

# Strong normalization of classical natural deduction with disjunctions

Koji Nakazawa<sup>a,\*</sup>

<sup>a</sup>*Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan.*

Makoto Tatsuta<sup>b</sup>

<sup>b</sup>*National Institute of Informatics, 2-1-2 Hitotsubashi, Tokyo, 101-8430, Japan