

$$\begin{aligned} \alpha_\bullet\{M\} &\equiv (\alpha^{\dagger\circ})_\sharp\{M\} \equiv ((\alpha \bullet \beta).\beta)^\circ)_\sharp\{M\} \\ &\equiv (\beta.(\beta \bullet \alpha))_\sharp\{M\} \equiv ([\alpha]\beta)[{}^M/_\beta] \equiv [\alpha]M\ . \end{aligned}$$

(4) can be shown in a way similar to (3). □

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.

2.7 Appendix: Wadler's systems and translations

Types	$A, B ::=$	$X \mid A \wedge B \mid A \vee B \mid \neg A \mid A \supset B$
Terms	$O, M, N ::=$	$x \mid \langle M, N\rangle \mid \mathrm{fst}(M) \mid \mathrm{snd}(N) \mid \mu(\alpha,\beta).S$
		$\mid \lambda x.M \mid OM \mid \mu\alpha.S \mid \lambda x.S$
Statements	$S, T ::=$	$[\alpha]M \mid [\alpha,\beta]M \mid OM$
Typing rules		

$$\frac{\Gamma \mid S \vdash_{\lambda\mu} \Delta, \alpha : A, \beta : B}{\Gamma \vdash_{\lambda\mu} \Delta \mid \mu(\alpha,\beta).S \ : A \vee B}\ \vee I \qquad \frac{\Gamma \vdash_{\lambda\mu} \Delta \mid M : A \vee B}{\Gamma \mid [\alpha,\beta]M \vdash_{\lambda\mu} \Delta, \alpha : A, \beta : B}\ \vee E$$

The other typing rules (Ax), ($\supset I$), ($\supset E$), ($\wedge I$), ($\wedge E_1$), ($\wedge E_2$), ($\neg I$), ($\neg E$), (Act), and ($Pass$) are same as our system.

Syntax and typing rules of the $\lambda\mu$-calculus given in Wadler (2005)

Types	$A, B ::=$	$X \mid A \wedge B \mid A \vee B \mid \neg A \mid A \supset B$
Terms	$M, N ::=$	$x \mid \langle M, N\rangle \mid \langle M\rangle\mathrm{inl} \mid \langle N\rangle\mathrm{inr} \mid [K]\mathrm{not} \mid \lambda x.M \mid S.\alpha$
Coterms	$K, L ::=$	$\alpha \mid [K, L] \mid \mathrm{fst}[K] \mid \mathrm{snd}[L] \mid \mathrm{not}\langle M\rangle \mid M@K \mid x.S$
Statements	$S, T ::=$	$M \bullet K$

The typing rules of the dual calculus (Wadler (2005)) are same as our system.

Syntax and typing rules of the dual calculus given in Wadler (2005)

Values	V, W	$::=$	$x \mid \langle V, W\rangle \mid \mathrm{fst}(V) \mid \mathrm{snd}(W) \mid \lambda x.S \mid \lambda x.M$ $\mid \mu(\alpha,\beta).[\alpha]V \mid \mu(\alpha,\beta).[\beta]W$	
Evaluation context	E	$::=$	$\{-\} \mid \langle E, N\rangle \mid \langle V, E\rangle \mid \mathrm{fst}(E) \mid \mathrm{snd}(E) \mid EM \mid VE$	
Statement context	D	$::=$	$[\alpha]E \mid [\alpha,\beta]E \mid EM \mid VE$	
$(\beta\&_1)$	$\mathrm{fst}\langle V, W\rangle$	$=_v$	V	
$(\beta\&_2)$	$\mathrm{snd}\langle V, W\rangle$	$=_v$	W	
$(\beta\vee)$	$[\alpha',\beta']\mu(\alpha,\beta).S$	$=_v$	$S[\alpha'/\alpha,\beta'/\beta]$	
$(\beta\neg)$	$(\lambda x.S)V$	$=_v$	$S[V/x]$	
$(\beta\supset)$	$(\lambda x.M)V$	$=_v$	$M[V/x]$	
$(\beta\mu)$	$[\alpha']\mu\alpha.S$	$=_v$	$S[\alpha'/\alpha]$	
$(\eta\&)$	$V : A \& B$	$=_v$	$\langle \mathrm{fst}V, \mathrm{snd}\,V\rangle$	
$(\eta\vee)$	$M : A \vee B$	$=_v$	$\mu(\alpha,\beta).[\alpha,\beta]M$	(α,β: fresh)
$(\eta\neg)$	$V : \neg A$	$=_v$	$\lambda x.Vx$	(x: fresh)
$(\eta\supset)$	$V : A \& B$	$=_v$	$\lambda x.Vx$	(x: fresh)
$(\eta\mu)$	M	$=_v$	$\mu\alpha.[\alpha]M$	(α: fresh)
(name)	$D\{M\}$	$=_v$	$(\lambda x.D\{x\})M$	(x: fresh)
(comp)	$D\{(\lambda x.N)M\}$	$=_v$	$(\lambda x.D\{N\})M$	
(ς)	$D\{\mu\alpha.S\}$	$=_v$	$S[D\{-\}/[\alpha]\{-\}]$	

Equality axioms of the $\lambda\mu_v^{wad}$-calculus

$(\beta\&_1)$	$\mathrm{fst}\langle M, N\rangle$	$=_n$	M	
$(\beta\&_2)$	$\mathrm{snd}\langle M, N\rangle$	$=_n$	N	
$(\beta\vee)$	$[\alpha',\beta']\mu(\alpha,\beta).S$	$=_n$	$S[\alpha'/\alpha,\beta'/\beta]$	
$(\beta\neg)$	$(\lambda x.S)N$	$=_n$	$S[N/x]$	
$(\beta\supset)$	$(\lambda x.M)N$	$=_n$	$M[N/x]$	
$(\beta\mu)$	$[\alpha']\mu\alpha.S$	$=_n$	$S[\alpha'/\alpha]$	
$(\eta\&)$	$M : A \& B$	$=_n$	$\langle \mathrm{fst}M, \mathrm{snd}\,M\rangle$	
$(\eta\vee)$	$M : A \vee B$	$=_n$	$\mu(\alpha,\beta).[\alpha,\beta]M$	(α,β: fresh)
$(\eta\neg)$	$M : \neg A$	$=_n$	$\lambda x.Mx$	(x: fresh)
$(\eta\supset)$	$M : A \& B$	$=_n$	$\lambda x.Mx$	(x: fresh)
$(\eta\mu)$	M	$=_n$	$\mu\alpha.[\alpha]M$	(α: fresh)
$(\varsigma\&_1)$	$\mathrm{fst}(\mu\alpha.S)$	$=_n$	$\mu\beta.S[[\beta]\mathrm{fst}\{-[\alpha]\{-\}]$	
$(\varsigma\&_2)$	$\mathrm{snd}(\mu\alpha.S)$	$=_n$	$\mu\beta.S[[\beta]\mathrm{snd}\{-\}/[\alpha]\{-\}]$	
$(\varsigma\vee)$	$[\beta,\gamma]\mu\alpha.S$	$=_n$	$S[[\beta,\gamma]\{-\}/[\alpha]\{-\}]$	
$(\varsigma\neg)$	$(\mu\alpha.S)M$	$=_n$	$S[\{-\}M/[\alpha]\{-\}]$	
$(\varsigma\supset)$	$(\mu\alpha.S)M$	$=_n$	$\mu\beta.S[[\beta]\{-\}M/[\alpha]\{-\}]$	

Equality axioms of the $\lambda\mu_n^{wad}$-calculus

Values	V, W	$::=$	$x \mid \langle V, W\rangle \mid (V \bullet \mathrm{fst}[\alpha]).\alpha \mid (W \bullet \mathrm{snd}[\beta]).\beta$	
			$\mid \langle V\rangle\mathrm{inl} \mid \langle W\rangle\mathrm{inr} \mid \lambda x.M \mid [K]\mathrm{not}$	
Evaluation context	E	$::=$	$\{-\} \mid \langle E, N\rangle \mid \langle V, E\rangle \mid \langle E\rangle\mathrm{inl} \mid \langle E\rangle\mathrm{inr}$	
$(\beta\&_1)$	$\langle V, W\rangle \bullet \mathrm{fst}[K]$	$=^v$	$V \bullet K$	
$(\beta\&_2)$	$\langle V, W\rangle \bullet \mathrm{snd}[L]$	$=^v$	$W \bullet L$	
$(\beta\vee_1)$	$\langle V\rangle\mathrm{inl} \bullet [K, L]$	$=^v$	$V \bullet K$	
$(\beta\vee_2)$	$\langle W\rangle\mathrm{inr} \bullet [K, L]$	$=^v$	$W \bullet L$	
$(\beta\neg)$	$[K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle$	$=^v$	$M \bullet K$	
$(\beta\supset)$	$\lambda x.N \bullet (M@K)$	$=^v$	$M \bullet x.(N \bullet K)$	
(βR)	$(S).\alpha \bullet K$	$=^v$	$S[K/\alpha]$	
(βL)	$V \bullet x.(S)$	$=^v$	$S[V/x]$	
$(\eta\&)$	$V : A \& B$	$=^v$	$\langle (V \bullet \mathrm{fst}[\alpha]).\alpha, (V \bullet \mathrm{snd}[\beta]).\beta\rangle$	$(\alpha, \beta$: fresh)
$(\eta\vee)$	$K : A \vee B$	$=^v$	$[x.(\langle x\rangle\mathrm{inl} \bullet K), y.(\langle y\rangle\mathrm{inr} \bullet K)]$	$(x, y$: fresh)
$(\eta\neg)$	$V : \neg A$	$=^v$	$[x.(V \bullet \mathrm{not}\langle x\rangle)]\mathrm{not}$	(x: fresh)
$(\eta\supset)$	$V : A \supset B$	$=^v$	$\lambda x.((V \bullet (x@\beta)).\beta)$	(x: fresh)
(ηR)	M	$=^v$	$(M \bullet \alpha).\alpha$	(α: fresh)
(ηL)	K	$=^v$	$x.(x \bullet K)$	(x: fresh)
(name)	$E\{M\} \bullet K$	$=^v$	$M \bullet x.(E\{x\} \bullet K)$	(x: fresh)

Equality axioms of Wadler's call-by-value dual calculus ($DC_v^{\eta=}$)

Covalues	P, Q	$::=$	$\alpha \mid [P, Q] \mid x.(\langle x\rangle\mathrm{inl} \bullet P) \mid y.(\langle y\rangle\mathrm{inr} \bullet Q)$	
			$\mid \mathrm{fst}[P] \mid \mathrm{snd}[Q] \mid M@Q \mid \mathrm{not}\langle M\rangle$	
Coevaluation context	F	$::=$	$\{-\} \mid [K, F] \mid [F, P] \mid \mathrm{fst}[F] \mid \mathrm{snd}[F]$	
$(\beta\&_1)$	$\langle M, N\rangle \bullet \mathrm{fst}[P]$	$=^n$	$M \bullet P$	
$(\beta\&_2)$	$\langle M, N\rangle \bullet \mathrm{snd}[Q]$	$=^n$	$N \bullet Q$	
$(\beta\vee_1)$	$\langle M\rangle\mathrm{inl} \bullet [P, Q]$	$=^n$	$M \bullet P$	
$(\beta\vee_2)$	$\langle Q\rangle\mathrm{inr} \bullet [P, Q]$	$=^n$	$N \bullet Q$	
$(\beta\neg)$	$[K]\mathrm{not} \bullet \mathrm{not}\langle M\rangle$	$=^n$	$M \bullet K$	
$(\beta\supset)$	$\lambda x.N \bullet (M@K)$	$=^n$	$M \bullet x.(N \bullet K)$	
(βR)	$(S).\alpha \bullet P$	$=^n$	$S[P/\alpha]$	
(βL)	$M \bullet x.(S)$	$=^n$	$S[M/x]$	
$(\eta\&)$	$M : A \& B$	$=^n$	$\langle (M \bullet \mathrm{fst}[\alpha]).\alpha, (M \bullet \mathrm{snd}[\beta]).\beta\rangle$	$(\alpha, \beta$: fresh)
$(\eta\vee)$	$P : A \vee B$	$=^n$	$[x.(\langle x\rangle\mathrm{inl} \bullet P), y.(\langle y\rangle\mathrm{inr} \bullet P)]$	$(x, y$: fresh)
$(\eta\neg)$	$P : \neg A$	$=^n$	$\mathrm{not}\langle ([\alpha]\mathrm{not} \bullet P).\alpha\rangle$	(α: fresh)
$(\eta\supset)$	$M : A \supset B$	$=^n$	$\lambda x.((M \bullet (x@\beta)).\beta))$	(x: fresh)
(ηR)	M	$=^n$	$(M \bullet \alpha).\alpha$	(α: fresh)
(ηL)	K	$=^n$	$x.(x \bullet K)$	(x: fresh)
(name)	$M \bullet F\{K\}$	$=^n$	$(M \bullet F\{\alpha\}).\alpha \bullet K$	(α: fresh)

Equality axioms of Wadler's call-by-name dual calculus ($DC_n^{\eta=}$)

$(x)^*$	$\equiv$	x	$(\langle M, N\rangle)^*$	$\equiv$	$\langle M^*, N^*\rangle$
$(\mathrm{fst}(O))^*$	$\equiv$	$(O^* \bullet \mathrm{fst}[\alpha]).\alpha$	$(\mathrm{snd}(O))^*$	$\equiv$	$(O^* \bullet \mathrm{snd}[\beta]).\beta$
$(\lambda x.S)^*$	$\equiv$	$[x.(S)^*]\mathrm{not}$	$(OM)^*$	$\equiv$	$O^* \bullet \mathrm{not}\langle M^*\rangle$
$(\mu\alpha.S)^*$	$\equiv$	$(S^*).\alpha$	$([\alpha]M)^*$	$\equiv$	$M^* \bullet \alpha$
$(\mu(\alpha,\beta).S)^*$	$\equiv$	$(\langle(\langle(S)^*.\beta\rangle\mathrm{inr} \bullet \gamma).\alpha\rangle\mathrm{inl} \bullet \gamma).\gamma$			
$([\alpha,\beta]M)^*$	$\equiv$	$M^* \bullet [\alpha,\beta]$			
$(\lambda x.M)^*$	$\equiv$	$\lambda x.M^*$	$(OM)^*$	$\equiv$	$(O^* \bullet (M^* @ \beta)).\beta$

Wadler’s translation from the $\lambda\mu$-calculus into the dual calculus

$(x)_*$	$\equiv$	x	$(\alpha)_*\{O\}$	$\equiv$	$[\alpha]O$
$(\langle M, N\rangle)_*$	$\equiv$	$\langle M_*, N_*\rangle$	$([K, L])_*\{O\}$	$\equiv$	$L_*\{\mu\beta.K_*\{\mu\alpha.[\alpha,\beta]O\}\}$
$(\langle M\rangle\mathrm{inl})_*$	$\equiv$	$\mu(\alpha,\beta).[\alpha]M_*$	$(\mathrm{fst}[K])_*\{O\}$	$\equiv$	$K_*\{\mathrm{fst}(O)\}$
$(\langle N\rangle\mathrm{inr})_*$	$\equiv$	$\mu(\alpha,\beta).[\beta]N_*$	$(\mathrm{snd}[L])_*\{O\}$	$\equiv$	$L_*\{\mathrm{snd}(O)\}$
$([K]\mathrm{not})_*$	$\equiv$	$\lambda x.K_*\{x\}$	$(\mathrm{not}\langle M\rangle)_*\{O\}$	$\equiv$	OM_*
$(\lambda x.M)_*$	$\equiv$	$\lambda x.M_*$	$(M @ K)_*\{O\}$	$\equiv$	$K_*\{OM_*\}$
$(S.\alpha)_*$	$\equiv$	$\mu\alpha.S_*$	$(x.S)_*\{O\}$	$\equiv$	$(\lambda x.S_*)O$
		$(M \bullet K)_*$	$\equiv$	$K_*\{M_*\}$	

Wadler’s translation from the dual calculus into the $\lambda\mu$-calculus

Chapter 3

Polarized dual calculus and logical predicates for polarized linear logic

3.1 Introduction

Much work has been done in order to extend Curry-Howard correspondence to classical logic in the last ten years. The first step was taken by Griffin [25] who observed that `call/cc` corresponded to Peirce's Law. Since then, a number of term calculi for classical logic have been introduced. Among those, Parigot [40] introduced a particularly nice one, *the $\lambda\mu$-calculus*. This calculus corresponds to classical natural deduction in just the same way that the λ-calculus corresponds intuitionistic natural deduction. In the meantime, it has been known since Filinski [17] that there is a computational duality between call-by-value and call-by-name in the presence of continuations. Selinger [45] investigated the duality by giving categorical semantics to the call-by-value and the call-by-name $\lambda\mu$-calculus. Wadler [48] introduced the dual calculus to show this duality in a purely syntactical way. This calculus is a term syntax for classical sequent calculus, and explains the computational duality of call-by-name / call-by-value by the logical duality, namely the duality of the left-hand side / the right-hand side in sequent calculus.

Another approach to understand the duality between call-by-value and call-by-name is *polarized linear logic* (LLP) of Laurent [33]. It is a variant of linear logic with a good semantics in terms of coherent spaces. The most fundamental feature of LLP is that it has a clear distinction between *negative* formulas, for which structural rules can be freely used, and *positive* formulas, for which structural rules are forbidden. LLP is useful in understanding the constructive aspect of classical logic. In particular, LLP suggests a close relationship between the call-by-value / call-by-name duality and positive / negative duality. Laurent defined two translations from the call-by-name and the call-by-value $\lambda\mu$-calculi into LLP, and showed their soundness, *i.e.* these translations preserve reductions. The call-by-name translation $(-)^\circ$ translates a classical formula into a negative one, in particular a classical implication $A \to B$ into a negative formula $!A^\circ \multimap B^\circ$ (so, we call this the negative-translation in this paper). On the other hand, the call-by-value translation $(-)^\bullet$ translates a classical formula into a positive one, in particular a classical implication $A \to B$ into a positive formula $!(A^\bullet \multimap ?B^\bullet)$ (so, we call this the positive-translation in this paper). Furthermore, Laurent showed fullness of the negative-translation (*i.e.* every proof of A° is (equivalent to) an image of a proof of A in classical logic via the negative-translation) in [34]. However, it is not proved (at least explicitly) that the positive-translation is also full. Another work to be done is to give a term syntax for LLP. Although proof-nets provide a nice parallel syntax, it is sometimes space-consuming, and complicated, especially in the presence of additives. Hence it is natural to introduce a term syntax, that is compact and moreover well-related to standard functional programming languages.

In this paper, we first give a term calculus for (a sufficiently large fragment of) LLP, called polarized dual calculus (DCP^-) which is based on the idea of Wadler's dual calculus. We then define two translations from the call-by-name / the call-by-value $\lambda\mu$-calculi into DCP^-, and show their soundness. These translations are almost straightforward adaptions of Laurent's (but the positive translation is slightly different). Finally, we prove fullness of these translations in the similar way to the logical predicate method used in Hasegawa [27].

The notion of logical predicate (unary logical relation) is a well-established tool for studying the semantics of various typed lambda calculi. In particular, logical predicates for intuitionistic linear logic were introduced in Hasegawa [26] for category-theoretic models of linear logic, and applied to prove full completeness of Girard translation from the simply typed lambda calculus to the linear lambda calculus in [27]. We adopt this method to show fullness of Laurent's translations. The use of logical predicates allows us to give a *uniform* proof to the fullness of two translations. In particular, just one Basic Lemma is sufficient for both the positive- and the negative-translations.

The rest of this paper is structured as follows. In Section 2, we introduce the system LLP^- as a fragment of LLP. In Section 3, we give a term calculus DCP^- for LLP^-. In Section 4, we review the call-by-name and the call-by-value $\lambda\mu$-calculus, define the positive- and the negative-translations from the $\lambda\mu$-calculi into DCP^-, and then show their soundness. From Section 5 to 7, we prove fullness of these translations by the logical predicate method.

$$\dfrac{}{\vdash P^{\perp}, P}\,(\mathrm{Ax}) \qquad \dfrac{\vdash \Sigma, P \quad \vdash \Lambda, P^{\perp}}{\vdash \Sigma, \Lambda}\,(\mathrm{Cut})$$

$$\dfrac{\vdash \Sigma, P \quad \vdash \Lambda, Q}{\vdash \Sigma, \Lambda, P \otimes Q}\,\otimes \qquad \dfrac{\vdash \Sigma, N, M}{\vdash \Sigma, N \parr M}\,\parr$$

$$\dfrac{\vdash \Sigma, P}{\vdash \Sigma, P \oplus Q}\,\oplus_1 \qquad \dfrac{\vdash \Sigma, Q}{\vdash \Sigma, P \oplus Q}\,\oplus_2 \qquad \dfrac{\vdash \Sigma, N \quad \vdash \Sigma, M}{\vdash \Sigma, N \,\&\, M}\,\&$$

$$\dfrac{\vdash \Sigma, P}{\vdash \Sigma, ?P}\,? \qquad \dfrac{\vdash \Sigma, N}{\vdash \Sigma, !N}\,!$$

$$\dfrac{\vdash \Sigma}{\vdash \Sigma, N}\,(\mathrm{Weakening}) \qquad \dfrac{\vdash \Sigma,\ N, N}{\vdash \Sigma,\ N}\,(\mathrm{Contraction})$$

Figure 3.1: Inference rules of LLP

3.2 LLP and LLP^{-}

Definition 3.1 (Formulas of LLP)

The *formulas* of LLP are defined as follows:

$$P, Q ::= X \mid P \otimes Q \mid P \oplus Q \mid !N \qquad (\textit{positive formulas})$$

$$N, M ::= X^{\perp} \mid N \parr M \mid N \,\&\, M \mid ?P \qquad (\textit{negative formulas})$$

where X and $X^{\perp}$ are atomic formulas. The *negation* of formula A (denoted by $A^{\perp}$) is defined as in linear logic.

Definition 3.2 (Sequents and inference rules of LLP)

The *sequents* of LLP have the form $\vdash \Sigma$ where Σ is a finite multi-set of formulas among which there is *at most one* positive formula. The *inference rules* of LLP are defined as in figure 3.1.

To give a simple term syntax later, we impose a restriction on LLP.

Definition 3.3 (LLP^{-})

The system LLP^{-} is obtained by restricting $\parr$-rule, &-rule and (Cut)-rule of LLP to those sequents which have *no positive formulas* (other than P, in the case of (Cut)-rule).

Remark 1

The restriction forces some sequents derivable in LLP to be non-derivable in LLP^{-}. For example, $\vdash X^{\perp} \parr Y^{\perp}, X$ is derivable in LLP by the following derivation, but not in LLP^{-}, because one cannot apply $\parr$-rule in the presence of the positive formula X.

$$\dfrac{\dfrac{\dfrac{}{\vdash X^\perp, X}}{\vdash X^\perp, Y^\perp, X}}{\vdash X^\perp \parr Y^\perp, X}\ \parr$$

However, we are mainly interested in proofs of *negative sequents* (*i.e.* those consisting of only negative formulas), and our restriction is quite harmless for them. In fact, we have:

Theorem 3.1
Let Σ be a negative sequent. If Σ has a derivation in LLP, then it also has a derivation in LLP$^-$.

In fact, the latter derivation can be obtained by simply permuting some inference rules in the former derivation, and the permutations needed are invisible in terms of proof-nets. Hence one could say that LLP and LLP$^-$ have the same proof-nets for negative conclusions.

3.3 The system DCP$^-$

In this section, we will define a term calculus DCP$^-$ for LLP$^-$. The types of DCP$^-$ are formulas of LLP$^-$. The *variables* of DCP$^-$ are denoted by $x, y, z, \ldots$.

Definition 3.4 (Terms and Sequents of DCP$^-$)
The terms of DCP$^-$ consist of *positive terms* (denoted by $t, u, \ldots$), *negative terms* (denoted by $k, l, \ldots$), and *neutral terms* (denoted by $\tau, \sigma, \ldots$) which are defined as follows:

$$t,\ u ::= x \mid t \otimes u \mid \mathrm{inl}(t) \mid \mathrm{inr}(u) \mid\ !k \qquad (\textit{positive terms})$$

$$k,\ l ::= x.\tau \mid [k, l] \mid (x, y)\tau \mid\ ?t \qquad (\textit{negative terms})$$

$$\tau,\ \sigma ::= t \bullet k \qquad (\textit{neutral terms})$$

$x.\tau$ and $(x, y)\tau$ are abstractions with x (and y) bound in τ. The set of *free variables* occurring in t, k and τ are denoted by $\mathrm{FV}(t)$, $\mathrm{FV}(k)$ and $\mathrm{FV}(\tau)$ respectively. We identify two terms in the α-equivalence relation, and we will use $\equiv$ for the syntactic identity on terms. The expression $t[u/x]$ denotes a term obtained by substituting u for each free occurrence of a variable x in t (the expressions $k[u/x]$ and $\tau[u/x]$ are used similarly). These are defined in such a way that they do not cause free variable captures.

A *context* of DCP$^-$ (ranged over Σ, Λ, Ξ, $\Theta, \ldots$) is a finite set of variables annotated with *negative* types (denoted by $x_1 : N_1, \ldots x_m : N_m$), in which each variable occurs at most once.

$$\frac{}{\vdash x : P^{\perp}\ ;\ x : P}\ (\mathrm{Ax}) \qquad \frac{\vdash \Sigma\ ;\ t : P \quad \vdash \Lambda\ ;\ k : P^{\perp}}{\vdash \Sigma, \Lambda\ ;\ t \bullet k}\ (\mathrm{Cut})$$

$$\frac{\vdash \Sigma\ ;\ t : P \quad \vdash \Lambda\ ;\ u : Q}{\vdash \Sigma, \Lambda\ ;\ t \otimes u : P \otimes Q}\ \otimes \qquad \frac{\vdash \Sigma, x : N, y : M\ ;\ \tau}{\vdash \Sigma\ ;\ (x,y)\tau : N \parr M}\ \parr$$

$$\frac{\vdash \Sigma\ ;\ t : P}{\vdash \Sigma\ ;\ \mathrm{inl}(t) : P \oplus Q}\ \oplus_1 \qquad \frac{\vdash \Sigma\ ;\ u : Q}{\vdash \Sigma\ ;\ \mathrm{inr}(u) : P \oplus Q}\ \oplus_2 \qquad \frac{\vdash \Sigma\ ;\ k : N \quad \vdash \Sigma\ ;\ l : M}{\vdash \Sigma\ ;\ [k,l] : N \,\&\, M}\ \&$$

$$\frac{\vdash \Sigma\ ;\ t : P}{\vdash \Sigma\ ;\ ?t : ?Pb}\ ? \qquad \frac{\vdash \Sigma\ ;\ k : N}{\vdash \Sigma\ ;\ !k : !N}\ !$$

$$\frac{\vdash \Sigma\ ;\ \Pi}{\vdash \Sigma, x : N\ ;\ \Pi}\ (\mathrm{Weakening}) \qquad \frac{\vdash \Sigma, x : N, y : N\ ;\ \Pi}{\vdash \Sigma, z : N\ ;\ \Pi[z/x, z/y]}\ (\mathrm{Contraction})$$

$$\frac{\vdash \Sigma, x : N\ ;\ \tau}{\vdash \Sigma\ ;\ x.\tau : N}\ (\mathrm{Focus}) \qquad \frac{\vdash \Sigma\ ;\ k : N}{\vdash \Sigma, x : N\ ;\ x \bullet k}\ (\mathrm{Unfocus})$$

Figure 3.2: Types and the typing rules for DCP^{-}

A *typing judgement* of DCP^{-} takes either of the following forms:

$$\vdash \Sigma\ ;\ t : P, \quad \vdash \Sigma\ ;\ k : N \quad \text{or} \quad \vdash \Sigma\ ;\ \tau\ .$$

When it is not necessary to distinguish $t : P$, $k : N$ and τ, we write Π to denote one of them. In this case, $\Pi[u/x]$ means $t[u/x] : P$, $k[u/x] : N$ or $\tau[u/x]$.

Definition 3.5 (The typing Rules)
The *typing rules* of DCP^{-} are displayed in figure 3.2, where the (Cut)-rule and the $\otimes$-rule are defined only when the contexts Σ and Λ have no common element, and the variable x occurring in (Weakening)-rule and (Unfocus)-rule is a fresh (*i.e.* new) variable.

Remark 2
(Unfocus)-rule and (Cut)-rule overlap. In fact, (Unfocus)-rule can be derived from (Cut)-rule and (Ax)-rule. However, the correspondence with proofs of LLP^{-} and derivations of DCP^{-} becomes more closely by the presence of (Unfocus)-rule. (This is also mentioned by Wadler [48] in the paragraph starting with “Rules Cut, Id, RE, and LE overlap;” of Section 3)

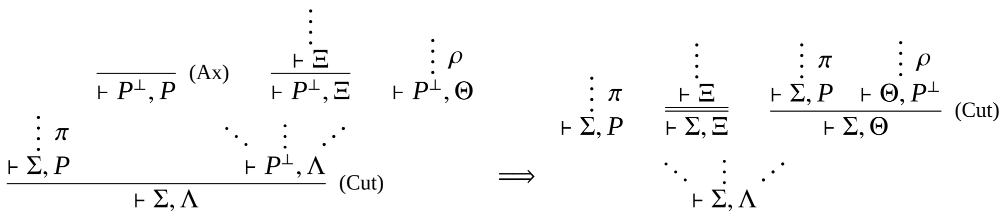
Remark 3
The restriction we imposed on LLP^{-} simplifies the term syntax a lot. For instance, &-rule and $\parr$-rule would be much more complicated without the restriction, and moreover (Focus)-rule and (Unfocus)-rule would be required for positive types too.

Definition 3.6 (Reduction Rules)
The *reduction relation* $\longrightarrow$ of DCP$^-$ is defined to be the compatible closure of the following rules.

$$\begin{array}{lrll}
(\beta) & t \otimes u \bullet (x,y)\tau & \longrightarrow_\beta & u \bullet y.(t \bullet x.\tau) \\
 & \mathrm{inl}(t) \bullet [k,l] & \longrightarrow_\beta & t \bullet k \\
 & \mathrm{inr}(u) \bullet [k,l] & \longrightarrow_\beta & u \bullet l \\
 & !k \bullet ?t & \longrightarrow_\beta & t \bullet k \\
(\varepsilon) & t \bullet x.\tau & \longrightarrow_\varepsilon & \tau[t/x] \\
(\eta) & x.(x \bullet k) & \longrightarrow_\eta & k \qquad \textit{where } x \notin FV(k) \\
 & (x,y)(x \otimes y \bullet k) & \longrightarrow_\eta & k \qquad \textit{where } x,y \notin FV(k) \\
 & !x.(t \bullet ?x) & \longrightarrow_\eta & t \qquad \textit{where } x \notin FV(t)
\end{array}$$

In the following, we use $\longrightarrow^*$, $\longrightarrow^+$ and $=$ as the reflexive transitive closure, the transitive closure and the reflexive symmetric transitive closure of $\longrightarrow$ respectively.

The (β)-rules and the (ε)-rule are intended to capture a natural cut-elimination procedure for LLP$^-$. More specifically, each of the (β)-rules corresponds to a logical reduction step for $\otimes/\parr$, $\oplus_i/\&$ $(i = 1,2)$ and $!/?$. The (ε)-rule roughly corresponds to the following structural reduction step.



In the left proof, ancestors of the negative formula $P^\perp$ are indicated. It must be introduced as an axiom $\overline{\vdash P^\perp, P}$, by (Weakening)-rule $\dfrac{\vdash \Xi}{\vdash P^\perp, \Xi}$, or by a logical inference rule $\begin{array}{c}\vdots\,\rho\\ \vdash P^\perp, \Theta\end{array}$ with $P^\perp$ being the main formula. The above reduction step replaces an axiom by the proof π, and a (Weakening)-rule $\dfrac{\vdash \Xi}{\vdash P^\perp, \Xi}$ by $\dfrac{\vdash \Xi}{\vdash \Sigma, \Xi}$, and $\begin{array}{c}\vdots\,\rho\\ \vdash P^\perp, \Theta\end{array}$ by $\dfrac{\begin{array}{c}\vdots\,\pi\\ \vdash \Sigma, P\end{array} \quad \begin{array}{c}\vdots\,\rho\\ \vdash \Theta, P^\perp\end{array}}{\vdash \Sigma, \Theta}\,(Cut)$. Since Θ consists of only negative formulas by the restriction of LLP$^-$, this (Cut)-rule is certainly LLP$^-$'s. The (η)-rules correspond to the simplification procedure of LLP$^-$ proofs.

We now mention some properties of DCP$^-$. Firstly, this system has subject reduction property: if $\vdash \Sigma; t\colon P$ (resp. $\vdash \Sigma; k\colon N$, $\vdash \Sigma; \tau$) and $t \longrightarrow t'$ (resp. $k \longrightarrow k'$, $\tau \longrightarrow \tau'$) then $\vdash \Sigma; t'\colon P$ (resp. $\vdash \Sigma; k'\colon N$, $\vdash \Sigma; \tau'$). Secondly, it has substitution property: if $\vdash \Sigma, x\colon P^\perp; \Pi$ and $\vdash \Lambda; t\colon P$ then $\vdash \Sigma, \Lambda; \Pi[t/x]$. Finally, it is strongly normalizing. However, it does not enjoy Church-Rosser property. For example, $(x,x')((x \otimes x') \bullet z.(y \bullet ?z))$ reduces to $z.(y \bullet ?z)$

by (η)-rule, $(x, x')(y \bullet ?(x \otimes x'))$ by (ε)-rule, and these are normal. This example reflects the fact that LLP$^-$ with simplification rules is not confluent as follows :

$$\dfrac{\dfrac{\dfrac{}{\vdash P, P^\perp} \quad \dfrac{}{\vdash Q, Q^\perp}}{\vdash P \otimes Q, P^\perp, Q^\perp} \quad \dfrac{\dfrac{}{\vdash P \otimes Q, P^\perp \parr Q^\perp}}{\vdash ?(P \otimes Q), P^\perp \parr Q^\perp}}{\dfrac{\vdash ?(P \otimes Q), P^\perp, Q^\perp}{\vdash ?(P \otimes Q), P^\perp \parr Q^\perp}}\,\mathit{Cut} \quad \text{reduces to} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash P, P^\perp} \quad \dfrac{}{\vdash Q, Q^\perp}}{\vdash P \otimes Q, P^\perp, Q^\perp}}{\vdash ?(P \otimes Q), P^\perp, Q^\perp}}{\vdash ?(P \otimes Q), P^\perp \parr Q^\perp} \quad \text{and} \quad \dfrac{\dfrac{}{\vdash P \otimes Q, P^\perp \parr Q^\perp}}{\vdash ?(P \otimes Q), P^\perp \parr Q^\perp}\,.$$

But this is not so problematic; in fact, if (η)-rules are omitted, then the remaining reductions of DCP$^-$ (*i.e.* (ε)- and (β)-rules) enjoy Church-Rosser.

3.4 The $\lambda\mu$-calculus and the translations into DCP$^-$

3.4.1 The $\lambda\mu$-calculus

We consider two variants of the $\lambda\mu$-calculus, call-by-name and call-by-value, and interpret them in DCP$^-$. First of all, we review the syntax of the $\lambda\mu$-calculus.

Definition 3.7 ($\lambda\mu$-types)
Let $X, Y, \dots$ range over the set of base types. The *types* of the $\lambda\mu$-calculus (denoted by $A, B, \dots$) is generated by the following grammar.

$$A, B ::= X \mid A \to B$$

Definition 3.8 ($\lambda\mu$-terms)
Given two disjoint countable sets of variables, one is called *λ-variables* (denoted by $x, y, z, \dots$) and the other is called *μ-names* (denoted by $\alpha, \beta, \gamma, \dots$). The *(unnamed) terms*, ranged over $w, v, \dots$, and *named-terms*, ranged over $\tau, \sigma, \dots$, of the $\lambda\mu$-calculus are defined by:

$$w, v ::= x \mid \lambda x.w \mid wv \mid \mu\alpha.\tau \qquad (\textit{terms})$$

$$\tau, \sigma ::= [\alpha]w \qquad (\textit{named-terms})$$

We consider terms modulo α-conversion on λ-variables and μ-names. The sets of *free variables* and *free names* of a $\lambda\mu$-term w (resp. τ), denoted by FV(w) and FN(w) (resp. FV(τ) and FN(τ)) respectively, are defined as usual.

Definition 3.9 ($\lambda\mu$-typing rules)
A *typing judgement* of the $\lambda\mu$-calculus takes the form $\Gamma \vdash \Delta \mid u : A$ or $\Gamma \vdash \Delta \mid \tau$, where Γ denotes a *λ-context, i.e.* $x_1 : A_1, \dots, x_n : A_n$, and Δ denotes a *μ-context, i.e.* $\alpha_1 : B_1, \dots, \alpha_m : B_m$. The *typing rules* for the $\lambda\mu$-calculus are defined in the figure 3.3.

$$\frac{}{\Gamma, x : A \vdash \Delta \mid x : A}\ \text{var}$$

$$\frac{\Gamma, x : A \vdash \Delta \mid w : B}{\Gamma \vdash \Delta \mid \lambda x.w : A \to B}\ \lambda\text{-abs} \qquad \frac{\Gamma \vdash \Delta \mid w : A \to B \quad \Sigma \vdash \Lambda \mid v : A}{\Gamma, \Sigma \vdash \Delta, \Lambda \mid wv : B}\ \text{app}$$

$$\frac{\Gamma \vdash \Delta, \alpha : A \mid w}{\Gamma \vdash \Delta \mid \mu\alpha.w : A}\ \mu\text{-abs} \qquad \frac{\Gamma \vdash \Delta \mid w : A}{\Gamma \vdash \Delta, \alpha : A \mid [\alpha]w}\ \text{naming}$$

Figure 3.3: Typing rules for the $\lambda\mu$-calculus

Definition 3.10 (call-by-name reduction rules)
The one-step *call-by-name reduction* relation for the $\lambda\mu$-calculus, written by $\longrightarrow_n$, is defined as the compatible closure of the following rules.

$$\begin{array}{lll}
(\beta) & (\lambda x.w)v \longrightarrow w[v/x] & \\
(\zeta) & (\mu\alpha.\tau)w \longrightarrow \mu\beta.\tau[{}^{[\beta](-)w}/_{[\alpha](-)}] & \\
(\mu_\beta) & [\beta](\mu\alpha.\tau) \longrightarrow \tau[\beta/\alpha] & \\
(\mu_\eta) & \mu\alpha.[\alpha]w \longrightarrow w & (\text{where } \alpha \notin \text{FN}(w))
\end{array}$$

where $w[v/x]$ is the standard substitution of the $\lambda\mu$-terms, and $\tau[\beta/\alpha]$ is just renaming of the free name α. $\tau[{}^{[\beta](-)w}/_{[\alpha](-)}]$ is the result of recursively replacing any subterm of the form $[\alpha]v$ by $[\beta]vw$ in τ.

Definition 3.11 (call-by-value reduction rules)
A *value* is either a variable or a λ-abstraction.

$$V, W ::= x \mid \lambda x.w$$

Let V, W range over values. The one-step *call-by-value reduction* relation for the $\lambda\mu$-calculus, written by $\longrightarrow_v$, is defined as the compatible closure of the following rules.

$$\begin{array}{lll}
(\beta) & (\lambda x.w)V \longrightarrow w[V/x] & \\
(\zeta_{fun}) & (\mu\alpha.\tau)w \longrightarrow \mu\beta.\tau[{}^{[\beta](-)w}/_{[\alpha](-)}] & \\
(\zeta_{arg}) & V(\mu\alpha.\tau) \longrightarrow \mu\beta.\tau[{}^{[\beta]V(-)}/_{[\alpha](-)}] & \\
(\mu_\eta) & \mu\alpha.[\alpha]w \longrightarrow w & (\text{where } \alpha \notin \text{FN}(w)) \\
(\mu_\beta) & [\beta](\mu\alpha.\tau) \longrightarrow \tau[\beta/\alpha] &
\end{array}$$

where $\tau[{}^{[\beta]V(-)}/_{[\alpha](-)}]$ is the result of recursively replacing any subterm of the form $[\alpha]w$ by $[\beta]Vw$ in τ.

We write $\longrightarrow_n^*$ for the reflexive and transitive closure of $\longrightarrow_n$. Similarly for $\longrightarrow_v$.

3.4.2 The negative-translation from the $\lambda\mu$-calculus into DCP$^-$

In this subsection, we give the negative-translation from the $\lambda\mu$-calculus into DCP$^-$, and show that it preserves the call-by-name reductions. It is called negative because it maps the $\lambda\mu$-types to the negative DCP$^-$-types. In particular, it maps $A \to B$ to $!A^\circ \multimap B^\circ$.

According to this translation, both λ-variables and μ-names of the $\lambda\mu$-calculus are interpreted by variables of DCP$^-$, so in the sequel we also use $\alpha, \beta, \gamma, \ldots$ as variables of DCP$^-$.

Definition 3.12 (the negative-translation)
The *negative-translation* consists of three translations: $(-)^\circ$, $[\![-]\!]_{(-)}$ and $[\![-]\!]$. The first one translates a type of the $\lambda\mu$-calculus to a negative type of DCP$^-$, the second one translates a (unnamed) term of the $\lambda\mu$-calculus together with a positive term of DCP$^-$ to a neutral term of DCP$^-$, and the third one translates a named term to a neutral term of DCP$^-$. They are defined as follows :

$$\begin{array}{rlrl}
(X)^\circ & := X^\perp & (A \to B)^\circ & := ?(A^\circ)^\perp \parr B^\circ \\
[\![x]\!]_t & \equiv x \bullet ?t & [\![\lambda x.w]\!]_t & \equiv t \bullet ((x,\alpha)[\![w]\!]_\alpha) \quad \text{(where } \alpha \text{ is fresh)} \\
[\![wv]\!]_t & \equiv [\![w]\!]_{!\overline{[\![v]\!]}\otimes t} & [\![\mu\alpha.\tau]\!]_t & \equiv [\![\tau]\!][t/\alpha] \\
[\![[\alpha]w]\!] & \equiv [\![w]\!]_\alpha & &
\end{array}$$

where t is a positive term of DCP$^-$ and $\overline{[\![w]\!]}$ is an abbreviation of $\beta.[\![w]\!]_\beta$ (β is fresh).

Let Γ be a λ-context $x_1 : A_1, \ldots, x_n : A_n$ and Δ be a μ-context $\alpha_1 : B_1, \ldots, \alpha_m : B_m$ of the $\lambda\mu$-calculus respectively. We define the contexts $?(\Gamma^\circ)^\perp$ and Δ° as $x_1 : ?(A_1^\circ)^\perp, \ldots, x_n : ?(A_n^\circ)^\perp$ and $\alpha_1 : B_1^\circ, \ldots, \alpha_m : B_m^\circ$.

Proposition 3.2
The negative-translation is sound for derivation, that is
(1) if $\Gamma \vdash_{\lambda\mu} \Delta \mid w : A$ and $\vdash_{\text{DCP-}} \Sigma\, ;\ t : (A^\circ)^\perp$ then $\vdash_{\text{DCP-}} ?(\Gamma^\circ)^\perp, \Delta^\circ, \Sigma\, ;\ [\![w]\!]_t$, and
(2) if $\Gamma \vdash_{\lambda\mu} \Delta \mid \tau$ then $\vdash_{\text{DCP-}} ?(\Gamma^\circ)^\perp, \Delta^\circ\, ;\ [\![\tau]\!]$.

Proof. Simultaneous induction of (1) and (2) on $\vdash_{\lambda\mu}$.

Case of (Ax) : assume $\Gamma, x : N \vdash \Delta \mid x : A$ and $\vdash \Sigma\, ;\ t : (A^\circ)^\perp$, then we obtain the following:

$$\dfrac{\dfrac{\vdash x : ?(A^\circ)^\perp\, ;\ x : !A^\circ \qquad \dfrac{\vdash \Sigma\, ;\ t : (A^\circ)^\perp}{\vdash \Sigma\, ;\ ?t : ?(A^\circ)^\perp}}{\vdash x : ?(A^\circ)^\perp, \Sigma\, ;\ x \bullet ?t}}{\vdash x : ?(A^\circ)^\perp, ?(\Gamma^\circ)^\perp, \Delta^\circ, \Sigma; x \bullet ?t}\, Wk$$

Case of (λ-abs) : we consider the case that $\Gamma \vdash \Delta \mid \lambda x.w : A \to B$ is derived from $\Gamma, x : A \vdash \Delta \mid w : B$, and suppose $\vdash \Sigma\ ;\ t : (?(A^\circ)^\perp \parr B^\circ)^\perp$. Using the induction hypothesis to this sequent and $\vdash z : B^\circ\ ;\ z : (B^\circ)^\perp$, we obtain $\vdash ?(\Gamma^\circ)^\perp, \Delta^\circ, \Sigma\ ;\ t \bullet (x,z)\llbracket w \rrbracket_z$. So, we can derive the conclusion of this case as follows.

$$\dfrac{\dfrac{\vdash ?(\Gamma^\circ)^\perp, \Delta^\circ, x : ?(A^\circ)^\perp, z : B^\circ\ ;\ \llbracket w \rrbracket_z}{\vdash ?(\Gamma^\circ)^\perp, \Delta^\circ\ ;\ (x,z)\llbracket w \rrbracket_z : ?(A^\circ)^\perp \parr B^\circ} \quad \vdash \Sigma\ ;\ t : (?(A^\circ)^\perp \parr B^\circ)^\perp}{\vdash ?(\Gamma^\circ)^\perp, \Delta^\circ, \Sigma\ ;\ t \bullet (x,z)\llbracket w \rrbracket_z}$$

Case of (app) : we consider the case that $\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2 \mid wv : B$ is derived from $\Gamma_1 \vdash \Delta_1 \mid w : A \to B$ and $\Gamma_2 \vdash \Delta_2 \mid v : A$, and suppose $\vdash \Sigma\ ;\ t : (B^\circ)^\perp$. Using the induction hypothesis to the latter sequent and $\vdash z : A^\circ\ ;\ z : (A^\circ)^\perp$, we obtain $\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ, z : A^\circ\ ;\ \llbracket v \rrbracket_z$. So, we can derive $\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ, \Sigma\ ;\ !\overline{\llbracket v \rrbracket} \otimes t : !A^\circ \otimes (B^\circ)^\perp$ as follows.

$$\dfrac{\dfrac{\dfrac{\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ, z : A^\circ\ ;\ \llbracket v \rrbracket_z : A^\circ}{\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ\ ;\ \overline{\llbracket v \rrbracket} : A^\circ}}{\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ\ ;\ !\overline{\llbracket v \rrbracket} : !A^\circ} \qquad \vdash \Sigma\ ;\ t : (B^\circ)^\perp}{\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ, \Sigma\ ;\ !\overline{\llbracket v \rrbracket} \otimes t : !A^\circ \otimes (B^\circ)^\perp}$$

Now, we apply the induction hypothesis to $\Gamma_1 \vdash \Delta_1 \mid w : A \to B$ and $\vdash ?(\Gamma_2^\circ)^\perp, \Delta_2^\circ, \Sigma\ ;\ !\overline{\llbracket v \rrbracket} \otimes t : !A^\circ \otimes (B^\circ)^\perp$, then we obtain the conclusion of this case $\vdash ?(\Gamma_1^\circ)^\perp, ?(\Gamma_2^\circ)^\perp, \Delta_1^\circ, \Delta_2^\circ, \Sigma\ ;\ \llbracket w \rrbracket_{!\overline{\llbracket v \rrbracket} \otimes t}$.

Case of (μ-app) : we consider the case that $\Gamma \vdash \Delta \mid \mu\alpha.\tau : A$ is derived from $\Gamma \vdash \Delta, \alpha : A \mid \tau$, and suppose $\vdash \Sigma\ ;\ t : (A^\circ)^\perp$. Apply the induction hypothesis to $\Gamma \vdash \Delta, \alpha : A \mid \tau$, then we have $\vdash ?(\Gamma^\circ)^\perp, \Delta^\circ, \alpha : A^\circ\ ;\ \llbracket \tau \rrbracket$. From the substitution lemma, we obtain the conclusion of this case $\vdash ?(\Gamma^\circ)^\perp, \Delta^\circ, \Sigma\ ;\ \llbracket \tau \rrbracket[t/\alpha]$.

Case of (naming) : we consider the case that $\Gamma \vdash \Delta, \alpha : A \mid [\alpha]w$ is derived from $\Gamma \vdash \Delta \mid w : A$. Now, we apply the induction hypothesis to $\Gamma \vdash \Delta \mid w : A$ and $\vdash \alpha : A^\circ\ ;\ \alpha : (A^\circ)^\perp$, then we obtain the conclusion of this case $\vdash ?(\Gamma^\circ)^\perp, \Delta^n t\ ;\ \llbracket w \rrbracket_\alpha$. □

From (1), $\vdash_{\text{DCP-}} \beta : A^\circ ; \beta : (A^\circ)^\perp$ and (Focus)-rule, it follows that if $\Gamma \vdash_{\lambda\mu} \Delta \mid w : A$ then $\vdash_{\text{DCP-}} ?(\Gamma^\circ)^\perp, \Delta^\circ\ ;\ \overline{\llbracket w \rrbracket} : A^\circ$.

Lemma 3.3

Let w and v be $\lambda\mu$-terms, and t and u be positive terms of DCP$^-$. Then the following hold.

(1) If $t \longrightarrow_n u$ then $\llbracket w \rrbracket_t \longrightarrow^* \llbracket w \rrbracket_u$.

(2) If $\alpha \notin FN(w)$ then $\llbracket w \rrbracket_t[u/\alpha] \equiv \llbracket w \rrbracket_{t[u/\alpha]}$.
If $\alpha \notin FN(\tau)$ then $\llbracket \tau \rrbracket[u/\alpha] \equiv \llbracket \tau \rrbracket$.

(3) $\llbracket w \rrbracket_t[{}^{!\overline{\llbracket v \rrbracket}}/{}_x] \longrightarrow^* \llbracket w[{}^v/{}_x] \rrbracket_{t[{}^{!\overline{\llbracket v \rrbracket}}/{}_x]}$, and
$\llbracket \tau \rrbracket[{}^{!\overline{\llbracket v \rrbracket}}/{}_x] \longrightarrow^* \llbracket \tau[{}^v/{}_x] \rrbracket$.

(4) If β is fresh then
$\llbracket w \rrbracket_t[{}^{!\overline{\llbracket v \rrbracket}\otimes\beta}/{}_\alpha] \equiv \llbracket w[{}^{[\beta](-)v}/{}_{[\alpha](-)}] \rrbracket_{t[{}^{!\overline{\llbracket v \rrbracket}\otimes\beta}/{}_\alpha]}$, and
$\llbracket \tau \rrbracket[{}^{!\overline{\llbracket v \rrbracket}\otimes\beta}/{}_\alpha] \equiv \llbracket \tau[{}^{[\beta](-)v}/{}_{[\alpha](-)}] \rrbracket$.

(5) $\llbracket w \rrbracket_t[\beta/\alpha] \equiv \llbracket w[\beta/\alpha] \rrbracket_{t[\beta/\alpha]}$, and
$\llbracket \tau \rrbracket[\beta/\alpha] \equiv \llbracket \tau[\beta/\alpha] \rrbracket$.

Proof.
(1) By induction on w.

Case of x : we obtain $\llbracket x \rrbracket_t \equiv x \bullet ?t \longrightarrow x \bullet ?u \equiv \llbracket x \rrbracket_u$.

Case of wv : we obtain $\llbracket wv \rrbracket_t \equiv \llbracket w \rrbracket_{!\overline{\llbracket v \rrbracket}\otimes t} \longrightarrow^* \llbracket w \rrbracket_{!\overline{\llbracket v \rrbracket}\otimes u} \equiv \llbracket wv \rrbracket_u$ by the induction hypothesis.

Case of $\lambda x.w$: we obtain $\llbracket \lambda x.w \rrbracket_t \equiv t \bullet (x,\alpha)\llbracket w \rrbracket_\alpha \longrightarrow u \bullet (x,\alpha)\llbracket w \rrbracket_\alpha \equiv \llbracket \lambda x.w \rrbracket_u$.

Case of $\mu\alpha.\tau$: we obtain $\llbracket \mu\alpha.\tau \rrbracket_t \equiv \llbracket \tau \rrbracket[t/\alpha] \longrightarrow^* \llbracket \tau \rrbracket[u/\alpha] \equiv \llbracket \mu\alpha.\tau \rrbracket_u$.

(2) By inducion on w and τ.

Case of x : we obtain $\llbracket x \rrbracket_t[u/\alpha] \equiv (x \bullet ?t)[u/\alpha] \equiv x \bullet ?(t[u/\alpha]) \equiv \llbracket x \rrbracket_{t[u/\alpha]}$.

Case of wv : we obtain

$$\llbracket wv \rrbracket_t[u/\alpha] \equiv \llbracket w \rrbracket_{!\overline{\llbracket v \rrbracket}\otimes t}[u/\alpha] \overset{I.H.}{\equiv} \llbracket w \rrbracket_{(!\overline{\llbracket v \rrbracket}\otimes t)[u/\alpha]} \overset{I.H.}{\equiv} \llbracket w \rrbracket_{!\overline{\llbracket v \rrbracket}\otimes t[u/\alpha]} \equiv \llbracket wv \rrbracket_{t[u/\alpha]} \ .$$

Case of $\lambda x.w$: we obtain

$$\llbracket \lambda x.w \rrbracket_t[u/\alpha] \equiv (t \bullet (x,\beta)\llbracket w \rrbracket_\beta)[u/\alpha] \overset{I.H.}{\equiv} t[u/\alpha] \bullet (x,\beta)\llbracket w \rrbracket_\beta \equiv \llbracket \lambda x.w \rrbracket_{t[u/\alpha]} \ .$$

Case of $\mu\beta.\tau$: we obtain

$$\begin{aligned}\llbracket \mu\beta.\tau \rrbracket_t[u/\alpha] &\equiv \llbracket \tau \rrbracket[t/\beta][u/\alpha] \equiv \llbracket \tau \rrbracket[u/\alpha][t[u/\alpha]/\beta] \\ &\overset{I.H.}{\equiv} \llbracket \tau \rrbracket[t[u/\alpha]/\beta] \equiv \llbracket \mu\beta.\tau \rrbracket_{t[u/\alpha]} \ .\end{aligned}$$

Case of $[\beta]w$: From the hypothesis, $\beta \not\equiv \alpha$, so we obtain

$$\llbracket [\beta]w \rrbracket[u/\alpha] \equiv \llbracket w \rrbracket_\beta[u/\alpha] \overset{I.H.}{\equiv} \llbracket w \rrbracket_\beta \equiv \llbracket [\beta]w \rrbracket \ .$$

(3) By inducion on w and τ.

Case of x :

$$\llbracket x\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv (x\bullet ?t)[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv !\overline{\llbracket v\rrbracket}\bullet ?(t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}])$$
$$\longrightarrow (t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}])\bullet\beta.\llbracket v\rrbracket_\beta \longrightarrow \llbracket v\rrbracket_\beta[t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]/\beta] \overset{(2)}{\equiv} \llbracket v\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]}$$

Case of $z(\not\equiv x)$:

$$\llbracket z\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv (z\bullet ?t)[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv z\bullet ?(t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]) \equiv \llbracket z\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]}$$

Case of w_1w_2 :

$$\llbracket w_1w_2\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv \llbracket w_1\rrbracket_{!\overline{\llbracket w_2\rrbracket}\otimes t}[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \xrightarrow{I.H.}{}^{*} \llbracket w_1[v/x]\rrbracket_{(!\overline{\llbracket w_2\rrbracket}\otimes t)[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]}$$
$$\xrightarrow{I.H.\ and\ (1)}{}^{*} \llbracket w_1[v/x]\rrbracket_{!\overline{\llbracket w_2[v/x]\rrbracket}\otimes t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]} \equiv \llbracket (w_1w_2)[v/x]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]}$$

Case of $\lambda z.w$:

$$\llbracket \lambda z.w\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv (t\bullet (z,\beta)\llbracket w\rrbracket_\beta)[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \xrightarrow{I.H.}{}^{*} t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]\bullet (z,\beta)\llbracket w[v/x]\rrbracket_\beta$$
$$\equiv \llbracket \lambda z.w[v/x]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]} \equiv \llbracket (\lambda z.w)[v/x]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]}$$

Case of $\mu\alpha.\tau$:

$$\llbracket \mu\alpha.\tau\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv \llbracket\tau\rrbracket[t/\alpha][{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv \llbracket\tau\rrbracket[{}^{!\overline{\llbracket v\rrbracket}}/_{x}][t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]/\alpha] \xrightarrow{I.H.}{}^{*} \llbracket\tau[v/x]\rrbracket[t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]/\alpha]$$
$$\equiv \llbracket \mu\alpha.\tau[v/x]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]} \equiv \llbracket (\mu\alpha.\tau)[v/x]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}}/_{x}]}$$

Case of $[\alpha]w$:

$$\llbracket [\alpha]w\rrbracket[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \equiv \llbracket w\rrbracket_\alpha[{}^{!\overline{\llbracket v\rrbracket}}/_{x}] \xrightarrow{I.H.}{}^{*} \llbracket w[v/x]\rrbracket_\alpha \equiv \llbracket [\alpha](w[v/x])\rrbracket \equiv \llbracket ([\alpha]w)[v/x]\rrbracket$$

(4) By inducion on w and τ.

Case of x :

$$\llbracket x\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \equiv (x\bullet ?t)[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \equiv x\bullet ?(t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]) \equiv \llbracket x\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}$$

Case of w_1w_2 :

$$\llbracket w_1w_2\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \equiv \llbracket w_1\rrbracket_{!\overline{\llbracket w_2\rrbracket}\otimes t}[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \overset{I.H.}{\equiv} \llbracket w_1[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{(!\overline{\llbracket w_2\rrbracket}\otimes t)[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}$$
$$\overset{I.H.}{\equiv} \llbracket w_1[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{!\overline{\llbracket w_2[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket}\otimes t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}$$

$$
\begin{aligned}
&\equiv \llbracket w_1[{}^{[\beta](-)v}/_{[\alpha](-)}]\; w_2[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}\\
&\equiv \llbracket (w_1 w_2)[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}
\end{aligned}
$$

Case of $\lambda z.w$:

$$
\begin{aligned}
\llbracket \lambda z.w\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] &\equiv (t \bullet (z,\gamma)\llbracket w\rrbracket_\gamma)[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]\\
&\overset{I.H.}{\equiv} t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \bullet (z,\gamma)\llbracket w[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_\gamma\\
&\equiv \llbracket \lambda z.w[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}\\
&\equiv \llbracket (\lambda z.w)[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}
\end{aligned}
$$

Case of $\mu\gamma.\tau$:

$$
\begin{aligned}
\llbracket \mu\gamma.\tau\rrbracket_t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] &\equiv \llbracket \tau\rrbracket[t/\gamma][{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \equiv \llbracket \tau\rrbracket[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}][t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]/\gamma]\\
&\overset{I.H.}{\equiv} \llbracket \tau[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket[t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]/\gamma] \equiv \llbracket \mu\gamma.\tau[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}\\
&\equiv \llbracket (\mu\gamma.\tau)[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{t[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}]}
\end{aligned}
$$

Case of $[\alpha]w$:

$$
\begin{aligned}
\llbracket [\alpha]w\rrbracket[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] &\equiv \llbracket w\rrbracket_\alpha[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \overset{I.H.}{\equiv} \llbracket w[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_{!\overline{\llbracket v\rrbracket}\otimes\beta} \equiv \llbracket w[{}^{[\beta](-)v}/_{[\alpha](-)}]v\rrbracket_\beta\\
&\equiv \llbracket [\beta](w[{}^{[\beta](-)v}/_{[\alpha](-)}]v)\rrbracket \equiv \llbracket ([\alpha]w)[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket
\end{aligned}
$$

Case of $[\gamma]w$ (where $\gamma \not\equiv \alpha$) :

$$
\begin{aligned}
\llbracket [\gamma]w\rrbracket[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] &\equiv \llbracket w\rrbracket_\gamma[{}^{!\overline{\llbracket v\rrbracket}\otimes\beta}/_{\alpha}] \overset{I.H.}{\equiv} \llbracket w[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket_\gamma\\
&\equiv \llbracket [\gamma](w[{}^{[\beta](-)v}/_{[\alpha](-)}])\rrbracket \equiv \llbracket ([\gamma]w)[{}^{[\beta](-)v}/_{[\alpha](-)}]\rrbracket
\end{aligned}
$$

(5) By induction on w and τ.

Case of x :

$$
\llbracket x\rrbracket_t[\beta/\alpha] \equiv (x \bullet ?t)[\beta/\alpha] \equiv x \bullet ?(t[\beta/\alpha]) \equiv \llbracket x\rrbracket_{t[\beta/\alpha]}
$$

Case of $w_1 w_2$:

$$
\begin{aligned}
\llbracket w_1 w_2\rrbracket_t[\beta/\alpha] &\equiv \llbracket w_1\rrbracket_{!\overline{\llbracket w_2\rrbracket}\otimes t}[\beta/\alpha] \overset{I.H.}{\equiv} \llbracket w_1[\beta/\alpha]\rrbracket_{(!\overline{\llbracket w_2\rrbracket}\otimes t)[\beta/\alpha]}\\
&\overset{I.H.\,and\,(1)}{\equiv} \llbracket w_1[\beta/\alpha]\rrbracket_{!\overline{\llbracket w_2[\beta/\alpha]\rrbracket}\otimes t[\beta/\alpha]} \equiv \llbracket w_1[\beta/\alpha]\; w_2[\beta/\alpha]\rrbracket_{t[\beta/\alpha]}\\
&\equiv \llbracket (w_1 w_2)[\beta/\alpha]\rrbracket_{t[\beta/\alpha]}
\end{aligned}
$$

Case of $\lambda z.w$:

$$\begin{aligned}
[\![\lambda z.w]\!]_t[\beta/\alpha] &\equiv (t \bullet (z,\gamma)[\![w]\!]_\gamma)[\beta/\alpha] \overset{I.H.}{\equiv} t[\beta/\alpha] \bullet (z,\gamma)[\![w[\beta/\alpha]]\!]_\gamma \\
&\equiv [\![\lambda z.w[\beta/\alpha]]\!]_{t[\beta/\alpha]} \equiv [\![(\lambda z.w)[\beta/\alpha]]\!]_{t[\beta/\alpha]}
\end{aligned}$$

Case of $\mu\gamma.\tau$:

$$\begin{aligned}
[\![\mu\gamma.\tau]\!]_t[\beta/\alpha] &\equiv [\![\tau]\!][t/\gamma][\beta/\alpha] \equiv [\![\tau]\!][\beta/\alpha][t[\beta/\alpha]/\gamma] \\
&\overset{I.H.}{\equiv} [\![\tau[\beta/\alpha]]\!][t[\beta/\alpha]/\gamma] \equiv [\![\mu\gamma.\tau[\beta/\alpha]]\!]_{t[\beta/\alpha]} \\
&\equiv [\![(\mu\gamma.\tau)[\beta/\alpha]]\!]_{t[\beta/\alpha]}
\end{aligned}$$

Case of $[\alpha]w$:

$$[\![[\alpha]w]\!][\beta/\alpha] \equiv [\![w]\!]_\alpha[\beta/\alpha] \overset{I.H.}{\equiv} [\![w[\beta/\alpha]]\!]_\beta \equiv [\![[\beta](w[\beta/\alpha])]\!] \equiv [\![([\alpha]w)[\beta/\alpha]]\!]$$

Case of $[\gamma]w$ (where $\gamma \not\equiv \alpha$) :

$$[\![[\gamma]w]\!][\beta/\alpha] \equiv [\![w]\!]_\gamma[\beta/\alpha] \overset{I.H.}{\equiv} [\![w[\beta/\alpha]]\!]_\gamma \equiv [\![[\gamma](w[\beta/\alpha])]\!] \equiv [\![([\gamma]w)[\beta/\alpha]]\!]$$

□

Theorem 3.4 (soundness of the negative-translation)
If $w \longrightarrow_n v$ then $[\![w]\!]_t \longrightarrow^* [\![v]\!]_t$ holds for any positive term t.

Proof. By induction on $\longrightarrow_n$.
Base step :

$$\begin{aligned}
[\![(\lambda x.w)v]\!]_t &\equiv [\![\lambda x.w]\!]_{!\overline{[\![v]\!]}\otimes t} \equiv (!\overline{[\![v]\!]} \otimes t) \bullet (x,\alpha)[\![w]\!]_\alpha \longrightarrow t \bullet \alpha.(!\overline{[\![v]\!]} \bullet x.[\![w]\!]_\alpha) \\
&\longrightarrow t \bullet \alpha.([\![w]\!]_\alpha[{}^{!\overline{[\![v]\!]}}/_x]) \overset{Lem\ 3.3\ (1)}{\longrightarrow^*} t \bullet \alpha.([\![w[v/x]]\!]_\alpha) \\
&\longrightarrow [\![w[v/x]]\!]_\alpha[t/\alpha] \overset{Lem\ 3.3\ (2)}{\equiv} [\![w[v/x]]\!]_t \\
[\![(\mu\alpha.\tau)v]\!]_t &\equiv [\![\mu\alpha.\tau]\!]_{!\overline{[\![v]\!]}\otimes t} \equiv [\![\tau]\!][{}^{!\overline{[\![v]\!]}\otimes t}/_\alpha] \equiv [\![\tau]\!][{}^{!\overline{[\![v]\!]}\otimes\beta}/_\alpha][t/\beta] \qquad (\beta\text{: fresh}) \\
&\overset{Lem\ 3.3\ (4)}{\equiv} [\![\ \tau[{}^{[\beta](-)v}/_{[\alpha](-)}]\ ]\!][t/\beta] \equiv [\![\ \mu\beta.\tau[{}^{[\beta](-)v}/_{[\alpha](-)}]\ ]\!]_t \\
[\![\mu\alpha.[\alpha]w]\!]_t &\equiv [\![[\alpha]w]\!][t/\alpha] \equiv [\![w]\!]_\alpha[t/\alpha] \overset{Lem\ 3.3\ (2)}{\equiv} [\![w]\!]_t \\
[\![[\beta]\mu\alpha.\tau]\!] &\equiv [\![\mu\alpha.\tau]\!]_\beta \equiv [\![\tau]\!][\beta/\alpha] \overset{Lem\ 3.3\ (5)}{\equiv} [\![\tau[\beta/\alpha]]\!]
\end{aligned}$$

Inductive step :
Case of $wv \longrightarrow_n w'v$ (obtained from $w \longrightarrow_n w'$) :

$$[\![wv]\!]_t \equiv [\![w]\!]_{!\overline{[\![v]\!]}\otimes t} \overset{I.H.}{\longrightarrow^*} [\![w']\!]_{!\overline{[\![v]\!]}\otimes t} \equiv [\![w'v]\!]_t$$

Case of $wv \longrightarrow_n wv'$ (obtained from $v \longrightarrow_n v'$) :

$$[\![wv]\!]_t \equiv [\![w]\!]_{!\overline{[\![v]\!]}\otimes t} \overset{I.H.\,and\,Lem\ 3.3\ (1)}{\longrightarrow^*} [\![w]\!]_{!\overline{[\![v']\!]}\otimes t} \equiv [\![wv']\!]_t$$

Case of $\lambda x.w \longrightarrow_n \lambda x.w'$ (obtained from $w \longrightarrow_n w'$) :

$$[\![\lambda x.w]\!]_t \equiv t \bullet (x, \alpha)[\![w]\!]_\alpha \overset{I.H.}{\longrightarrow^*} t \bullet (x, \alpha)[\![w']\!]_\alpha \equiv [\![\lambda x.w']\!]_t$$

Case of $\mu\alpha.\tau \longrightarrow_n \mu\alpha.\tau'$ (obtained from $\tau \longrightarrow_n \tau'$) :

$$[\![\mu\alpha.\tau]\!]_t \equiv [\![\tau]\!][t/\alpha] \overset{I.H.}{\longrightarrow^*} [\![\tau']\!][t/\alpha] \equiv [\![\mu\alpha.\tau']\!]_t$$

Case of $[\alpha]w \longrightarrow_n [\alpha]w'$ (obtained from $w \longrightarrow_n w'$) :

$$[\![[\alpha]w]\!] \equiv [\![w]\!]_\alpha \overset{I.H.}{\longrightarrow^*} [\![w']\!]_\alpha \equiv [\![[\alpha]w']\!]$$

□

The case of (β)-reduction is strict, *i.e.* one-step (β)-reduction of the $\lambda\mu$-calculus is translated into at least one step of DCP$^-$ reductions. On the other hand, (ζ), (μ_η) and (μ_β)-reductions are translated into identity.

Remark 4
Our translation does not preserve the call-by-name η-rule of the $\lambda\mu$-calculus (*i.e.* $\lambda x.wx \longrightarrow_n w$ (where $x \notin \mathrm{FV}(w)$)) as the reduction rule. However, it does preserve it as the equational rule. Since $!\overline{[\![x]\!]} \equiv\ !z.[\![x]\!]_z \equiv\ !z.(x \bullet ?z) \longrightarrow x$ by the η-rule of DCP$^-$, we can prove $[\![\lambda x.wx]\!]_t = [\![w]\!]_t$ as follows.

$$\begin{aligned}[\![\lambda x.wx]\!]_t &\equiv t \bullet (x, \alpha)[\![wx]\!]_\alpha \equiv t \bullet (x, \alpha)[\![w]\!]_{!\overline{[\![x]\!]}\otimes\alpha} \longrightarrow^* t \bullet (x, \alpha)[\![w]\!]_{x\otimes\alpha} \\ &= t \bullet (x, \alpha)(x \otimes \alpha \bullet \overline{[\![w]\!]}) \longrightarrow t \bullet \overline{[\![w]\!]} \longrightarrow [\![w]\!]_t\end{aligned}$$

3.4.3 The positive-translation from the $\lambda\mu$-calculus into DCP$^-$

In this subsection, we give the positive-translation from the $\lambda\mu$-calculus into DCP$^-$, and show that it preserves the call-by-value reductions. It is called positive because it maps the $\lambda\mu$-types to the positive DCP$^-$-types. In particular, it maps $A \to B$ to $!(A^\bullet \multimap ?B^\bullet)$.

Definition 3.13 (the positive-translation)
The *positive-translation* consists of the four translations: $(-)^{\bullet}$, $(-)^{*}$, $[\![-]\!]_{(-)}$ and $[\![-]\!]$. The first one translates a type of the $\lambda\mu$-calculus to a positive type of DCP^{-}, the second one translates a value to a positive term of DCP^{-}, and the third and the last one are similar to those in the negative-translation. They are defined as follows:

$$
\begin{array}{rclrcl}
(X)^{\bullet} & = & X & (A \to B)^{\bullet} & = & !((A^{\bullet})^{\perp} \parr ?B^{\bullet}) \\
x^{*} & \equiv & x, & (\lambda x.w)^{*} & \equiv & !(x,\alpha)[\![w]\!]_{\alpha} \text{ (where } \alpha \text{ is fresh)}, \\
[\![V]\!]_t & \equiv & t \bullet ?V^{*}, & & & \\
[\![Vw]\!]_t & \equiv & [\![w]\!]_{!x.(V^{*} \bullet ?(x \otimes t))}, & [\![vw]\!]_t & \equiv & [\![v]\!]_{!x.[\![w]\!]_{!z.(x \bullet ?(z \otimes t))}} \text{ (where } v \text{ is not a value)} \\
[\![[\alpha]w]\!] & \equiv & [\![w]\!]_{\alpha}, & [\![\mu\alpha.\tau]\!]_t & \equiv & [\![\tau]\!][t/\alpha].
\end{array}
$$

In the above definition, we give two kinds of definition for application. This is for the following reason: to obtain a sound translation for the call-by-value reduction, we need to have two views on application depending on the situations. For example, "V is applied to $\mu\alpha.\tau$ from the left-hand side" in the case of (ζ_{arg})-rule, and "w is applied to $\mu\alpha.\tau$ from the right-hand side" in the case of (ζ_{fun})-rule. To solve this dilemma, we think of vw as $(\lambda x.xw)v$ (actually, we use a slightly modified form) when v is not a value. Then, we can always assume that an application is of the form Vw, because $\lambda x.xw$ is a value. Here, if we abbreviate $!x.(V^{*} \bullet ?(x \otimes t))$ by $(V,t)^{\triangleright}$, and $!x.[\![w]\!]_{!z.(x \bullet ?(z \otimes t))}$ by $(w,t)^{\triangleleft}$, then $[\![Vw]\!]_t$ and $[\![vw]\!]_t$ can be written as $[\![w]\!]_{(V,t)^{\triangleright}}$ and $[\![v]\!]_{(w,t)^{\triangleleft}}$ respectively.

Proposition 3.5
The positive-translation is sound for derivation, *i.e.* the followings hold.
(1) If $\Gamma \vdash_{\lambda\mu} \Delta \mid V : A$ then $\vdash_{\text{DCP-}} (\Gamma^{\bullet})^{\perp}, ?\Delta^{\bullet} \ ; \ V^{*} : A^{\bullet}$.
(2) If $\Gamma \vdash_{\lambda\mu} \Delta \mid w : A$ and $\vdash_{\text{DCP-}} \Sigma \ ; \ t : !(A^{\bullet})^{\perp}$, then $\vdash_{\text{DCP-}} (\Gamma^{\bullet})^{\perp}, ?\Delta^{\bullet}, \Sigma \ ; \ [\![w]\!]_t$.
(3) If $\Gamma \vdash_{\lambda\mu} \Delta \mid \tau$ then $\vdash_{\text{DCP-}} (\Gamma^{\bullet})^{\perp}, ?\Delta^{\bullet} \ ; \ [\![\tau]\!]$.

Proof. Simultaneous induction of (1),(2) and (3) on $\vdash_{\lambda\mu}$.

Case of (Ax) : assume $\Gamma, x : N \vdash \Delta \mid x : A$. In this case, we can prove (1) as follows:

$$
\dfrac{\dfrac{}{\vdash x : (A^{\bullet})^{\perp} \ ; \ x : A^{\bullet}}}{\vdash (\Gamma^{\bullet})^{\perp}, ?\Delta^{\bullet}, x : (A^{\bullet})^{\perp} \ ; \ x : A^{\bullet}} \; Wk
$$

Now, we suppose $\vdash \Sigma\ ;\ t : !(A^\bullet)^\perp$, then we can obtain (2) as follows:

$$\dfrac{\vdash \Sigma\ ;\ t : !(A^\bullet)^\perp \qquad \dfrac{\dfrac{\dfrac{}{\vdash x : (A^\bullet)^\perp\ ;\ x : A^\bullet}}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, x : (A^\bullet)^\perp\ ;\ x : A^\bullet}\ Wk}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, x : (A^\bullet)^\perp\ ;\ ?x : ?A^\bullet}}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, x : (A^\bullet)^\perp, \Sigma\ ; t \bullet ?x}$$

Case of (λ-abs) : we consider the case that $\Gamma \vdash \Delta \mid \lambda x.w : A \to B$ is derived from $\Gamma, x : A \vdash \Delta \mid w : B$. So, with $\vdash \alpha : ?B^\bullet\ ;\ \alpha : !(B^\bullet)^\perp$, we obtain $\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, x : (A^\bullet)^\perp, \alpha : ?B^\bullet\ ;\ [\![w]\!]_\alpha$ by the induction hypothesis (2). Therefore, we can prove (1) as follows:

$$\dfrac{\dfrac{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, x : (A^\bullet)^\perp, \alpha : ?B^\bullet\ ;\ [\![w]\!]_\alpha}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet\ ;\ (x,\alpha)[\![w]\!]_\alpha : (A^\bullet)^\perp \parr ?B^\bullet}}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet\ ;\ !(x,\alpha)[\![w]\!]_\alpha : !((A^\bullet)^\perp \parr ?B^\bullet)}$$

Now, we suppose $\vdash \Sigma\ ;\ t : !?(A^\bullet \otimes !(B^\bullet)^\perp)$, then we obtain (2) as follows:

$$\dfrac{\vdash \Sigma\ ;\ t : !?(A^\bullet \otimes !(B^\bullet)^\perp) \qquad \dfrac{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet\ ;\ !(x,\alpha)[\![w]\!]_\alpha : !((A^\bullet)^\perp \parr ?B^\bullet)}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet\ ;\ ?!(x,\alpha)[\![w]\!]_\alpha : ?!((A^\bullet)^\perp \parr ?B^\bullet)}}{\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, \Sigma\ ;\ t \bullet ?!(x,\alpha)[\![w]\!]_\alpha}$$

Case of (app1) : we consider the case that $\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2 \mid Vw : B$ is derived from $\Gamma_1 \vdash \Delta_1 \mid V : A \to B$ and $\Gamma_2 \vdash \Delta_2 \mid w : A$, and suppose $\vdash \Sigma\ ;\ t : !(B^\bullet)^\perp$. By the induction hypothesis (1), we obtain $\vdash (\Gamma_1^\bullet)^\perp, ?(\Delta_1^\bullet)\ ;\ V^* : !((A^\bullet)^\perp \parr ?B^\bullet)$. Hence, we have the following derivation.

$$\dfrac{\dfrac{\dfrac{\vdash (\Gamma_1^\bullet)^\perp, ?(\Delta_1^\bullet)\ ;\ V^* : !((A^\bullet)^\perp \parr ?B^\bullet) \qquad \dfrac{\dfrac{\dfrac{}{\vdash x : (A^\bullet)^\perp\ ;\ x : A^\bullet} \quad \vdash \Sigma\ ;\ t : !(B^\bullet)^\perp}{\vdash \Sigma, x : (A^\bullet)^\perp\ ;\ x \otimes t : A^\bullet \otimes !(B^\bullet)^\perp}}{\vdash \Sigma, x : (A^\bullet)^\perp\ ;\ ?(x \otimes t) : ?(A^\bullet \otimes !(B^\bullet)^\perp)}}{\vdash (\Gamma_1^\bullet)^\perp, ?(\Delta_1^\bullet), \Sigma, x : (A^\bullet)^\perp\ ;\ V^* \bullet ?(x \otimes t)}}{\vdash (\Gamma_1^\bullet)^\perp, ?(\Delta_1^\bullet), \Sigma\ ;\ x.(V^* \bullet ?(x \otimes t)) : (A^\bullet)^\perp}}{\vdash (\Gamma_1^\bullet)^\perp, ?(\Delta_1^\bullet), \Sigma\ ;\ !x.(V^* \bullet ?(x \otimes t)) : !(A^\bullet)^\perp}$$

Here, we apply the induction hypothesis (2) to $\Gamma_2 \vdash \Delta_2 \mid w : A$ and above sequent, we obtain the conclusion of this case: $\vdash (\Gamma_1^\bullet)^\perp, (\Gamma_2^\bullet)^\perp, ?(\Delta_1^\bullet), ?(\Delta_2^\bullet), \Sigma\ ;\ [\![w]\!]_{!x.(V^* \bullet ?(x \otimes t))}$.

Case of (app2) : in this case, we consider that $\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2 \mid vw : B$ (where v is not a value) is derived from $\Gamma_1 \vdash \Delta_1 \mid v : A \to B$ and $\Gamma_2 \vdash \Delta_2 \mid w : A$, and suppose $\vdash \Sigma\ ;\ t : !(B^\bullet)^\perp$. Then,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash x : ?(A^\bullet \otimes !(B^\bullet)^\perp)\ ;\ x : !((A^\bullet)^\perp \parr ?B^\bullet)} \qquad \dfrac{\dfrac{\dfrac{}{\vdash z : (A^\bullet)^\perp\ ;\ z : A^\bullet} \quad \dfrac{}{\vdash \alpha : ?B^\bullet\ ;\ \alpha : !(B^\bullet)^\perp}}{\vdash \alpha : ?B^\bullet, z : (A^\bullet)^\perp\ ;\ z \otimes \alpha : A^\bullet \otimes !(B^\bullet)^\perp}}{\vdash \alpha : ?B^\bullet, z : (A^\bullet)^\perp\ ;\ ?(z \otimes \alpha) : ?(A^\bullet \otimes !(B^\bullet)^\perp)}}{\vdash x : ?(A^\bullet \otimes !(B^\bullet)^\perp), \alpha : ?B^\bullet, z : (A^\bullet)^\perp\ ;\ x \bullet ?(z \otimes \alpha)}}{\vdash x : ?(A^\bullet \otimes !(B^\bullet)^\perp), \alpha : ?B^\bullet\ ;\ z.(x \bullet ?(z \otimes \alpha)) : (A^\bullet)^\perp}}{\vdash x : ?(A^\bullet \otimes !(B^\bullet)^\perp), \alpha : ?B^\bullet\ ;\ !z.(x \bullet ?(z \otimes \alpha)) : !(A^\bullet)^\perp}$$

and the induction hypothesis (2), we have

$$\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), x : ?(A^\bullet \otimes !(B^\bullet)^\perp), \alpha : ?B^\bullet \ ; \ \llbracket v \rrbracket_{!z.(x \bullet ?(z \otimes \alpha))} \ (\equiv \llbracket xv \rrbracket_\alpha)$$

Therefore, we obtain $\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), x : ?(A^\bullet \otimes !(B^\bullet)^\perp), \alpha : ?B^\bullet \ ; \ (\lambda x.xv)^* : !(?(A^\bullet \otimes !(B^\bullet)^\perp) \parr ?B^\bullet)$ from the following derivation.

$$\dfrac{\dfrac{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), x : ?(A^\bullet \otimes !(B^\bullet)^\perp), \alpha : ?B^\bullet \ ; \ \llbracket xv \rrbracket_\alpha}{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet) \ ; \ (x, \alpha)\llbracket xv \rrbracket_\alpha : (?(A^\bullet \otimes !(B^\bullet)^\perp) \parr ?B^\bullet)}}{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet) \ ; \ !(x, \alpha)\llbracket xv \rrbracket_\alpha : !(?(A^\bullet \otimes !(B^\bullet)^\perp) \parr ?B^\bullet)}$$

So, we can derive $\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), \Sigma \ ; \ !z.((\lambda x.xv)^* \bullet ?(z \otimes t)) : !?(A^\bullet \otimes !(B^\bullet)^\perp)$ as follows:

$$\dfrac{\dfrac{\dfrac{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet) \ ; \ (\lambda x.xv)^* : !(?(A^\bullet \otimes !(B^\bullet)^\perp) \parr ?B^\bullet) \quad \dfrac{\dfrac{\dfrac{}{\vdash z : ?(A^\bullet \otimes !(B^\bullet)^\perp) \ ; \ z : !((A^\bullet)^\perp \parr ?B^\bullet)} \quad \vdash \Sigma, \ ; \ t : !(B^\bullet)^\perp}{\vdash \Sigma, z : ?(A^\bullet \otimes !(B^\bullet)^\perp) \ ; \ z \otimes t : !((A^\bullet)^\perp \parr ?B^\bullet) \otimes !(B^\bullet)^\perp}}{\vdash \Sigma, z : ?(A^\bullet \otimes !(B^\bullet)^\perp) \ ; \ ?(z \otimes t) : ?(!((A^\bullet)^\perp \parr ?B^\bullet) \otimes !(B^\bullet)^\perp)}}{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), \Sigma, z : ?(A^\bullet \otimes !(B^\bullet)^\perp) \ ; \ (\lambda x.xv)^* \bullet ?(z \otimes t)}}{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), \Sigma \ ; \ z.((\lambda x.xv)^* \bullet ?(z \otimes t)) : ?(A^\bullet \otimes !(B^\bullet)^\perp)}}{\vdash (\Gamma_2^\bullet)^\perp, ?(\Delta_2^\bullet), \Sigma \ ; \ !z.((\lambda x.xv)^* \bullet ?(z \otimes t)) : !?(A^\bullet \otimes !(B^\bullet)^\perp)}$$

Finally, we obtain the conclusion

$$\vdash (\Gamma_1^\bullet)^\perp, (\Gamma_2^\bullet)^\perp, ?(\Delta_1^\bullet), ?(\Delta_2^\bullet), \Sigma \ ; \ \llbracket w \rrbracket_{!z.((\lambda x.xv)^* \bullet ?(z \otimes t))} \ (\equiv \llbracket vw \rrbracket_t)$$

from the induction hypothesis.

Case of (μ-app) : we consider the case that $\Gamma \vdash \Delta \mid \mu\alpha.\tau : A$ is derived from $\Gamma \vdash \Delta, \alpha : A \mid \tau$, and suppose $\vdash \Sigma \ ; \ t : !(A^\bullet)^\perp$. Apply the induction hypothesis to $\Gamma \vdash \Delta, \alpha : A \mid \tau$, then we have $\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, \alpha : ?A^\bullet \ ; \ \llbracket \tau \rrbracket$. From the substitution lemma, we obtain the conclusion of this case $\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet, \Sigma \ ; \ \llbracket \tau \rrbracket[t/\alpha]$.

Case of (naming) : we consider the case that $\Gamma \vdash \Delta, \alpha : A \mid [\alpha]w$ is derived from $\Gamma \vdash \Delta \mid w : A$. Now, we apply the induction hypothesis to $\Gamma \vdash \Delta \mid w : A$ and $\vdash \alpha : ?A^\bullet \ ; \ \alpha : !(A^\bullet)^\perp$, then we obtain the conclusion of this case $\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet \ ; \ \llbracket w \rrbracket_\alpha$.

□

From (1), $\vdash_{\text{DCP-}} \beta : ?A^\bullet ; \beta : !(A^\bullet)^\perp$ and (Focus)-rule, it follows that if $\Gamma \vdash_{\lambda\mu} \Delta \mid w : A$ then $\vdash_{\text{DCP-}} (\Gamma^\bullet)^\perp, ?\Delta^\bullet \ ; \ \overline{\llbracket w \rrbracket} : ?A^\bullet$.

Lemma 3.6

Let w and v be $\lambda\mu$-terms, and t and u be DCP$^-$-positive terms. Then the following hold.

(1) If $t \longrightarrow_v u$ then $\llbracket w \rrbracket_t \longrightarrow^* \llbracket w \rrbracket_u$.

(2) $W^*[V^*/x] \equiv (W[V/x])^*$,
$[\![w]\!]_t[V^*/x] \equiv [\![w[V/x]]\!]_{t[V^*/x]}$, and
$[\![\tau]\!][V^*/x] \equiv [\![\tau[V/x]]\!]$.

(3) If α is not in FN(V), FN(v) and FN(τ), then
$V^*[t/\alpha] \equiv V^*$,
$[\![v]\!]_u[t/\alpha] \equiv [\![v]\!]_{u[t/\alpha]}$, and
$[\![\tau]\!][t/\alpha] \equiv [\![\tau]\!]$.

(4) $V^*[(w,\beta)^\lhd/\alpha] \longrightarrow^* (V[^{[\beta](-)w}/_{[\alpha](-)}])^*$,
$[\![v]\!]_t[(w,\beta)^\lhd/\alpha] \longrightarrow^* [\![v[^{[\beta](-)w}/_{[\alpha](-)}]]\!]_{t[(w,\beta)^\lhd/\alpha]}$, and
$[\![\tau]\!][(w,\beta)^\lhd/\alpha] \longrightarrow^* [\![\tau[^{[\beta](-)w}/_{[\alpha](-)}]]\!]$.

(5) $W^*[(V,\beta)^\rhd/\alpha] \equiv (W[^{[\beta]V(-)}/_{[\alpha](-)}])^*$,
$[\![v]\!]_t[(V,\beta)^\rhd/\alpha] \equiv [\![v[^{[\beta]V(-)}/_{[\alpha](-)}]]\!]_{t[(V,\beta)^\rhd/\alpha]}$, and
$[\![\tau]\!][(V,\beta)^\rhd/\alpha] \equiv [\![\tau[^{[\beta]V(-)}/_{[\alpha](-)}]]\!]$.

(6) $V^*[\beta/\alpha] \equiv (V[\beta/\alpha])^*$,
$[\![w]\!]_t[\beta/\alpha] \equiv [\![w[\beta/\alpha]]\!]_{t[\beta/\alpha]}$, and
$[\![\tau]\!][\beta/\alpha] \equiv [\![\tau[\beta/\alpha]]\!]$.

Proof. (1) By induction on w.

Case of x : we obtain $[\![x]\!]_t \equiv t \bullet ?x \longrightarrow u \bullet ?x \equiv [\![x]\!]_u$.

Case of Vw : we obtain the following by the induction hypothesis:

$$[\![Vw]\!]_t \equiv [\![w]\!]_{!x.(V^*\bullet ?(x\otimes t))} \longrightarrow^* [\![w]\!]_{!x.(V^*\bullet ?(x\otimes u))} \equiv [\![Vw]\!]_u$$

Case of vw (where v is not a value) : we obtain $(w,t)^\lhd \longrightarrow^* (w,u)^\lhd$ by the induction hypothesis. Again, we apply the induction hypothesis, then we have

$$[\![vw]\!]_t \equiv [\![v]\!]_{(w,t)^\lhd} \longrightarrow^* [\![v]\!]_{(w,u)^\lhd} \equiv [\![vw]\!]_u \ .$$

Case of $\lambda x.w$: we obtain

$$[\![\lambda x.w]\!]_t \equiv t \bullet ?(\lambda x.w)^* \longrightarrow u \bullet ?(\lambda x.w)^* \equiv [\![\lambda x.w]\!]_u \ .$$

Case of $\mu\alpha.\tau$: we obtain

$$[\![\mu\alpha.\tau]\!]_t \equiv [\![\tau]\!][t/\alpha] \longrightarrow^* [\![\tau]\!][u/\alpha] \equiv [\![\mu\alpha.\tau]\!]_u \ .$$

(2) By inducion on W, w and τ.

Case of x : we obtain

$$\begin{aligned} x^*[V^*/x] &\equiv x[V^*/x] \equiv V^* \\ [\![x]\!]_t[V^*/x] &\equiv (t \bullet ?x^*)[V^*/x] \equiv t[V^*/x] \bullet ?(x^*[V^*/x]) \\ &\equiv t[V^*/x] \bullet ?V^* \equiv [\![V]\!]_{t[V^*/x]} \ . \end{aligned}$$

Case of $z (\not\equiv x)$: we obtain

$$\begin{aligned} z^*[V^*/x] &\equiv z[V^*/x] \equiv z \equiv z^* \\ [\![z]\!]_t[V^*/x] &\equiv (t \bullet ?z^*)[V^*/x] \equiv t[V^*/x] \bullet ?(z^*[V^*/x]) \\ &\equiv t[V^*/x] \bullet ?z^* \equiv [\![z]\!]_{t[V^*/x]} \ . \end{aligned}$$

Case of $\lambda z.w$: we obtain

$$\begin{aligned} (\lambda z.w)^*[V^*/x] &\equiv !(z,\alpha)[\![w]\!]_\alpha[V^*/x] \overset{I.H.}{\equiv} !(z,\alpha)[\![w[V/x]]\!]_\alpha \\ &\equiv (\lambda z.w[V/x])^* \equiv ((\lambda z.w)[V/x])^* \\ [\![\lambda z.w]\!]_t[V^*/x] &\equiv (t \bullet ?(\lambda z.w)^*)[V^*/x] \equiv t[V^*/x] \bullet ?(\lambda z.w)^*[V^*/x] \\ &\equiv t[V^*/x] \bullet ?(\lambda z.w[V/x])^* \equiv [\![(\lambda z.w)[V/x]]\!]_{t[V^*/x]} \ . \end{aligned}$$

Case of Ww (where W is a value): By induction hypothesis, we have $(W, t)^{\triangleright}[V^*/x] \equiv (W[V/x], t[V^*/x])^{\triangleright}$. Then we obtain

$$\begin{aligned} [\![Ww]\!]_t[V^*/x] &\equiv [\![w]\!]_{(W,t)^{\triangleright}}[V^*/x] \overset{I.H.}{\equiv} [\![w[V/x]]\!]_{(W,t)^{\triangleright}[V^*/x]} \\ &\equiv [\![w[V/x]]\!]_{(W[V/x],t[V^*/x])^{\triangleright}} \equiv [\![W[V/x]w[V/x]]\!]_{t[V^*/x]} \\ &\equiv [\![(Ww)[V/x]]\!]_{t[V^*/x]} \ . \end{aligned}$$

Case of vw (where v is not a value): By induction hypothesis, we have $(w, t)^{\triangleleft}[V^*/x] \equiv (w[V/x], t[V^*/x])^{\triangleleft}$. Then we obtain

$$\begin{aligned} [\![vw]\!]_t[V^*/x] &\equiv [\![v]\!]_{(w,t)^{\triangleleft}}[V^*/x] \overset{I.H.}{\equiv} [\![v[V/x]]\!]_{(w,t)^{\triangleleft}[V^*/x]} \\ &\equiv [\![v[V/x]]\!]_{(w[V/x],t[V^*/x])^{\triangleleft}} \equiv [\![v[V/x]w[V/x]]\!]_{t[V^*/x]} \\ &\equiv [\![(vw)[V/x]]\!]_{t[V^*/x]} \ . \end{aligned}$$

Case of $\mu\beta.\tau$: we obtain

$$[\![\mu\beta.\tau]\!]_t[V^*/x] \equiv [\![\tau]\!][t/\beta][V^*/x] \equiv [\![\tau]\!][V^*/x][t[V^*/x]/\beta]$$

$$\begin{aligned} &\overset{I.H.}{\equiv} [\![\tau[V/x]]\!][t[V^*/x]/\beta] \equiv [\![\mu\beta.\tau[V/x]]\!]_{t[V^*/x]} \\ &\equiv [\![(\mu\beta.\tau)[V/x]]\!]_{t[V^*/x]} \ . \end{aligned}$$

Case of $[\beta]w$: we obtain

$$\begin{aligned} [\![[\beta]w]\!][V^*/x] &\equiv [\![w]\!]_\beta[V^*/x] \overset{I.H.}{\equiv} [\![w[V/x]]\!]_\beta \equiv [\![[\beta](w[V/x])]\!] \\ &\equiv [\![([\beta]w)[V/x]]\!] \ . \end{aligned}$$

(3) By induction on V, v and τ.

Case of x : we obtain

$$\begin{aligned} x^*[t/\alpha] &\equiv x[t/\alpha] \equiv x \equiv x^* \\ [\![x]\!]_u[t/\alpha] &\equiv (u \bullet ?x^*)[t/\alpha] \equiv u[t/\alpha] \bullet ?(x^*[t/\alpha]) \\ &\equiv u[t/\alpha] \bullet ?x^* \equiv [\![x]\!]_{u[t/\alpha]} \ . \end{aligned}$$

Case of $\lambda z.w$: we obtain

$$\begin{aligned} (\lambda z.w)^*[t/\alpha] &\equiv !(z,\gamma)[\![w]\!]_\gamma[t/\alpha] \overset{I.H.}{\equiv} !(z,\gamma)[\![w]\!]_\gamma \equiv (\lambda z.w)^* \\ [\![\lambda z.w]\!]_u\,[t/\alpha] &\equiv (u \bullet ?(\lambda z.w)^*)[t/\alpha] \equiv u[t/\alpha] \bullet ?(\lambda z.w)^*[t/\alpha] \\ &\equiv u[t/\alpha] \bullet ?(\lambda z.w)^* \equiv [\![\lambda z.w]\!]_{u[t/\alpha]} \ . \end{aligned}$$

Case of Ww (where W is a value): By induction hypothesis, we have

$$(W,\ u)^{\triangleright}[t/\alpha] \equiv (W,\ u[t/\alpha])^{\triangleright}$$

Therefore we obtain

$$\begin{aligned} [\![Ww]\!]_u[t/\alpha] &\equiv [\![w]\!]_{(W,u)^{\triangleright}}[t/\alpha] \overset{I.H.}{\equiv} [\![w]\!]_{(W,u)^{\triangleright}[t/\alpha]} \\ &\equiv [\![w]\!]_{(W,u[t/\alpha])^{\triangleright}} \equiv [\![Ww]\!]_{u[t/\alpha]} \ . \end{aligned}$$

Case of w_1w_2 (where w_1 is not a value): By induction hypothesis, we have

$$(w_2,\ u)^{\triangleleft}[t/\alpha] \equiv (w_2,\ u[t/\alpha])^{\triangleleft} \ .$$

Therefore we obtain

$$\begin{aligned} [\![w_1w_2]\!]_u[t/\alpha] &\equiv [\![w_1]\!]_{(w_2,u)^{\triangleleft}}[t/\alpha] \overset{I.H.}{\equiv} [\![w_1]\!]_{(w_2,u)^{\triangleleft}[t/\alpha]} \\ &\equiv [\![w_1]\!]_{(w_2,u[t/\alpha])^{\triangleleft}} \equiv [\![w_1w_2]\!]_{u[t/\alpha]} \ . \end{aligned}$$

Case of $\mu\gamma.\tau$: we obtain

$$\begin{aligned}\llbracket \mu\gamma.\tau \rrbracket_u[t/\alpha] &\equiv \llbracket \tau \rrbracket[u/\gamma][t/\alpha] \equiv \llbracket \tau \rrbracket\,[t/\alpha][u[t/\alpha]/\gamma] \\ &\overset{I.H.}{\equiv} \llbracket \tau \rrbracket\,[u[t/\alpha]/\gamma] \equiv \llbracket \mu\gamma.\tau \rrbracket_{u[t/\alpha]} \quad .\end{aligned}$$

Case of $[\gamma]w$ (note that $\gamma \not\equiv \alpha$ by the hypothesis): we obtain

$$\llbracket\, [\gamma]w \,\rrbracket[t/\alpha] \equiv \llbracket w \rrbracket_\gamma[t/\alpha] \overset{I.H.}{\equiv} \llbracket w \rrbracket_\gamma \equiv \llbracket\, [\gamma]w \,\rrbracket \quad .$$

(4) By inducion on V, v and τ.

Case of x : we obtain

$$\begin{aligned}x^*[(w,\beta)^{\triangleleft}/\alpha] &\equiv x[(w,\beta)^{\triangleleft}/\alpha] \equiv x \equiv x^* \\ \llbracket x \rrbracket_t[(w,\beta)^{\triangleleft}/\alpha] &\equiv (t \bullet ?x^*)[(w,\beta)^{\triangleleft}/\alpha] \equiv t[(w,\beta)^{\triangleleft}/\alpha] \bullet ?(x^*[(w,\beta)^{\triangleleft}/\alpha]) \\ &\equiv t[(w,\beta)^{\triangleleft}/\alpha] \bullet ?x^* \equiv \llbracket x \rrbracket_{t[(w,\beta)^{\triangleleft}/\alpha]} \quad .\end{aligned}$$

Case of $\lambda z.v$: we obtain

$$\begin{aligned}(\lambda z.v)^*[(w,\beta)^{\triangleleft}/\alpha] &\equiv !(z,\gamma)\llbracket v \rrbracket_\gamma[(w,\beta)^{\triangleleft}/\alpha] \overset{I.H.}{\longrightarrow^*} !(z,\gamma)\llbracket v[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_\gamma \\ &\equiv (\lambda z.v[{}^{[\beta](-)w}/{}_{[\alpha](-)}])^* \equiv ((\lambda z.v)[{}^{[\beta](-)w}/{}_{[\alpha](-)}])^* \\ \llbracket \lambda z.v \rrbracket_t[(w,\beta)^{\triangleleft}/\alpha] &\equiv (t \bullet ?(\lambda z.v)^*)[(w,\beta)^{\triangleleft}/\alpha] \\ &\equiv t[(w,\beta)^{\triangleleft}/\alpha] \bullet ?(\lambda z.v)^*[(w,\beta)^{\triangleleft}/\alpha] \\ &\longrightarrow^* t[(w,\beta)^{\triangleleft}/\alpha] \bullet ?(\lambda z.v[{}^{[\beta](-)w}/{}_{[\alpha](-)}])^* \\ &\equiv \llbracket (\lambda z.v)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\triangleleft}/\alpha]} \quad .\end{aligned}$$

Case of Ww (where W is a value): By induction hypothesis, we have

$$(W,\, t)^{\triangleright}[(w,\beta)^{\triangleleft}/\alpha] \longrightarrow^* (W[{}^{[\beta](-)w}/{}_{[\alpha](-)}],\, t[(w,\beta)^{\triangleleft}/\alpha])^{\triangleright} \quad .$$

Therefore we obtain

$$\begin{aligned}\llbracket Ww \rrbracket_t[(w,\beta)^{\triangleleft}/\alpha] &\equiv \llbracket w \rrbracket_{(W,t)^{\triangleright}}[(w,\beta)^{\triangleleft}/\alpha] \overset{I.H.}{\longrightarrow^*} \llbracket w[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{(W,t)^{\triangleright}[(w,\beta)^{\triangleleft}/\alpha]} \\ &\overset{I.H.\ and\ (1)}{\longrightarrow^*} \llbracket w[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{(W[{}^{[\beta](-)w}/{}_{[\alpha](-)}],\, t[(w,\beta)^{\triangleleft}/\alpha])^{\triangleright}} \\ &\equiv \llbracket W[{}^{[\beta](-)w}/{}_{[\alpha](-)}]w[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\triangleleft}/\alpha]} \\ &\equiv \llbracket (Ww)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\triangleleft}/\alpha]} \quad .\end{aligned}$$

Case of w_1w_2 (where w_1 is not a value): By induction hypothesis, we have

$$(w_2,\, t)^{\triangleleft}[(w,\beta)^{\triangleleft}/\alpha] \longrightarrow^* (w_2[{}^{[\beta](-)w}/{}_{[\alpha](-)}],\, t[(w,\beta)^{\triangleleft}/\alpha])^{\triangleleft} \quad .$$

Therefore we obtain

$$\begin{aligned}
\llbracket w_1 w_2 \rrbracket_t[(w,\beta)^{\lhd}/\alpha] &\equiv \llbracket w_1 \rrbracket_{(w_2,\,t)^{\lhd}}[(w,\beta)^{\lhd}/\alpha] \xrightarrow{I.H.}{}^{*} \llbracket w_1[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{(w_2,\,t)^{\lhd}[(w,\beta)^{\lhd}/\alpha]} \\
&\xrightarrow{I.H.\ and\ (1)}{}^{*} \llbracket w_1[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{(w_2[{}^{[\beta](-)w}/{}_{[\alpha](-)}],\, t[(w,\beta)^{\lhd}/\alpha])^{\lhd}} \\
&\equiv \llbracket w_1[{}^{[\beta](-)w}/{}_{[\alpha](-)}] w_2[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\lhd}/\alpha]} \\
&\equiv \llbracket (w_1 w_2)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\lhd}/\alpha]} \quad .
\end{aligned}$$

Case of $\mu\gamma.\tau$: we obtain

$$\begin{aligned}
\llbracket \mu\gamma.\tau \rrbracket_t[(w,\beta)^{\lhd}/\alpha] &\equiv \llbracket \tau \rrbracket[t/\gamma][(w,\beta)^{\lhd}/\alpha] \equiv \llbracket \tau \rrbracket[(w,\beta)^{\lhd}/\alpha][t[(w,\beta)^{\lhd}/\alpha]/\gamma] \\
&\xrightarrow{I.H.}{}^{*} \llbracket \tau[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket[t[(w,\beta)^{\lhd}/\alpha]/\gamma] \\
&\equiv \llbracket \mu\gamma.\tau[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\lhd}/\alpha]} \equiv \llbracket (\mu\gamma.\tau)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{t[(w,\beta)^{\lhd}/\alpha]} \quad .
\end{aligned}$$

Case of $[\alpha]v$ (where v is not a value): we obtain

$$\begin{aligned}
\llbracket\, [\alpha]v \,\rrbracket[(w,\beta)^{\lhd}/\alpha] &\equiv \llbracket v \rrbracket_{\alpha}[(w,\beta)^{\lhd}/\alpha] \xrightarrow{I.H.}{}^{*} \llbracket v[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{(w,\beta)^{\lhd}} \\
&\equiv \llbracket v[{}^{[\beta](-)w}/{}_{[\alpha](-)}] w \rrbracket_{\beta} \equiv \llbracket\, [\beta]v[{}^{[\beta](-)w}/{}_{[\alpha](-)}] w \,\rrbracket_{\beta} \equiv \llbracket\, ([\alpha]v)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \,\rrbracket \quad .
\end{aligned}$$

Case of $[\alpha]V$ (where V is a value): we obtain

$$\begin{aligned}
\llbracket\, [\alpha]V \,\rrbracket[(w,\beta)^{\lhd}/\alpha] &\equiv \llbracket V \rrbracket_{\alpha}[(w,\beta)^{\lhd}/\alpha] \xrightarrow{I.H.}{}^{*} \llbracket V[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{(w,\beta)^{\lhd}} \\
&\equiv \llbracket V[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{!x.\llbracket w \rrbracket_{!z.(x \bullet ?(z \otimes \beta))}} \equiv\ !x.\llbracket w \rrbracket_{!z.(x \bullet ?(z \otimes \beta))} \bullet ?(V[{}^{[\beta](-)w}/{}_{[\alpha](-)}])^{*} \\
&\longrightarrow (V[{}^{[\beta](-)w}/{}_{[\alpha](-)}])^{*} \bullet x.\llbracket w \rrbracket_{!z.(x \bullet ?(z \otimes \beta))} \xrightarrow{(3)} \llbracket w \rrbracket_{!z.((V[{}^{[\beta](-)w}/{}_{[\alpha](-)}])^{*} \bullet ?(z \otimes \beta))} \\
&\equiv \llbracket\, [\beta]V[{}^{[\beta](-)w}/{}_{[\alpha](-)}] w \,\rrbracket \equiv \llbracket\, ([\alpha]V)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \,\rrbracket \quad .
\end{aligned}$$

Case of $[\gamma]v$ (where $\gamma \not\equiv \alpha$): we obtain

$$\begin{aligned}
\llbracket\, [\gamma]v \,\rrbracket[(w,\beta)^{\lhd}/\alpha] &\equiv \llbracket v \rrbracket_{\gamma}[(w,\beta)^{\lhd}/\alpha] \xrightarrow{I.H.}{}^{*} \llbracket v[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \rrbracket_{\gamma} \\
&\equiv \llbracket\, [\gamma](v[{}^{[\beta](-)w}/{}_{[\alpha](-)}]) \,\rrbracket \equiv \llbracket\, ([\gamma]v)[{}^{[\beta](-)w}/{}_{[\alpha](-)}] \,\rrbracket \quad .
\end{aligned}$$

(5) By inducion on W, v and τ.

Case of x : we obtain

$$\begin{aligned}
x^{*}[(V,\beta)^{\rhd}/\alpha] &\equiv x[(V,\beta)^{\rhd}/\alpha] \equiv x \equiv x^{*} \\
\llbracket x \rrbracket_t[(V,\beta)^{\rhd}/\alpha] &\equiv (t \bullet ?x^{*})[(V,\beta)^{\rhd}/\alpha] \equiv t[(V,\beta)^{\rhd}/\alpha] \bullet ?(x^{*}[(V,\beta)^{\rhd}/\alpha]) \\
&\equiv t[(V,\beta)^{\rhd}/\alpha] \bullet ?x^{*} \equiv \llbracket x \rrbracket_{t[(V,\beta)^{\rhd}/\alpha]} \quad .
\end{aligned}$$

Case of $\lambda z.v$: we obtain

$$\begin{aligned}
(\lambda z.v)^*[(V,\beta)^{\triangleright}/\alpha] &\equiv\ !(z,\gamma)\llbracket v\rrbracket_\gamma[(V,\beta)^{\triangleright}/\alpha] \overset{I.H.}{\equiv}\ !(z,\gamma)\llbracket v[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_\gamma \\
&\equiv (\lambda z.v[^{[\beta]V(-)}/_{[\alpha](-)}])^* \equiv ((\lambda z.v)[^{[\beta]V(-)}/_{[\alpha](-)}])^* \\
\llbracket \lambda z.v\rrbracket_t[(V,\beta)^{\triangleright}/\alpha] &\equiv (t \bullet ?(\lambda z.v)^*)[(V,\beta)^{\triangleright}/\alpha] \\
&\equiv t[(V,\beta)^{\triangleright}/\alpha] \bullet ?(\lambda z.v)^*[(V,\beta)^{\triangleright}/\alpha] \\
&\equiv t[(V,\beta)^{\triangleright}/\alpha] \bullet ?(\lambda z.v[^{[\beta]V(-)}/_{[\alpha](-)}])^* \\
&\equiv \llbracket(\lambda z.v)[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \quad .
\end{aligned}$$

Case of Ww (where W is a value): By induction hypothesis, we have

$$(W,\, t)^{\triangleright}[(V,\beta)^{\triangleright}/\alpha] \equiv (W[^{[\beta]V(-)}/_{[\alpha](-)}],\, t[(V,\beta)^{\triangleright}/\alpha])^{\triangleright} \quad .$$

Therefore we obtain

$$\begin{aligned}
\llbracket Ww\rrbracket_t[(V,\beta)^{\triangleright}/\alpha] &\equiv \llbracket w\rrbracket_{(W,\,t)^{\triangleright}}[(V,\beta)^{\triangleright}/\alpha] \\
&\overset{I.H.}{\equiv} \llbracket w[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{(W,\,t)^{\triangleright}[(V,\beta)^{\triangleright}/\alpha]} \\
&\equiv \llbracket w[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{(W[^{[\beta]V(-)}/_{[\alpha](-)}],\, t[(V,\beta)^{\triangleright}/\alpha])^{\triangleright}} \\
&\equiv \llbracket W[^{[\beta]V(-)}/_{[\alpha](-)}]w[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \\
&\equiv \llbracket (Ww)[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \quad .
\end{aligned}$$

Case of w_1w_2 (where w_1 is not a value): By induction hypothesis, we have

$$(w_2,\, t)^{\triangleleft}[(V,\beta)^{\triangleright}/\alpha] \equiv (w_2[^{[\beta]V(-)}/_{[\alpha](-)}],\, t[(V,\beta)^{\triangleright}/\alpha])^{\triangleleft} \quad .$$

Therefore we obtain

$$\begin{aligned}
\llbracket w_1w_2\rrbracket_t[(V,\beta)^{\triangleright}/\alpha] &\equiv \llbracket w_1\rrbracket_{(w_2,\,t)^{\triangleleft}}[(V,\beta)^{\triangleright}/\alpha] \\
&\overset{I.H.}{\equiv} \llbracket w_1[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{(w_2,\,t)^{\triangleleft}[(V,\beta)^{\triangleright}/\alpha]} \\
&\equiv \llbracket w_1[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{(w_2[^{[\beta]V(-)}/_{[\alpha](-)}],\, t[(V,\beta)^{\triangleright}/\alpha])^{\triangleleft}} \\
&\equiv \llbracket w_1[^{[\beta]V(-)}/_{[\alpha](-)}]w_2[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \\
&\equiv \llbracket (w_1w_2)[^{[\beta]V(-)}/_{[\alpha](-)}]\rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \quad .
\end{aligned}$$

Case of $\mu\gamma.\tau$: we obtain

$$\llbracket \mu\gamma.\tau\rrbracket_t[(V,\beta)^{\triangleright}/\alpha] \equiv \llbracket \tau\rrbracket[t/\gamma][(V,\beta)^{\triangleright}/\alpha]$$

$$
\begin{aligned}
&\equiv \llbracket \tau \rrbracket [(V,\beta)^{\triangleright}/\alpha][t[(V,\beta)^{\triangleright}/\alpha]/\gamma] \\
&\overset{I.H.}{\equiv} \llbracket \tau[{}^{[\beta]V(-)}/_{[\alpha](-)}] \rrbracket [t[(V,\beta)^{\triangleright}/\alpha]/\gamma] \\
&\equiv \llbracket \mu\gamma.\tau[{}^{[\beta]V(-)}/_{[\alpha](-)}] \rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \\
&\equiv \llbracket (\mu\gamma.\tau)[{}^{[\beta]V(-)}/_{[\alpha](-)}] \rrbracket_{t[(V,\beta)^{\triangleright}/\alpha]} \quad .
\end{aligned}
$$

Case of $[\alpha]w$: we obtain

$$
\begin{aligned}
\llbracket\, [\alpha]w \,\rrbracket[(V,\beta)^{\triangleright}/\alpha] &\equiv \llbracket w \rrbracket_{\alpha}[(V,\beta)^{\triangleright}/\alpha] \overset{I.H.}{\equiv} \llbracket w[{}^{[\beta]V(-)}/_{[\alpha](-)}] \rrbracket_{(V,\beta)^{\triangleright}} \\
&\equiv \llbracket Vw[{}^{[\beta]V(-)}/_{[\alpha](-)}] \rrbracket_{\beta} \equiv \llbracket\, [\beta]Vw[{}^{[\beta]V(-)}/_{[\alpha](-)}] \,\rrbracket \\
&\equiv \llbracket\, ([\alpha]w)[{}^{[\beta]V(-)}/_{[\alpha](-)}] \,\rrbracket \quad .
\end{aligned}
$$

Case of $[\gamma]w$ (where $\gamma \not\equiv \alpha$): we obtain

$$
\begin{aligned}
\llbracket\, [\gamma]w \,\rrbracket[(V,\beta)^{\triangleright}/\alpha] &\equiv \llbracket w \rrbracket_{\gamma}[(V,\beta)^{\triangleright}/\alpha] \overset{I.H.}{\equiv} \llbracket w[{}^{[\beta]V(-)}/_{[\alpha](-)}] \rrbracket_{\gamma} \\
&\equiv \llbracket\, [\gamma](w[{}^{[\beta]V(-)}/_{[\alpha](-)}]) \,\rrbracket \equiv \llbracket\, ([\gamma]w)[{}^{[\beta]V(-)}/_{[\alpha](-)}] \,\rrbracket \quad .
\end{aligned}
$$

(6) By induction on V, w and τ.

Case of x :

$$
x^{*}[\beta/\alpha] \equiv x[\beta/\alpha] \equiv x \equiv x^{*}
$$

$$
\llbracket x \rrbracket_{t}[\beta/\alpha] \equiv (t \bullet ?x^{*})[\beta/\alpha] \equiv t[\beta/\alpha] \bullet ?(x^{*}[\beta/\alpha]) \equiv t[\beta/\alpha] \bullet ?x^{*} \equiv \llbracket x \rrbracket_{t[\beta/\alpha]}
$$

Case of $\lambda z.w$:

$$
\begin{aligned}
(\lambda z.w)^{*}[\beta/\alpha] &\equiv\ !(z,\gamma)\llbracket w \rrbracket_{\gamma}\,[\beta/\alpha] \overset{I.H.}{\equiv}\ !(z,\gamma)\llbracket w[\beta/\alpha] \rrbracket_{\gamma} \\
&\equiv (\lambda z.w[\beta/\alpha])^{*} \equiv ((\lambda z.w)[\beta/\alpha])^{*} \\
\llbracket \lambda z.w \rrbracket_{t}[\beta/\alpha] &\equiv (t \bullet ?(\lambda z.w)^{*})[\beta/\alpha] \\
&\equiv t[\beta/\alpha] \bullet ?(\lambda z.w)^{*}[\beta/\alpha] \equiv t[\beta/\alpha] \bullet ?(\lambda z.w[\beta/\alpha])^{*} \\
&\equiv \llbracket \lambda z.w[\beta/\alpha] \rrbracket_{t[\beta/\alpha]} \equiv \llbracket (\lambda z.w)[\beta/\alpha] \rrbracket_{t[\beta/\alpha]}
\end{aligned}
$$

Case of Ww (where W is a value): By induction hypothesis, we have

$$
(W,\, t)^{\triangleright}[\beta/\alpha] \equiv (W[\beta/\alpha],\, t[\beta/\alpha])^{\triangleright} \quad .
$$

Therefore we obtain

$$
\llbracket Ww \rrbracket_{t}[\beta/\alpha] \equiv \llbracket w \rrbracket_{(W,t)^{\triangleright}}[\beta/\alpha] \overset{I.H.}{\equiv} \llbracket w[\beta/\alpha] \rrbracket_{(W,t)^{\triangleright}[\beta/\alpha]} \equiv \llbracket w[\beta/\alpha] \rrbracket_{(W[\beta/\alpha],\, t[\beta/\alpha])^{\triangleright}}
$$

$$\equiv [\![W[\beta/\alpha]\, w[\beta/\alpha]]\!]_{t[\beta/\alpha]} \equiv [\![(Ww)[\beta/\alpha]]\!]_{t[\beta/\alpha]} \;\; .$$

Case of $w_1 w_2$ (where w_1 is not a value): By induction hypothesis, we have

$$(w_2,\, t)^{\triangleleft}[\beta/\alpha] \equiv (w_2[\beta/\alpha],\, t[\beta/\alpha])^{\triangleleft} \;\; .$$

Therefore we obtain

$$\begin{aligned}
[\![w_1 w_2]\!]_t[\beta/\alpha] \equiv [\![w_1]\!]_{(w_2,\, t)^{\triangleleft}}[\beta/\alpha] &\overset{I.H.}{\equiv} [\![w_1[\beta/\alpha]]\!]_{(w_2,\, t)^{\triangleleft}[\beta/\alpha]} \\
&\equiv [\![w_1[\beta/\alpha]]\!]_{(w_2[\beta/\alpha],\, t[\beta/\alpha])^{\triangleleft}} \\
&\equiv [\![w_1[\beta/\alpha] w_2[\beta/\alpha]]\!]_{t[\beta/\alpha]} \equiv [\![(w_1 w_2)[\beta/\alpha]]\!]_{t[\beta/\alpha]} \;\; .
\end{aligned}$$

Case of $\mu\gamma.\tau$:

$$\begin{aligned}
[\![\mu\gamma.\tau]\!]_t[\beta/\alpha] &\equiv [\![\tau]\!][t/\gamma][\beta/\alpha] \equiv [\![\tau]\!][\beta/\alpha][t[\beta/\alpha]/\gamma] \\
&\overset{I.H.}{\equiv} [\![\tau[\beta/\alpha]]\!][t[\beta/\alpha]/\gamma] \\
&\equiv [\![\mu\gamma.\tau[\beta/\alpha]]\!]_{t[\beta/\alpha]} \equiv [\![(\mu\gamma.\tau)[\beta/\alpha]]\!]_{t[\beta/\alpha]}
\end{aligned}$$

Case of $[\alpha]w$:

$$[\![[\alpha]w]\!][\beta/\alpha] \equiv [\![w]\!]_{\alpha}[\beta/\alpha] \overset{I.H.}{\equiv} [\![w[\beta/\alpha]]\!]_{\beta} \equiv [\![[\beta](w[\beta/\alpha])]\!] \equiv [\![([\alpha]w)[\beta/\alpha]]\!]$$

Case of $[\gamma]w$ (where $\gamma \not\equiv \alpha$) :

$$[\![[\gamma]w]\!][\beta/\alpha] \equiv [\![w]\!]_{\gamma}[\beta/\alpha] \overset{I.H.}{\equiv} [\![w[\beta/\alpha]]\!]_{\gamma} \equiv [\![[\gamma](w[\beta/\alpha])]\!] \equiv [\![([\gamma]w)[\beta/\alpha]]\!]$$

$\square$

Theorem 3.7 (soundness of the positive-translation)
If $w \longrightarrow_v v$ then $[\![w]\!]_t \longrightarrow^* [\![v]\!]_t$ holds for any positive term t.

Proof. By induction on $\longrightarrow_v$.
Base step :

$$\begin{aligned}
[\![(\lambda x.w)V]\!]_t \equiv [\![V]\!]_{!z.((\lambda x.w)^* \bullet ?(z \otimes t))} &\equiv !z.((\lambda x.w)^* \bullet ?(x \otimes t)) \bullet ?V^* \\
&\longrightarrow V^* \bullet z.((\lambda x.w)^* \bullet ?(z \otimes t)) \\
&\longrightarrow (\lambda x.w)^* \bullet ?(V^* \otimes t) \equiv !(x, \alpha)[\![w]\!]_{\alpha} \bullet ?(V^* \otimes t) \\
&\longrightarrow (V^* \otimes t) \bullet (x, \alpha)[\![w]\!]_{\alpha}
\end{aligned}$$

$$\begin{aligned}
&\longrightarrow^* [\![w]\!]_\alpha[V^*/x, t/\alpha] \overset{Lem\ 3.6\ (1),(3)}{\equiv} [\![w[V/x]]\!]_t \\
[\![(\mu\alpha.\tau)w]\!]_t &\equiv [\![\mu\alpha.\tau]\!]_{(w,t)^\triangleleft} \equiv [\![\tau]\!][(w,t)^\triangleleft/\alpha] \equiv [\![\tau]\!][(w,\beta)^\triangleleft/\alpha][t/\beta] \qquad (\beta \text{ is fresh}) \\
&\overset{Lem\ 3.6\ (4)}{\longrightarrow^*} [\![\ \tau[^{[\beta](-)w}/_{[\alpha](-)}]\ ]\!][t/\beta] \equiv [\![\ \mu\beta.\tau[^{[\beta](-)w}/_{[\alpha](-)}]\ ]\!]_t \\
[\![V(\mu\alpha.\tau)]\!]_t &\equiv [\![\mu\alpha.\tau]\!]_{(V,t)^\triangleright} \equiv [\![\tau]\!][(V,t)^\triangleright/\alpha] \equiv [\![\tau]\!][(V,t)^\triangleright/\alpha][t/\beta] \\
&\overset{Lem\ 3.6\ (5)}{\equiv} [\![\ \tau[^{[\beta]V(-)}/_{[\alpha](-)}]\ ]\!][t/\beta] \equiv [\![\ \mu\beta.\tau[^{[\beta]V(-)}/_{[\alpha](-)}]\ ]\!]_t \\
[\![\mu\alpha.[\alpha]w]\!]_t &\equiv [\![[\alpha]w]\!][t/\alpha] \equiv [\![w]\!]_\alpha[t/\alpha] \overset{Lem\ 3.6\ (3)}{\equiv} [\![w]\!]_t \\
[\![[\beta]\mu\alpha.\tau]\!] &\equiv [\![\mu\alpha.\tau]\!]_\beta \equiv [\![\tau]\!][\beta/\alpha] \overset{Lem\ 3.6\ (6)}{\equiv} [\![\tau[\beta/\alpha]]\!]
\end{aligned}$$

Inductive step :
Case of $\lambda x.w \longrightarrow_v \lambda x.w'$ (obtained from $w \longrightarrow_v w'$) :

$$[\![\lambda x.w]\!]_t \equiv t \bullet ?!(x,\alpha)[\![w]\!]_\alpha \overset{I.H.}{\longrightarrow^*} t \bullet ?!(x,\alpha)[\![w']\!]_\alpha \equiv [\![\lambda x.w']\!]_t$$

Case of $vw \longrightarrow_v v'w$ (obtained from $v \longrightarrow_v v'$, and v is not a value) :

$$[\![vw]\!]_t \equiv [\![v]\!]_{(w,\,t)^\triangleleft} \overset{I.H.}{\longrightarrow^*} [\![v']\!]_{(w,\,t)^\triangleleft} \equiv [\![v'w]\!]_t$$

Case of $vw \longrightarrow_v vw'$ (obtained from $w \longrightarrow_v w'$, and v is not a value) : by the induction hypothesis, we can obtain $(w,\ t)^\triangleleft \longrightarrow^* (w',\ t)^\triangleleft$. Using Lemma 3.6 (1), we obtain the conclusion of this case in the following way.

$$[\![vw]\!]_t \equiv [\![v]\!]_{(w,\,t)^\triangleleft} \longrightarrow^* [\![v]\!]_{(w'\ t)^\triangleleft} \equiv [\![vw']\!]_t$$

Case of $Vw \longrightarrow_v V'w$ (obtained from $V \longrightarrow_v V'$, and V is a value) : by the induction hypothesis, we can obtain $(V,\ t)^\triangleright \longrightarrow^* (V',\ t)^\triangleright$. Using Lemma 3.6 (1), we obtain the conclusion of this case in the following way.

$$[\![Vw]\!]_t \equiv [\![w]\!]_{(V,\,t)^\triangleright} \longrightarrow^* [\![w]\!]_{(V',\,t)^\triangleright} \equiv [\![V'w]\!]_t$$

Case of $Vw \longrightarrow_v Vw'$ (obtained from $w \longrightarrow_v w'$, and V is a value) :

$$[\![Vw]\!]_t \equiv [\![w]\!]_{(V,\,t)^\triangleright} \overset{I.H.}{\longrightarrow^*} [\![w']\!]_{(V,\,t)^\triangleright} \equiv [\![Vw']\!]_t$$

Case of $\mu\alpha.\tau \longrightarrow_v \mu\alpha.\tau'$ (obtained from $\tau \longrightarrow_v \tau'$) :

$$[\![\mu\alpha.\tau]\!]_t \equiv [\![\tau]\!][t/\alpha] \overset{I.H.}{\longrightarrow^*} [\![\tau']\!][t/\alpha] \equiv [\![\mu\alpha.\tau']\!]_t$$

Case of $[\alpha]w \longrightarrow_v [\alpha]w'$ (obtained from $w \longrightarrow_v w'$) :

$$[\![[\alpha]w]\!] \equiv [\![w]\!]_\alpha \overset{I.H.}{\longrightarrow^*} [\![w']\!]_\alpha \equiv [\![[\alpha]w']\!]$$

□

Remark 5

Similarly to the call-by-name case, our translation does not preserve the call-by-value η-rule of the $\lambda\mu$-calculus (*i.e.* $\lambda x.Vx \longrightarrow_v V$ (where $x \notin \mathrm{FV}(V)$)) as the reduction rule, but it does preserve it as the equational rule. When x and α are fresh for t, we can prove $!(x,\alpha)(t \bullet ?(x \otimes \alpha)) = t$ using the η-rules of DCP$^-$ as the following way: $!(x,\alpha)(t \bullet ?(x \otimes \alpha)) = !(x,\alpha)(!z.(t \bullet ?z) \bullet ?(x \otimes \alpha)) \longrightarrow !(x,\alpha)(x \otimes \alpha \bullet z.(t \bullet ?z)) \longrightarrow !z.(t \bullet ?z) \longrightarrow t$. So, we obtain $(\lambda x.Vx)^* = V^*$ by

$$\begin{aligned}(\lambda x.Vx)^* &\equiv !(x,\alpha)[\![Vx]\!]_\alpha \equiv !(x,\alpha)[\![x]\!]_{!z.(V^* \bullet ?(z \otimes \alpha))} \equiv !(x,\alpha)(!z.(V^* \bullet ?(z \otimes \alpha)) \bullet ?x) \\ &\longrightarrow !(x,\alpha)(x \bullet z.(V^* \bullet ?(z \otimes \alpha))) \longrightarrow !(x,\alpha)(V^* \bullet ?(x \otimes \alpha)) = V^*.\end{aligned}$$

3.5 Logical Predicates and Basic Lemma

In the following, we only consider the $\otimes, ⅋, !, ?$-fragment of DCP$^-$ for simplicity, and still call this fragment DCP$^-$. In this section, we develop the logical predicate method for LLP, and then prove the Basic Lemma that works for both the negative- and the positive-translations *uniformly*. We denote λ- (resp. μ-) contexts of the $\lambda\mu$-calculus by Γ (resp. Δ), and contexts of DCP$^-$ by Σ and Λ.

We define $[\Gamma;\Delta]^\circ$ by $?(\Gamma^\circ)^\perp, \Delta^\circ$, and $[\Gamma;\Delta]^\bullet$ by $(\Gamma^\bullet)^\perp, ?(\Delta^\bullet)$. In the following, † stands for either $\circ$ or $\bullet$. Note that $[\Gamma;\Delta]^\dagger$ is always a context of DCP$^-$.

Definition 3.14

For any positive type P and negative type N, and a special symbol $\perp$ define

$$\begin{aligned}\mathfrak{D}^\dagger_P(\Gamma;\Delta) &:= \{\, t : \text{DCP}^-\text{-pos.term} \mid \vdash [\Gamma;\Delta]^\dagger \,;\, t : P \,\},\\ \mathfrak{D}^\dagger_N(\Gamma;\Delta) &:= \{\, k : \text{DCP}^-\text{-neg.term} \mid \vdash [\Gamma;\Delta]^\dagger \,;\, k : N \,\}, \textit{and}\\ \mathfrak{D}^\dagger_\perp(\Gamma;\Delta) &:= \{\, \tau : \text{DCP}^-\text{-neut.term} \mid \vdash [\Gamma;\Delta]^\dagger \,;\, \tau \,\}.\end{aligned}$$

Definition 3.15

For any context Γ, Δ and type A of the $\lambda\mu$-calculus, define

$$\Lambda^\mu_A(\Gamma;\Delta) := \{\, w : \lambda\mu\text{-term} \mid \Gamma \vdash \Delta \mid w : A \,\}, \textit{and}$$

$$\Lambda^{\mu}(\Gamma;\Delta) := \{\, \tau : \lambda\mu\text{-named term} \mid \Gamma \vdash \Delta \mid \tau \,\}.$$

Definition 3.16 ($\mathfrak{D}^{\dagger}$-predicate)
Let ξ be a positive type (resp. negative type, $\perp$), π and π' be positive (resp. negative, neutral) terms. A family $\mathcal{S}$ of sets indexed by λ- and μ-contexts of the $\lambda\mu$-calculus is called a $\mathfrak{D}^{\dagger}$-*predicate on* ξ when $\mathcal{S}(\Gamma;\Delta) \subset \mathfrak{D}^{\dagger}_{\xi}(\Gamma;\Delta)$ and

(monotonicity) if $\Gamma \subset \Gamma'$ and $\Delta \subset \Delta'$ then $\mathcal{S}(\Gamma;\Delta) \subset \mathcal{S}(\Gamma';\Delta')$, and

(equality) if $\pi \in \mathcal{S}(\Gamma;\Delta)$, $\pi' \in \mathfrak{D}^{\dagger}_{\xi}(\Gamma;\Delta)$ and $\pi = \pi'$ then $\pi' \in \mathcal{S}(\Gamma;\Delta)$.

Let $\mathcal{S}$, $\mathcal{T}$ be $\mathfrak{D}^{\dagger}$-predicates on ξ. The relation $\mathcal{S} \subset \mathcal{T}$ is defined by $\forall\Gamma,\Delta\,(\, \mathcal{S}(\Gamma;\Delta) \subset \mathcal{T}(\Gamma;\Delta)\,)$

In the sequel, we fix a $\mathfrak{D}^{\dagger}$-predicate $\mathcal{B}$ on $\perp$. In terms of this $\mathcal{B}$, negation is defined as follows.

Definition 3.17 (negation)
Let $\mathcal{S}$ be a family of sets indexed by the λ- and μ-contexts of the $\lambda\mu$-calculus and $\mathcal{S}(\Gamma;\Delta) \subset \mathfrak{D}^{\dagger}_{P}(\Gamma;\Delta)$. We define as follows.

$$\mathcal{S}^{\perp}(\Gamma;\Delta) := \{\, k \in \mathfrak{D}^{\dagger}_{P^{\perp}}(\Gamma;\Delta) \mid \forall\Gamma' \supset \Gamma\, \forall\Delta' \supset \Delta\, \forall t \in \mathcal{S}(\Gamma';\Delta')\, (t \bullet k \in \mathcal{B}(\Gamma';\Delta'))\,\}.$$

We define $\mathcal{S}^{\perp}$ for $\mathcal{S}$ such that $\mathcal{S}(\Gamma;\Delta) \subset \mathfrak{D}^{\dagger}_{N}(\Gamma;\Delta)$ similarly.

Lemma 3.8
Let $\mathcal{S}$ be an indexed family as above and $\mathcal{S}(\Gamma;\Delta) \subset \mathfrak{D}^{\dagger}_{\xi}(\Gamma;\Delta)$. Then $\mathcal{S}^{\perp}$ is a $\mathfrak{D}^{\dagger}$-predicate on $\xi^{\perp}$.

Proof. We show that $\mathcal{S}^{\perp}$ satisfies the (**monotonicity**) and (**equality**) conditions. Here, we consider the case that ξ is a negative type (for the positive type case, it is similarly proved as this case).
(**monotonicity**): Assume $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $t \in \mathcal{S}^{\perp}(\Gamma;\Delta)$, then for any $\Gamma'' \supset \Gamma'$, $\Delta'' \supset \Delta'$ and $k \in \mathcal{S}(\Gamma'';\Delta'')$, we have $t \bullet k \in \mathcal{B}(\Gamma'';\Delta'')$ by the definition of negation. Hence, we obtain $t \in \mathcal{S}^{\perp}(\Gamma';\Delta')$.
(**equality**): Assume $t \in \mathcal{S}^{\perp}(\Gamma;\Delta)$ and $t = t'$, then for any $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $k \in \mathcal{S}(\Gamma';\Delta')$, we have $t' \bullet k = t \bullet k \in \mathcal{B}(\Gamma';\Delta')$. Since $\mathcal{B}$ satisfies (**equality**), we obtain $t' \bullet k \in \mathcal{B}(\Gamma'';\Delta'')$. Therefore, we conclude $t' \in \mathcal{S}^{\perp}(\Gamma;\Delta)$. □

Lemma 3.9
Let $\mathcal{S}$ and $\mathcal{T}$ be $\mathfrak{D}^\dagger$-predicates on ξ. Then we have

(1) $\mathcal{S} \subset \mathcal{S}^{\perp\perp}$,

(2) If $\mathcal{S} \subset \mathcal{T}$ then $\mathcal{T}^\perp \subset \mathcal{S}^\perp$, and

(3) $\mathcal{S}^\perp = \mathcal{S}^{\perp\perp\perp}$.

Proof. We consider the case that ξ is a negative type (for the positive type case, it is similarly proved as this case).

(1) Suppose $k \in \mathcal{S}(\Gamma; \Delta)$. For any $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $t \in \mathcal{S}^\perp(\Gamma'; \Delta')$, we have $t \bullet k \in \mathcal{B}(\Gamma'; \Delta')$ since $k \in \mathcal{S}(\Gamma'; \Delta')$. Therefore, we obtain $k \in \mathcal{S}^{\perp\perp}(\Gamma; \Delta)$.

(2) Suppose $\mathcal{S} \subset \mathcal{T}$ and $t \in \mathcal{T}^\perp(\Gamma; \Delta)$. For any $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $k \in \mathcal{S}(\Gamma'; \Delta')$, we have $t \bullet k \in \mathcal{B}(\Gamma'; \Delta')$ since $k \in \mathcal{T}(\Gamma'; \Delta')$. Therefore, we obtain $t \in \mathcal{S}^\perp(\Gamma; \Delta)$.

(3) It is immediate from (1) and (2).

□

Definition 3.18
Suppose $\mathcal{P}$, $\mathcal{Q}$ are $\mathfrak{D}^\dagger$-predicates on P, Q, and $\mathcal{N}$, $\mathcal{M}$ are $\mathfrak{D}^\dagger$-predicates on N, M respectively. We then define as follows.

$$\langle \mathcal{P}, \mathcal{Q} \rangle(\Gamma; \Delta) := \{\, t \otimes u \mid t \in \mathcal{P}(\Gamma; \Delta),\ u \in \mathcal{Q}(\Gamma; \Delta) \,\}$$
$$\mathcal{P}^?(\Gamma; \Delta) := \{\, ?t \mid t \in \mathcal{P}(\Gamma; \Delta) \,\}$$
$$\mathcal{P} \otimes \mathcal{Q} := \langle \mathcal{P}, \mathcal{Q} \rangle^{\perp\perp}, \qquad \mathcal{N} \parr \mathcal{M} := \langle \mathcal{N}^\perp, \mathcal{M}^\perp \rangle^\perp$$
$$?\mathcal{P} := (\mathcal{P}^?)^{\perp\perp}, \qquad !\mathcal{N} := ((\mathcal{N}^\perp)^?)^\perp$$

Lemma 3.10
Let $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{N}$ and $\mathcal{M}$ be $\mathfrak{D}^\dagger$-predicates on P, Q, N and M respectively. Then $\mathcal{P} \otimes \mathcal{Q}$, $?\mathcal{P}$, $\mathcal{N} \parr \mathcal{M}$ and $!\mathcal{N}$ are $\mathfrak{D}^\dagger$-predicate on $P \otimes Q$, $?P$, $N \parr M$ and $!N$ respectively.

Proof. It immediately follows from Lemma 3.8. □

Definition 3.19 (logical $\mathfrak{D}^\dagger$-predicate)
Let ξ range over the types of DCP$^-$ and $\perp$. A family $\{\mathcal{S}_\xi\}$ of $\mathfrak{D}^\dagger$-predicates is called a *logical $\mathfrak{D}^\dagger$-predicate* when the following conditions hold:

- each $\mathcal{S}_\xi$ is a $\mathfrak{D}^\dagger$-predicate on ξ.
- $\mathcal{S}_{X^\perp} = {\mathcal{S}_X}^\perp$, $\mathcal{S}_X = {\mathcal{S}_{X^\perp}}^\perp$ and $\mathcal{S}_\perp = \mathcal{B}$.
- $\mathcal{S}_{P\otimes Q} = \mathcal{S}_P \otimes \mathcal{S}_Q$, $\mathcal{S}_{N \parr M} = \mathcal{S}_N \parr \mathcal{S}_M$, $\mathcal{S}_{!N} = !\mathcal{S}_N$, and $\mathcal{S}_{?P} = ?\mathcal{S}_P$.

Lemma 3.11
If $\{\mathcal{S}_\xi\}$ is a logical $\mathfrak{D}^\dagger$-predicate, then $(\mathcal{S}_\xi)^\perp = \mathcal{S}_{\xi^\perp}$.

Proof. By induction on ξ.
If ξ is a atomic type X or $X^\perp$, it is immediate.
If ξ is $P \otimes Q$, we have

$$\begin{aligned}(\mathcal{S}_{P\otimes Q})^\perp &= (\mathcal{S}_P \otimes \mathcal{S}_Q)^\perp = \langle \mathcal{S}_P, \mathcal{S}_Q\rangle^{\perp\perp\perp} \overset{Lem\ 3.9\ (3)}{=} \langle \mathcal{S}_P, \mathcal{S}_Q\rangle^\perp \\ &\overset{I.H.}{=} \langle \mathcal{S}^\perp_{P^\perp}, \mathcal{S}^\perp_{Q^\perp}\rangle^\perp = \mathcal{S}_{P^\perp} \parr \mathcal{S}_{Q^\perp} = \mathcal{S}_{P^\perp \parr Q^\perp} = \mathcal{S}_{(P\otimes Q)^\perp} \quad .\end{aligned}$$

If ξ is $N \parr M$, we have

$$\begin{aligned}(\mathcal{S}_{N\parr M})^\perp &= (\mathcal{S}_N \parr \mathcal{S}_M)^\perp = \langle \mathcal{S}^\perp_N, \mathcal{S}^\perp_M\rangle^{\perp\perp} \overset{I.H.}{=} \langle \mathcal{S}_{N^\perp}, \mathcal{S}_{M^\perp}\rangle^{\perp\perp} \\ &= \mathcal{S}_{N^\perp} \otimes \mathcal{S}_{M^\perp} = \mathcal{S}_{N^\perp \otimes M^\perp} = \mathcal{S}_{(N\parr M)^\perp} \quad .\end{aligned}$$

If ξ is $?P$, we have

$$\begin{aligned}(\mathcal{S}_{?P})^\perp &= (?\mathcal{S}_P)^\perp = \mathcal{S}_P^{?\perp\perp\perp} \overset{Lem\ 3.9\ (3)}{=} \mathcal{S}_P^{?\perp} \\ &\overset{I.H.}{=} \mathcal{S}_{P^\perp}^{\perp ? \perp} = !\mathcal{S}_{P^\perp} = \mathcal{S}_{!P^\perp} = \mathcal{S}_{(?P)^\perp} \quad .\end{aligned}$$

If ξ is $!N$, we have

$$(\mathcal{S}_{!N})^\perp = (!\mathcal{S}_N)^\perp = \mathcal{S}_N^{\perp ? \perp\perp} \overset{I.H.}{=} \mathcal{S}_{N^\perp}^{?\perp\perp} = ?\mathcal{S}_{N^\perp} = \mathcal{S}_{?N^\perp} \quad .$$

□

Lemma 3.12 (Basic Lemma)
Let $\{\mathcal{S}_\xi\}$ be a logical $\mathfrak{D}^\dagger$-predicate, $\Sigma \equiv x_1 : N_1, \ldots, x_n : N_n$ be a context of DCP$^-$, and Γ, Δ be contexts of the $\lambda\mu$-calculus. For any $s_i \in \mathcal{S}_{N_i^\perp}(\Gamma; \Delta)$ $(1 \leq i \leq n)$, the following hold.

(1) If $\vdash \Sigma\,;\ k : N$, then $k[s_1/x_1, \ldots, s_n/x_n] \in \mathcal{S}_N(\Gamma; \Delta)$.

(2) If $\vdash \Sigma\,;\ u : P$, then $u[s_1/x_1, \ldots, s_n/x_n] \in \mathcal{S}_P(\Gamma; \Delta)$.

(3) If $\vdash \Sigma\,;\ \tau$, then $\tau[s_1/x_1, \ldots, s_n/x_n] \in \mathcal{B}(\Gamma; \Delta)$.

Proof. By induction on the derivation of DCP$^-$. We consider the last rule of the derivation.
Case of (Ax)-rule: it is immediate from the hypothesis.
Case of (Cut)-rule: Suppose that $\vdash \Sigma, \Lambda\ ;\ t \bullet k$ is derived from $\vdash \Sigma\ ;\ t : P$ and $\vdash \Lambda\ ;\ k : P^\perp$ by (Cut)-rule. By induction hypothesis, we have $t[\vec{s}/\vec{x}] \in \mathcal{S}_P(\Gamma;\Delta)$ and $k[\vec{s}/\vec{x}] \in \mathcal{S}_{P^\perp}(\Gamma;\Delta) = \mathcal{S}_P^\perp(\Gamma;\Delta)$. Hence we obtain

$$(t \bullet k)[\vec{s}/\vec{x}] \equiv t[\vec{s}/\vec{x}] \bullet k[\vec{s}/\vec{x}] \in \mathcal{B}(\Gamma;\Delta)\ .$$

Case of $\otimes$-rule: Suppose that $\vdash \Sigma, \Lambda\ ;\ t \otimes u \colon P \otimes Q$ is derived from $\vdash \Sigma\ ;\ t : P$ and $\vdash \Lambda\ ;\ u : Q$ by $\otimes$-rule. By induction hypothesis, we have $t[\vec{s}/\vec{x}] \in \mathcal{S}_P(\Gamma;\Delta)$ and $u[\vec{s}/\vec{x}] \in \mathcal{S}_Q(\Gamma;\Delta)$. Hence we obtain

$$\begin{aligned}(t \otimes u)[\vec{s}/\vec{x}] \equiv t[\vec{s}/\vec{x}] \otimes u[\vec{s}/\vec{x}] &\in \langle \mathcal{S}_P, \mathcal{S}_P\rangle(\Gamma;\Delta) \subset \langle \mathcal{S}_P, \mathcal{S}_P\rangle^{\perp\perp}(\Gamma;\Delta)\\ &= (\mathcal{S}_P \otimes \mathcal{S}_Q)(\Gamma;\Delta) = \mathcal{S}_{P\otimes Q}(\Gamma;\Delta)\end{aligned}$$

Case of $⅋$-rule: Suppose that $\vdash \Sigma\ ;\ (y,z)\tau : N ⅋ M$ is derived from $\vdash \Sigma, y \colon N, z : M\ ;\ \tau$ by $⅋$-rule Let $s_i \in \mathcal{S}_{N_i^\perp}(\Gamma;\Delta)$ $(1 \le i \le n)$, and $\Gamma \subset \Gamma'$, $\Delta \subset \Delta'$ and $t \in \mathcal{S}_{N^\perp}(\Gamma';\Delta')$ and $u \in \mathcal{S}_{M^\perp}(\Gamma';\Delta')$. By induction hypothesis,

$$(t \otimes u) \bullet ((y,z)\tau)[\vec{s}/\vec{x}] \equiv (t \otimes u) \bullet (y,z)(\tau[\vec{s}/\vec{x}]) = \tau[\vec{s}/\vec{x}, t/y, u/z] \in \mathcal{B}(\Gamma';\Delta')\ .$$

Therefore we obtain

$$((y,z)\tau)[\vec{s}/\vec{x}] \in \langle \mathcal{S}_{N^\perp}, \mathcal{S}_{M^\perp}\rangle^\perp(\Gamma;\Delta) = \langle \mathcal{S}_N^\perp, \mathcal{S}_M^\perp\rangle^\perp(\Gamma;\Delta) = (\mathcal{S}_N ⅋ \mathcal{S}_M)(\Gamma;\Delta) = \mathcal{S}_{N⅋M}(\Gamma;\Delta)\ .$$

Case of !-rule: Suppose that $\vdash \Sigma\ ;\ !k : !N$ is derived from $\vdash \Sigma\ ;\ k : N$ by !-rule Let $\Gamma \subset \Gamma'$, $\Delta \subset \Delta'$ and $?t \in (\mathcal{S}_N^\perp)^?(\Gamma';\Delta')$, *i.e.*, $t \in (\mathcal{S}_N^\perp)(\Gamma';\Delta')$. Since $k[\vec{s}/\vec{x}] \in \mathcal{S}_N(\Gamma;\Delta)$ by induction hypothesis,

$$(!k)[\vec{s}/\vec{x}] \bullet ?t \equiv !(k[\vec{s}/\vec{x}]) \bullet ?t = t \bullet (k[\vec{s}/\vec{x}]) \in \mathcal{B}(\Gamma;\Delta)\ .$$

So, we obtain

$$(!k)[\vec{s}/\vec{x}] \in ((\mathcal{S}_N^\perp)^?)^\perp(\Gamma;\Delta) \equiv !\mathcal{S}_N(\Gamma;\Delta) = \mathcal{S}_{!N}(\Gamma;\Delta)\ .$$

Case of the ?-rule: Suppose that $\vdash \Sigma\ ;\ ?t : ?P$ is derived from $\vdash \Sigma\ ;\ t : P$ by ?-rule Since $t[\vec{s}/\vec{x}] \in \mathcal{S}_P(\Gamma;\Delta)$ by induction hypothesis, we obtain

$$(?t)[\vec{s}/\vec{x}] \in \mathcal{S}_P^?(\Gamma;\Delta) \subset (\mathcal{S}_P^?)^{\perp\perp}(\Gamma;\Delta) = ?\mathcal{S}_P(\Gamma;\Delta) = \mathcal{S}_{?P}(\Gamma;\Delta)\ .$$

Case of (Focus)-rule: Suppose that $\vdash \Sigma\ ;\ z.\tau : N$ is derived from $\vdash \Sigma, z : N\ ;\ \tau$ by (Focus)-rule. For any $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $t \in \mathcal{S}_N^\perp(\Gamma';\Delta')$,

$$t \bullet (z.\tau)[\vec{s}/\vec{x}] \equiv t \bullet z.(\tau[\vec{s}/\vec{x}]) = \tau[\vec{s}/\vec{x}, t/z] \in \mathcal{B}(\Gamma';\Delta')$$

by induction hypothesis. So, we obtain

$$(z.\tau)[\vec{s}/\vec{x}] \in \mathcal{S}_N^{\perp\perp}(\Gamma;\Delta) = \mathcal{S}_N(\Gamma;\Delta)\ \ .$$

Case of (Unfocus)-rule: Suppose that $\vdash \Sigma, z : N\ ;\ z \bullet k$ is derived from $\vdash \Sigma\ ;\ k : N$ by (Unfocus)-rule. For any $t \in \mathcal{S}_N^{\perp}(\Gamma;\Delta)$, by induction hypothesis we have

$$(z \bullet k)[\vec{s}/\vec{x}, t/z] \equiv t \bullet k[\vec{s}/\vec{x}] \in \mathscr{B}(\Gamma;\Delta)\ \ .$$

Case of (Weakening)-rule: We now consider the case of positive terms. Suppose that $\vdash \Sigma, z \colon N\ ;\ t \colon P$ is derived from $\vdash \Sigma\ ;\ t \colon P$ by (Weakening)-rule. Since z is a fresh variable, we have the conclusion of this case by induction hypothesis as follows:

$$t[\vec{s}/\vec{x}, s/z] \equiv t[\vec{s}/\vec{x}] \in \mathcal{S}_P(\Gamma;\Delta)\ \ .$$

It is similarly shown the case of negative and neutral terms as this case.
Case of (Contraction)-rule: We now consider the case of positive terms. Suppose that $\vdash \Sigma, z \colon N\ ;\ t[z/x, z/y] \colon P$ is derived from $\vdash \Sigma, x \colon N, y \colon N\ ;\ t \colon P$ by (Contraction)-rule. Then, we have the conclusion of this case by induction hypothesis as follows:

$$(t[z/x, z/y])[\vec{s}/\vec{x}, s/z] \equiv t[\vec{s}/\vec{x}, s/x, s/y] \in \mathcal{S}_P(\Gamma;\Delta)\ \ .$$

It is similarly shown the case of negative and neutral terms as this case. □

3.6 Fullness of the negative-translation

In this section, we discuss only *the negative-translation*, therefore we consider $\overline{[\![w]\!]}$ and $[\![\tau]\!]$ as the images of w and τ by *the negative-translation*.

Definition 3.20
For a type A and contexts Γ, Δ of the $\lambda\mu$-calculus, we define
$\mathscr{B}^\circ(\Gamma;\Delta) := \{\tau \in \mathfrak{D}^\circ_\perp(\Gamma;\Delta) \mid \exists\sigma \in \Lambda^\mu(\Gamma;\Delta)\ (\tau = [\![\sigma]\!])\}$, and
$\mathbb{P}_A(\Gamma;\Delta) := \{\, k \in \mathfrak{D}^\circ_{A^\circ}(\Gamma;\Delta) \mid \exists w \in \Lambda^\mu_A(\Gamma;\Delta)\ (k = \overline{[\![w]\!]})\,\}$.

Lemma 3.13
$\mathscr{B}^\circ$ is a $\mathfrak{D}^\circ$-predicate on $\perp$, and $\mathbb{P}_A$ is a $\mathfrak{D}^\circ$-predicate on A°.

Proof. For each case, we will check the (**monotonicity**) and (**equality**) condition of $\mathfrak{D}^\circ$-predicate.
(**monotonicity**): Assume $\Gamma \subset \Gamma'$, $\Delta \subset \Delta'$. If $\tau \in \mathcal{B}^\circ(\Gamma; \Delta)$, then there is a $\sigma \in \Lambda^\mu(\Gamma; \Delta) \subset \Lambda^\mu(\Gamma'; \Delta')$ such that $\tau = [\![\sigma]\!]$. So, we have $\tau \in \mathcal{B}^\circ(\Gamma'; \Delta')$. Suppose we pick a $k \in \mathbb{P}_A(\Gamma; \Delta)$, then there is a $w \in \Lambda^\mu_A(\Gamma; \Delta) \subset \Lambda^\mu_A(\Gamma'; \Delta')$ such that $k = \overline{[\![w]\!]}$. Hence we have $k \in \mathbb{P}_A(\Gamma'; \Delta')$.
(**equality**): Suppose $\tau \in \mathcal{B}^\circ(\Gamma; \Delta)$ and $\tau = \tau'$, then there is a $\sigma \in \Lambda^\mu(\Gamma; \Delta)$ such that $\tau' = \tau = [\![\sigma]\!]$. Therefore, we have $\tau' \in \mathcal{B}^\circ(\Gamma; \Delta)$. And suppose $k \in \mathbb{P}_A(\Gamma; \Delta)$ and $k = k'$. Then there is a $w \in \Lambda^\mu_A(\Gamma; \Delta)$ such that $k' = k = \overline{[\![w]\!]}$. Hence we have $k' \in \mathbb{P}_A(\Gamma; \Delta)$. □

Lemma 3.14
(i) If $\alpha : A \in \Delta$, then $\alpha \in \mathbb{P}^\perp_A(\Gamma; \Delta)$ for any Γ.
(ii) If $x : A \in \Gamma$, then $x \in ?(\mathbb{P}^\perp_A)^\perp(\Gamma; \Delta)$ for any Δ.

Proof. (i) Suppose $\alpha : A \in \Delta$ and $\Gamma' \supset \Gamma$ and $\Delta' \supset \Delta$, $k \in \mathbb{P}_A(\Gamma'; \Delta')$. There exists $w \in \Lambda^\mu_A(\Gamma'; \Delta')$ such that $k = \overline{[\![w]\!]}$, and then $\alpha \bullet k = \alpha \bullet \overline{[\![w]\!]} = [\![w]\!]_\alpha \equiv [\![[\alpha]w]\!] \in \mathcal{B}^\circ(\Gamma'; \Delta')$. Therefore $\alpha \in \mathbb{P}^\perp_A(\Gamma; \Delta)$.
(ii) Suppose $x : A \in \Gamma$. Notice that $(?\mathbb{P}^\perp_A)^\perp = ((\mathbb{P}^\perp_A)^?)^{\perp\perp\perp} = ((\mathbb{P}^\perp_A)^?)^\perp$. So, we will prove $x \in ((\mathbb{P}^\perp_A)^?)^\perp(\Gamma; \Delta)$. For any $\Gamma' \supset \Gamma$ and $\Delta' \supset \Delta$, $k \in (\mathbb{P}^\perp_A)^?(\Gamma'; \Delta')$. Then there exists $t \in \mathbb{P}^\perp_A(\Gamma'; \Delta')$ such that $k = ?t$. Therefore $x \bullet k = x \bullet ?t \equiv [\![x]\!]_t = t \bullet \overline{[\![x]\!]} \in \mathcal{B}^\circ(\Gamma'; \Delta')$ and so $x \in ((\mathbb{P}^\perp_A)^?)^\perp(\Gamma; \Delta)$. □

Lemma 3.15
$\mathbb{P}^{\perp\perp}_A = \mathbb{P}_A$

Proof. It suffices to show $\mathbb{P}^{\perp\perp}_A \subset \mathbb{P}_A$. Take any Γ, Δ and $k \in \mathbb{P}^{\perp\perp}_A(\Gamma; \Delta)$. We have $\alpha \in \mathbb{P}^\perp_A(\Gamma; \Delta, \alpha : A)$ from the previous lemma, therefore $\alpha \bullet k \in \mathcal{B}^\circ(\Gamma; \Delta, \alpha : A)$. So, there exists a $\lambda\mu$ named-term, say σ, such that $\alpha \bullet k = [\![\sigma]\!]$. Then we obtain $k = \alpha.(\alpha \bullet k) = \alpha.[\![\sigma]\!] \equiv \alpha.([\![\sigma]\!][\alpha/\alpha]) \equiv \alpha.[\![\mu\alpha.\sigma]\!]_\alpha \equiv \overline{[\![\mu\alpha.\sigma]\!]} \in \mathbb{P}_A(\Gamma; \Delta)$. □

Lemma 3.16
Let Γ, Δ be contexts of the $\lambda\mu$-calculus and $t \in (?\mathbb{P}^\perp_A)^\perp(\Gamma; \Delta)$. Then there is a $w \in \Lambda^\mu_A(\Gamma; \Delta)$ such that $t = !\overline{[\![w]\!]}$.

Proof. Suppose $t \in (?\mathbb{P}^\perp_A)^\perp(\Gamma; \Delta)$, then t is a positive term of type $!A^\circ$. So, t is a variable or there is a $k \in \mathfrak{D}^\circ_{A^\circ}(\Gamma; \Delta)$ such that $t \equiv !k$. If t is a variable, say x, then $x : A \in \Gamma$ and $x = !\overline{[\![x]\!]}$.

Otherwise let $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $u \in \mathbb{P}_A^\perp(\Gamma';\Delta')$. Then $?u$ is in $(\mathbb{P}_A^\perp)^?(\Gamma';\Delta')$ and so $u \bullet k = !k \bullet ?u \in \mathcal{B}^\circ(\Gamma';\Delta')$ because of $(?\mathbb{P}_A^\perp)^\perp = ((\mathbb{P}_A^\perp)^?)^\perp$. Then we obtain $k \in \mathbb{P}_A^{\perp\perp}(\Gamma;\Delta) = \mathbb{P}_A(\Gamma;\Delta)$. So, there is a $\lambda\mu$-term $w \in \Lambda_A^\mu(\Gamma;\Delta)$ such that $t \equiv !k = !\overline{[\![w]\!]}$.

□

Lemma 3.17
$?\mathbb{P}_A{}^\perp \parr \mathbb{P}_B = \mathbb{P}_{A\to B}$

Proof. $(\subset)$: We take any Γ and Δ, and let $k \in (?\mathbb{P}_A^\perp \parr \mathbb{P}_B)(\Gamma;\Delta)$. Since $?\mathbb{P}_A^\perp \parr \mathbb{P}_B = \langle(?\mathbb{P}_A^\perp)^\perp, \mathbb{P}_B^\perp\rangle^\perp$ and $x \in (?\mathbb{P}_A^\perp)^\perp(\Gamma, x : A; \Delta, \alpha : B)$ and $\alpha \in \mathbb{P}_B^\perp(\Gamma, x : A; \Delta, \alpha : B)$, we have $(x \otimes \alpha) \bullet k \in \mathcal{B}^\circ(\Gamma, x : A; \Delta, \alpha : B)$. Therefore there exists a $\lambda\mu$ named-term σ such that $\sigma \in \Lambda^\mu(\Gamma, x : A; \Delta, \alpha : B)$ and $x \otimes \alpha \bullet k = [\![\sigma]\!]$. Then we obtain $k = (x,\alpha)(x \otimes \alpha \bullet k) = (x,\alpha)[\![\sigma]\!] \equiv (x,\alpha)[\![\mu\alpha.\sigma]\!]_\alpha = z.(z \bullet (x,\alpha)[\![\mu\alpha.\sigma]\!]_\alpha) \equiv z.[\![\lambda x\mu\alpha.\sigma]\!]_z \equiv \overline{[\![\lambda x\mu\alpha.\sigma]\!]} \in \mathbb{P}_{A\to B}(\Gamma;\Delta)$

Conversely, if we assume $k \in \mathbb{P}_{A\to B}(\Gamma;\Delta)$ for any Γ and Δ. Then there exists a $\lambda\mu$-term $v \in \Lambda^\mu_{A\to B}(\Gamma;\Delta)$ such that $k = \overline{[\![v]\!]}$. On the other hand, assume $\Gamma \subset \Gamma'$, $\Delta \subset \Delta'$, $t \in (?\mathbb{P}_A^\perp)^\perp(\Gamma';\Delta')$ and $u \in \mathbb{P}_B^\perp(\Gamma';\Delta')$, there is a $\lambda\mu$-term $w \in \Lambda_A^\mu(\Gamma';\Delta')$ such that $t = !\overline{[\![w]\!]}$ by the previous lemma. Then $(t \otimes u) \bullet k = (!\overline{[\![w]\!]} \otimes u) \bullet \overline{[\![v]\!]} = [\![v]\!]_{!\overline{[\![w]\!]} \otimes u} \equiv [\![vw]\!]_u = u \bullet \overline{[\![vw]\!]} \in \mathcal{B}^\circ(\Gamma';\Delta')$. Therefore we obtain $k \in \langle(?\mathbb{P}_A^\perp)^\perp, \mathbb{P}_B^\perp\rangle^\perp(\Gamma;\Delta) = (?\mathbb{P}_A^\perp \parr \mathbb{P}_B)(\Gamma;\Delta)$.

□

Proposition 3.18
There is a logical $\mathfrak{D}^\circ$-predicate $\{\widehat{\mathbb{P}}_\xi\}$ such that $\widehat{\mathbb{P}}_{A^\circ} = \mathbb{P}_A$ holds for any $\lambda\mu$-type A.

Proof. When we define $\widehat{\mathbb{P}}_{X^\perp} := \mathbb{P}_X$, $\widehat{\mathbb{P}}_X := \mathbb{P}_X{}^\perp$ and $\widehat{\mathbb{P}}_\perp := \mathcal{B}^\circ$, then the logical $\mathfrak{D}^\circ$-predicate $\{\widehat{\mathbb{P}}_\xi\}$ is defined recursively. Now, we check $\widehat{\mathbb{P}}_{A^\circ} = \mathbb{P}_A$ holds for any $\lambda\mu$-type A. This is shown by induction on A. When A is the basic type X, it is trivial. For the case of arrow type $A \to B$, we have

$$\widehat{\mathbb{P}}_{(A\to B)^\circ} = \widehat{\mathbb{P}}_{?(A^\circ)^\perp \parr B^\circ} = ?\widehat{\mathbb{P}}_{A^\circ} \parr \widehat{\mathbb{P}}_{B^\circ} \overset{I.H.}{=} ?\mathbb{P}_A \parr \mathbb{P}_B = \mathbb{P}_{A\to B}$$

□

Then we obtain the following theorem immediately by applying the Basic Lemma and Lemma 3.14 to $\{\widehat{\mathbb{P}}_\xi\}$.

Theorem 3.19 (fullness of the negative-translation)
Let Γ and Δ be contexts of the $\lambda\mu$-calculus, and suppose $\vdash\ ?(\Gamma^\circ)^\perp, \Delta^\circ\ ;\ k : A^\circ$. Then there exists $w \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $k = \overline{[\![w]\!]}$.

Proof. For all $x : A \in \Gamma$ and $\alpha : B \in \Delta$, $x \in ?(\mathbb{P}^\perp_A)^\perp(\Gamma;\Delta) = ?(\widehat{\mathbb{P}}^\perp_{A^\circ})^\perp(\Gamma;\Delta) = \widehat{\mathbb{P}}_{(?(A^\circ)^\perp)^\perp}(\Gamma;\Delta)$, and $\alpha \in \mathbb{P}^\perp_{B^\circ}(\Gamma;\Delta) = \widehat{\mathbb{P}}^\perp_{B^\circ}(\Gamma;\Delta) = \widehat{\mathbb{P}}_{(B^\circ)^\perp}(\Gamma;\Delta)$ hold. So, we now apply the Basic Lemma for the logical $\mathfrak{D}^\circ$-predicate $\{\widehat{\mathbb{P}}_\xi\}$, then we obtain $k \equiv k[\vec{x}/\vec{x}, \vec{\alpha}/\vec{\alpha}] \in \widehat{\mathbb{P}}_{A^\circ}(\Gamma;\Delta) = \mathbb{P}_A(\Gamma;\Delta)$. This means there is a $\lambda\mu$-term $w \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $k = \overline{[\![w]\!]}$.

□

3.7 Fullness of the positive-translation

In this section, we discuss only *the positive-translation*, therefore we consider $\overline{[\![w]\!]}$ and $[\![\tau]\!]$ as the images of w and τ by *the positive-translation*.

Definition 3.21
For a type A and contexts Γ, Δ of the $\lambda\mu$-calculus, we define
$\mathcal{B}^\bullet(\Gamma;\Delta) := \{\tau \in \mathfrak{D}^\bullet_\perp(\Gamma;\Delta) \mid \exists\sigma \in \Lambda^\mu(\Gamma;\Delta)\ (\tau = [\![\sigma]\!])\}$, and
$\mathbb{R}_A(\Gamma;\Delta) := \{\, t \in \mathfrak{D}^\bullet_{A^\bullet}(\Gamma;\Delta) \mid \exists V \in \Lambda^\mu_A(\Gamma;\Delta)\ (t = V^*)\,\}$.

Lemma 3.20
$\mathcal{B}^\bullet$ is a $\mathfrak{D}^\bullet$-predicate on $\perp$, and $\mathbb{R}_A$ is a $\mathfrak{D}^\bullet$-predicate on $A^\bullet$.

Proof. For each case, we will check the (**monotonicity**) and (**equality**) condition of $\mathfrak{D}^\bullet$-predicate.
(**monotonicity**): Assume $\Gamma \subset \Gamma'$ and $\Delta \subset \Delta'$. If $\tau \in \mathcal{B}^\bullet(\Gamma;\Delta)$, then there is a $\sigma \in \Lambda^\mu(\Gamma;\Delta) \subset \Lambda^\mu(\Gamma';\Delta')$ such that $\tau = [\![\sigma]\!]$. So, we have $\tau \in \mathcal{B}^\bullet(\Gamma';\Delta')$. Now, suppose we pick a $t \in \mathbb{R}_A(\Gamma;\Delta)$, then there is a value $V \in \Lambda^\mu_A(\Gamma;\Delta) \subset \Lambda^\mu_A(\Gamma';\Delta')$ such that $t = V^*$. Hence we have $t \in \mathbb{R}_A(\Gamma';\Delta')$.
(**equality**): Suppose $\tau \in \mathcal{B}^\bullet(\Gamma;\Delta)$ and $\tau = \tau'$, then there is a $\sigma \in \Lambda^\mu(\Gamma;\Delta)$ such that $\tau' = \tau = [\![\sigma]\!]$. Therefore, we have $\tau' \in \mathcal{B}^\bullet(\Gamma;\Delta)$. And suppose $t \in \mathbb{R}_A(\Gamma;\Delta)$ and $t = t'$. Then there is a value $V \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $t' = t = V^*$. Hence we have $t' \in \mathbb{R}_A(\Gamma;\Delta)$. □

Lemma 3.21
Let t be a positive term. Then $t = !(x,\alpha)(t \bullet ?(x \otimes \alpha))$ for any $x, \alpha \notin \mathrm{FV}(t)$.

Proof. This can be shown as follows:

$$\begin{aligned} !(x,\alpha)(t \bullet ?(x \otimes \alpha)) &= !(x,\alpha)(!z.(t \bullet ?z) \bullet ?(x \otimes \alpha)) \\ &\longrightarrow !(x,\alpha)((x \otimes \alpha) \bullet z.(t \bullet ?z)) \\ &\longrightarrow !z.(t \bullet ?z) \longrightarrow t \ . \end{aligned}$$

□

Lemma 3.22
$\mathbb{R}_A^{\perp\perp} = \mathbb{R}_A$.

Proof. Firstly, we consider the case when the type A is an atomic type X. Since $\mathfrak{D}_X^{\bullet}(\Gamma;\Delta) = \{x \mid x : X \in \Gamma\}$ (because variables are the only terms of the atomic type X), $\mathbb{R}_X(\Gamma;\Delta) \subset \mathbb{R}_X^{\perp\perp}(\Gamma;\Delta) \subset \{x \mid x : X \in \Gamma\}$. If $x : X$ is in Γ then $x \in \mathbb{R}_X(\Gamma;\Delta)$ since $x = x^*$ and $x \in \Lambda_X^{\mu}(\Gamma;\Delta)$. So, we obtain $\mathbb{R}_X(\Gamma;\Delta) = \mathbb{R}_X^{\perp\perp}(\Gamma;\Delta)$.

Secondly, we consider the case when the type A is an arrow type $A \to B$. We claim that $?(x \otimes \alpha) \in \mathbb{R}_{A\to B}^{\perp}(\Gamma, x : A; \Delta, \alpha : B)$ since for any $\Gamma' \supset (\Gamma, x : A)$, $\Delta' \supset (\Delta, \alpha : B)$ and $t \in \mathbb{R}_{A\to B}(\Gamma';\Delta')$, there exists a value V such that $V^* = t$ and we obtain

$$\begin{aligned} t \bullet ?(x \otimes \alpha) = V^* \bullet ?(x \otimes \alpha) = !z.(V^* \bullet ?(z \otimes \alpha)) \bullet ?x &\equiv [\![x]\!]_{!z.(V^* \bullet ?(z\otimes\alpha))} \\ &\equiv [\![Vx]\!]_{\alpha} \equiv [\![[\alpha]Vx]\!] \in \mathcal{B}^{\bullet}(\Gamma';\Delta'). \end{aligned}$$

Now, assume $u \in \mathbb{R}_{A\to B}^{\perp\perp}(\Gamma;\Delta)$. Since $u \bullet ?(x \otimes \alpha) \in \mathcal{B}^{\bullet}(\Gamma, x : A; \Delta, \alpha : B)$, then there exists a named-term τ in $\Lambda^{\mu}(\Gamma, x : A; \Delta, \alpha : B)$ such that $[\![\tau]\!] = u \bullet ?(x \otimes \alpha)$. By Lemma 3.21, we have $u = !(x,\alpha)(u \bullet ?(x \otimes \alpha)) = !(x,\alpha)[\![\tau]\!] \equiv !(x,\alpha)[\![\mu\alpha.\tau]\!]_{\alpha} \equiv (\lambda x \mu\alpha.\tau)^* \in \mathbb{R}_{A\to B}(\Gamma;\Delta)$. □

Lemma 3.23
(i) If $x : A \in \Gamma$, then $x \in \mathbb{R}_A(\Gamma;\Delta)$ for any Δ.
(ii) If $\alpha : A \in \Delta$, then $\alpha \in \mathbb{R}_A^{?\perp}(\Gamma;\Delta)$ for any Γ.

Proof. (i) Suppose $x\colon A \in \Gamma$, then $x \in \Lambda_A^{\mu}(\Gamma;\Delta)$ for any Δ. Hence we have $x \in \mathbb{R}_A(\Gamma;\Delta)$ since $x \equiv x^*$.
(ii) Suppose $\alpha\colon A \in \Delta$, $\Gamma \subset \Gamma'$, $\Delta \subset \Delta'$ and $k \in \mathbb{R}_A^{?}(\Gamma';\Delta')$. Then there is a $t \in \mathbb{R}_A(\Gamma';\Delta')$ such that $k \equiv ?t$, so there is a $V \in \Lambda_A^{\mu}(\Gamma';\Delta')$ such that $t = V^*$. Hence we have

$$\alpha \bullet k \equiv \alpha \bullet ?t = \alpha \bullet ?V^* \equiv [\![V]\!]_{\alpha} \equiv [\![[\alpha]V]\!] \in \mathcal{B}^{\bullet}(\Gamma';\Delta') \ .$$

Therefore we obtain $\alpha \in \mathbb{R}_A^{?\perp}(\Gamma;\Delta)$. □

Lemma 3.24
$?\mathbb{R}_A(\Gamma;\Delta) = \{\, k \in \mathfrak{D}^{\bullet}_{?A^{\bullet}}(\Gamma;\Delta) \mid \exists w \in \Lambda^{\mu}_A(\Gamma;\Delta)\,(\overline{[\![w]\!]} = k)\,\}$

Proof. ($\subset$): Assume $k \in ?\mathbb{R}_A(\Gamma;\Delta) = \mathbb{R}_A^{?\perp\perp}(\Gamma;\Delta)$, there exists a named-term $\tau \in \Lambda^{\mu}(\Gamma;\Delta,\alpha : A)$ such that $\alpha \bullet k = [\![\tau]\!]$ since $\alpha \in \mathbb{R}_A^{?\perp}(\Gamma;\Delta,\alpha : A)$ from the previous lemma. Hence, we have $k = \alpha.(\alpha \bullet k) = \alpha.[\![\tau]\!] \equiv \overline{[\![\mu\alpha.\tau]\!]} \in RHS$.
($\supset$): Assume that there exists $w \in \Lambda^{\mu}_A(\Gamma;\Delta)$ such that $k = \overline{[\![w]\!]}$, and suppose $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$ and $t \in \mathbb{R}^{?\perp}(\Gamma';\Delta')$. Since $x \in \mathbb{R}_A(\Gamma', x : A;\Delta')$ from the previous lemma, we have $?x \in \mathbb{R}_A^{?}(\Gamma', x : A;\Delta')$. Hence there exists a named-term $\tau \in \Lambda^{\mu}(\Gamma', x : A;\Delta')$ such that $t \bullet ?x = [\![\tau]\!] \equiv [\![\mu\alpha.\tau]\!]_{\alpha}$. So, we have $!(x,\alpha)(t \bullet ?x) = !(x,\alpha)[\![\mu\alpha.\tau]\!]_{\alpha} \equiv (\lambda x\mu\alpha.\tau)^{*}$. Therefore we obtain $t = !x.(t \bullet ?x) = !x.(!(x,\alpha)(t \bullet ?x) \bullet ?(x \otimes \alpha)) = !x.((\lambda x\mu\alpha.\tau)^{*} \bullet ?(x \otimes \alpha))$. Hence,

$$\begin{aligned} t \;\bullet\; k &= !x.((\lambda x\mu\alpha.\tau)^{*} \bullet ?(x \otimes \alpha)) \bullet \overline{[\![w]\!]} = [\![w]\!]_{!x.((\lambda x\mu\alpha.\tau)^{*} \bullet ?(x\otimes\alpha))} \\ &\equiv [\![(\lambda x\mu\alpha.\tau)w]\!]_{\alpha} \equiv [\![[\alpha](\lambda x\mu\alpha.\tau)w]\!] \in \mathcal{B}^{\bullet}(\Gamma';\Delta'). \end{aligned}$$

So, we obtain $k \in \mathbb{R}_A^{?\perp\perp}(\Gamma;\Delta) = ?\mathbb{R}_A(\Gamma;\Delta)$. □

Lemma 3.25
$\mathbb{R}_{A\to B} = !(\mathbb{R}_A^{\perp} \parr ?\mathbb{R}_B)$

Proof. ($\subset$): Given Γ and Δ, suppose $t \in \mathbb{R}_{A\to B}(\Gamma;\Delta)$. There exists a value $V \in \Lambda^{\mu}_{A\to B}(\Gamma;\Delta)$ such that $t = V^{*}$. By Lemma 3.21, we have $t = V^{*} = !(x,\alpha)(V^{*} \bullet ?(x \otimes \alpha))$ for fresh variables x and α. Let $\Gamma' \supset \Gamma$, $\Delta' \supset \Delta$, $u \in \mathbb{R}_A(\Gamma';\Delta')$ and $u' \in (?\mathbb{R}_B)^{\perp}(\Gamma';\Delta')$. Then there exists a value $W \in \Lambda^{\mu}_A(\Gamma';\Delta')$ such that $u = W^{*}$. So we have

$$\begin{aligned} (u \otimes u') \bullet (x,\alpha)(V^{*} \bullet ?(x \otimes \alpha)) &= (W^{*} \otimes u') \bullet (x,\alpha)(V^{*} \bullet ?(x \otimes \alpha)) \\ &= V^{*} \bullet ?(W^{*} \otimes u') = !z.(V^{*} \bullet ?(z \otimes u')) \bullet ?W^{*} \\ &\equiv [\![W]\!]_{!z.(V^{*} \bullet ?(z\otimes u'))} \equiv [\![VW]\!]_{u'} \\ &\equiv u' \bullet \overline{[\![VW]\!]} \in \mathcal{B}^{\bullet}(\Gamma';\Delta') \end{aligned}$$

since $\overline{[\![VW]\!]} \in ?\mathbb{R}_B(\Gamma';\Delta')$ from the previous lemma. Hence we obtain $(x,\alpha)(V^{*} \bullet ?(x\otimes\alpha)) \in \langle \mathbb{R}_A, (?\mathbb{R}_B)^{\perp}\rangle^{\perp}(\Gamma;\Delta)$, and then we have

$$t = !(x,\alpha)(V^{*} \bullet ?(x \otimes \alpha)) \in !\langle \mathbb{R}_A, (?\mathbb{R}_B)^{\perp}\rangle^{\perp}(\Gamma;\Delta) = !(\mathbb{R}_A^{\perp} \parr ?\mathbb{R}_B)(\Gamma;\Delta)\;.$$

($\supset$): Assume $t \in !(\mathbb{R}_A^{\perp} \parr ?\mathbb{R}_B)(\Gamma;\Delta) = \langle \mathbb{R}_A, (?\mathbb{R}_B)^{\perp}\rangle^{\perp\perp?\perp}(\Gamma;\Delta)$. Since $x \in \mathbb{R}_A(\Gamma, x : A;\Delta)$ and $\alpha \in (?\mathbb{R}_B)^{\perp}(\Gamma;\Delta,\alpha : B)$ from Lemma 3.23, there exists a named-term $\tau \in \Lambda^{\mu}(\Gamma, x : A;\Delta,\alpha :$

B) such that $t \bullet ?(x \otimes \alpha) = [\![\tau]\!]$. Therefore we obtain $t = !(x,\alpha)(t \bullet ?(x \otimes \alpha)) = !(x,\alpha)[\![\tau]\!] \equiv$ $!(x,\alpha)[\![\mu\alpha.\tau]\!]_\alpha \equiv (\lambda x\mu\alpha.\tau)^* \in \mathbb{R}_{A\to B}(\Gamma;\Delta)$. □

Now, we can prove the following proposition similar to as Proposition 3.18.

Proposition 3.26
Then there is a logical $\mathfrak{D}^\bullet$-predicate $\{\widehat{\mathbb{R}_\xi}\}$ such that $\widehat{\mathbb{R}_{A^\bullet}} = \mathbb{R}_A$ holds for any type A of the $\lambda\mu$-calculus.

Proof. When we define $\widehat{\mathbb{R}_X} := \mathbb{R}_X$, $\widehat{\mathbb{R}_{X^\perp}} := \mathbb{R}_X{}^\perp$ and $\widehat{\mathbb{R}_\perp} := \mathscr{B}^\bullet$, then the logical $\mathfrak{D}^\bullet$-predicate $\{\widehat{\mathbb{R}_\xi}\}$ is defined recursively. Now, we check $\widehat{\mathbb{R}_{A^\bullet}} = \mathbb{R}_A$ holds for any $\lambda\mu$-type A. This is shown by induction on A. When A is the basic type X, it is trivial. For the case of arrow type $A \to B$, we have $\widehat{\mathbb{R}_{(A\to B)^\bullet}} = \widehat{\mathbb{R}_{!((A^\bullet)^\perp ⅋ ?B^\bullet)}} = !(\widehat{\mathbb{R}}^\perp_{A^\bullet} ⅋ ?\widehat{\mathbb{R}_{B^\bullet}}) \overset{I.H.}{=} !(\mathbb{R}^\perp_A ⅋ ?\mathbb{R}_B) \overset{Lem\ 3.25}{=} \mathbb{R}_{A\to B}$ □

Therefore we have the following theorem by applying the Basic Lemma and Lemma 3.23 to $\{\widehat{\mathbb{R}_\xi}\}$.

Theorem 3.27 (fullness of the positive-translation)
Let Γ and Δ be contexts of the $\lambda\mu$-calculus, then the following hold.
(i) if $\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet\ ;\ t : A^\bullet$ then there exists $V \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $t = V^*$.
(ii) if $\vdash (\Gamma^\bullet)^\perp, ?\Delta^\bullet\ ;\ k : ?A^\bullet$ then there exists $w \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $k = \overline{[\![w]\!]}$.

Proof. By Lemma 3.23 and Proposition 3.26, we have

$$\begin{aligned} x &\in \mathbb{R}_A(\Gamma;\Delta) = \widehat{\mathbb{R}_{A^\bullet}}(\Gamma;\Delta) = \widehat{\mathbb{R}}^{\perp\perp}_{A^\bullet}(\Gamma;\Delta) = \widehat{\mathbb{R}}_{(A^\bullet)^{\perp\perp}}(\Gamma;\Delta), \mathit{and} \\ \alpha &\in \mathbb{R}^{?\perp}_B(\Gamma;\Delta) = \widehat{\mathbb{R}}^{?\perp}_{B^\bullet}(\Gamma;\Delta) = \widehat{\mathbb{R}}^{?\perp\perp\perp}_{B^\bullet}(\Gamma;\Delta) \\ &= (?\widehat{\mathbb{R}_{B^\bullet}})^\perp(\Gamma;\Delta) = \widehat{\mathbb{R}}_{(?B^\bullet)^\perp}(\Gamma;\Delta) \end{aligned}$$

for all $x : A \in \Gamma$ and $\alpha : B \in \Delta$. So, we now apply the Basic Lemma for the logical $\mathfrak{D}^\bullet$-predicate $\{\widehat{\mathbb{R}_\xi}\}$, then we obtain:
(i) $t \equiv t[\vec{x}/\vec{x}, \vec{\alpha}/\vec{\alpha}] \in \widehat{\mathbb{R}_{A^\bullet}}(\Gamma;\Delta) = \mathbb{R}_A(\Gamma;\Delta)$. This means there is a value $V \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $t = V^*$, and
(ii) $k \equiv k[\vec{x}/\vec{x}, \vec{\alpha}/\vec{\alpha}] \in \widehat{\mathbb{R}_{?A^\bullet}}(\Gamma;\Delta) = ?\mathbb{R}_A(\Gamma;\Delta)$. By Lemma 3.24, This means there is a $\lambda\mu$-term $w \in \Lambda^\mu_A(\Gamma;\Delta)$ such that $k = \overline{[\![w]\!]}$. □

Chapter 4

Conclusion and Future Work

Conclusion

The main aim of the thesis was to observe the relationship between the computational duality and the logical duality. The computational duality is the duality between the call-by-name and call-by-value strategies. The logical duality is the duality of classical logic so-called de Morgan's duality. This logical duality of classical logic appears as right and left symmetry of Gentzen's sequent calculus LK, and positive and negative duality of Laurent's polarized linear logic. Wadler's dual calculus was a suitable system for researching this logical duality.

Chapter 2 discussed the relationship between the computational duality of call-by-value / call-by-name and the logical duality of LK. Especially, to study the relationship between the computational procedure and the cut-elimination procedure of LK, we replaced the equalities in Wadler's paper with reductions. We first analized Wadler's results, and specified the problematic rules of the $\lambda\mu$-calculus that cannot be simulated by the reductions of the dual calculus. These problematic rules are not essential rules of the $\lambda\mu$-calculus because they are not the normalization procedures of proofs, and there is no influence even if we remove these rules from call-by-value system. We refined the call-by-value and the call-by-name systems of the $\lambda\mu$-calculus and the dual calculus by deleting these problematic rules. These systems are defined as reduction systems. Then we gave the call-by-name translations between the call-by-name $\lambda\mu$-calculus and the call-by-name dual calculus, and showed that these translations preserve call-by-name reductions and satisfy reloading property. We also gave the call-by-value translations between the call-by-value $\lambda\mu$-calculus and the call-by-value dual calculus, and showed that these translations satisfy the properties similar to the call-by-name translations. Then we introduced the translation from the call-by-name $\lambda\mu$-calculus into the call-by-value one and its inverse translation by composing the above translations via the dual translations on the dual calculus. Finally, we proved that these translations preserves reductions and reloading property from the above results. The results of this chapter showed duality between the call-by-name and call-by-value $\lambda\mu$-calculi as reduction systems. This means that we succeded to give the best possible answer to Wadler's open question.

Chapter 3 discussed the relationship between the computational duality of call-by-value / call-by-name and the logical duality of positive / negative. We introduced a term calculus for a sufficiently large fragment of Laurent's polarized linear logic, called polarized dual calculus DCP$^-$, which is based on the idea of the dual calculus. Then we defined two translations from the call-by-name / the call-by-value $\lambda\mu$-calculi into DCP$^-$, and showed their soundness of derivations and reductions. Finally, we proved the fullness of these translations in a way similar to the logical predicate method used by Hasegawa.

Future Work

In Chapter 2, we gave the best possible answer to Wadler's open question, but the $\lambda\mu$-calculi and the dual calculi that we had introduced did not enjoy storongly normalization. This fact does not necessarily mean these systems are meaningless. Actually, Tzevelekos [47] showed that the dual calculus satisfies strongly normalization and Church-Rosser property by assuming appropriate side-conditions. There is a possibility that our $\lambda\mu$-calculi can be refined to satisfy strongly normalizing and Church-Rosser property if we assume some side-conditions.

Another work in the future is to extend the results in this thesis. If we want to apply our results to more practical and powerful programming languages, we should discuss and extend our results about two important concepts: a fixed-point operator and data types. From this motivation, Kakutani [32] extended the $\lambda\mu$-calculi by adding a fixed-point operator and an iteration operator to the call-by-name system and the call-by-value one respectively. He followed Selinger's category-theoretic approach, and showed duality between call-by-name recursion and call-by-value iteration. Therefore we might be able to extend the results in this thesis via this line, and explain the duality between recursion and iteration by Wadler's syntactical approach.

Bibliography

[1] Y. Akama. On mints' reduction for ccc-calculus. In *TLCA '93: Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 1–12, London, UK, 1993. Springer-Verlag.

[2] R. M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*. Number 46 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1998.

[3] F. Barbanera and S. Berardi. A strong normalization result for classical logic. *Annals of Pure and Applied Logic*, 76(2):99–116, 1995.

[4] F. Barbanera and S. Berardi. A symmetric lambda calculus for classical program extraction. *Information and Computation*, 125(2):103–117, 1996.

[5] H. Barendregt. Lambda calculi with types. In *Handbook of Logic in Computer Science, Volumes 1 (Background: Mathematical Structures) and 2 (Background: Computational Structures), Abramsky & Gabbay & Maibaum (Eds.), Clarendon*, volume 2, pages 117–309. Oxford University Press, Inc., 1992.

[6] P. N. Benton, G. Bierman, and V. Paiva. Computational types from a logical perspective. *Journal of Functional Programming*, 8(2):177–193, 1998.

[7] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.

[8] T. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76(2-3):95–120, 1988.

[9] P.-L. Curien and H. Herbelin. The duality of computation. In *ICFP '00: Proceedings of the fifth ACM SIGPLAN international conference on Functional programming*, pages 233–243. ACM Press, 2000.

[10] H. B. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland Publishing Company, Amsterdam, 1958.

[11] V. Danos, J.-B. Joinet, and H. Schellinx. LKQ and LKT: Sequent calculi for second order logic based upon linear decomposition of classical implication. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 211–224. Cambridge University Press, 1995.

[12] V. Danos, J.-B. Joinet, and H. Schellinx. A new deconstructive logic: Linear logic. *Journal of Symbolic Logic*, 62(1):755–807, 1997.

[13] P. de Groote. A CPS-translation of the $\lambda\mu$-calculus. In *Proceedings 19th Intl. Coll. on Trees in Algebra and Programming, CAAP'94, Edinburgh, UK, 11–13 Apr 1994*, volume 787, pages 85–99, Berlin, 1994. Springer-Verlag.

[14] P. de Groote. Strong normalization of classical natural deduction with disjunction. In *Fifth International Conference on Typed Lambda Calculi and Applications, TLCA'01*, volume 2044 of *Lecture Notes in Computer Science*, pages 182–196. Springer-Verlag, 2001.

[15] Ken etsu Fujita. A sound and complete cps-translation for lambda-mu-calculus. In *Typed Lambda Calculi and Applications, 6th International Conference, Valencia, Spain, June 10-12, Proceedings*, pages 120–134, 2003.

[16] M. Felleisen, D. P. Friedman, E. Kohlbecker, and B. Duba. A syntactic theory of sequential control. *Theoretical Computer Science*, 52(3):205–237, 1987.

[17] A. Filinski. Declarative continuations and categorical duality. Master's thesis, Univ. of Copenhagen, 1989.

[18] H. Geuvers. *Logics and Type Systems*. PhD thesis, University of Nijmegen, 1993.

[19] N. Ghani. Beta-eta equality for coproducts. In *Proceedings of TLCA'95*, number 902 in Lecture Notes in Computer Science, pages 171–185. Springer-Verlag, 1995.

[20] J.-Y. Girard. *Interprétation fonctionelle et élimination des coupures dans l'arithmétique d'ordre supérieur*. PhD thesis, Univ. Paris VII, 1972.

[21] J.-Y. Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45(2):159–192, 1986.

[22] J.-Y. Girard. Linear logic. *Theoretical Compter Science*, 50:1–102, 1987.

[23] J.-Y. Girard. A new constructive logic: classical logic. *Mathematical Structures in Computer Science*, 1(3):225–296, 1991.

[24] J.-Y. Girard, P. Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, New York, NY, USA, 1989.

[25] T. G. Griffin. A formulae-as-type notion of control. In *POPL '90: Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 47–58, New York, NY, USA, 1990. ACM Press.

[26] M. Hasegawa. Logical Predicates for Intuitionistic Linear Type Theories. In *Typed Lambda Calculi and Applications (TLCA'99)*, volume 1581 of *Lecture Notes in Computer Science*, pages 198–212. Springer-Verlag, 1999.

[27] M. Hasegawa. Girard translation and logical predicates. *Journal of Funct. Prog.*, pages 77–89, 2000.

[28] H. Herbelin. A lambda-calculus structure isomorphic to gentzen-style sequent calculus structure. In *CSL '94: Selected Papers from the 8th International Workshop on Computer Science Logic*, pages 61–75. Springer-Verlag, 1994.

[29] J. Roger Hindley. *Basic simple type theory*. Cambridge University Press, New York, NY, USA, 1997.

[30] W. A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, Inc., New York, N.Y., 1980.

[31] C. B. Jay and N. Ghani. The virtues of eta-expansion. *Journal of Functional Programming*, 5(2):135–154, 1995.

[32] Y. Kakutani. Duality between call-by-name recursion and call-by-value iteration. In *CSL '02: Proceedings of the 16th International Workshop and 11th Annual Conference of the EACSL on Computer Science Logic*, pages 506–521, London, UK, 2002. Springer-Verlag.

[33] O. Laurent. *Étude de la polarisation en logique.* PhD thesis, Univ. Aix-Marseille 2, 2002.

[34] O. Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1):161–188, 2003.

[35] O. Laurent, M. Quatrini, and Lorenzo T. de Falco. Polarized and focalized linear and classical proofs. *Annals of Pure and Applied Logic*, 134(2–3):217–264, 2005.

[36] O. Laurent and L. Regnier. About translations of classical logic into polarized linear logic. In *Proceedings of the eighteenth annual IEEE symposium on Logic In Computer Science*, pages 11–20. IEEE Computer Society Press, 2003.

[37] E. Moggi. Computational lambda-calculus and monads. In *Proceedings 4th Annual IEEE Symp. on Logic in Computer Science, LICS'89, Pacific Grove, CA, USA, 5–8 June 1989*, pages 14–23. IEEE Computer Society Press, Washington, DC, 1989.

[38] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.

[39] C-H. L. Ong and C. A. Stewart. A curry-howard foundation for functional computation with control. In *Proc. of the Symposium on Principles of Programming Languages*, pages 215–227, 1997.

[40] M. Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proc. of International Conference on Logic Programming and Automated Deduction*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer-Verlag, 1992.

[41] Michel Parigot. Free deduction: An analysis of "computations" in classical logic. In *RCLP*, pages 361–380, 1991.

[42] G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

[43] J. C. Reynolds. Towards a theory of type structure. In *Programming Symposium, Proceedings Colloque sur la Programmation*, pages 408–423, London, UK, 1974. Springer-Verlag.

[44] A. Sabry and M. Felleisen. Reasoning about programs in continuation-passing style. *In Lisp and Symbolic Computation*, 6(3/4):289–360, 1993.

[45] P. Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, pages 207–260, 2001.

[46] A. S. Troelstra and H. Schwichtenberg. *Basic proof theory*. Cambridge University Press, New York, NY, USA, 1996.

[47] N. Tzevelekos. Investigations on the dual calculus. *Theoretical Computer Science*, 360(1):289–326, 2006.

[48] P. Wadler. Call-by-Value is Dual to Call-by-Name. In *International Conference on Functional Programming, Uppsala, Sweden*, pages 25–29, 2003.

[49] P. Wadler. Call-by-Value is Dual to Call-by-Name – Reloaded. In *Rewriting Techniques and Applications, Nara, Japan*, pages 185–203. Springer, 2005.

[50] Y. Yamagata. Strong normalization of the second order symmetric lambda-mu calculus. *Information and Computation*, 193(1):1–20, 2004.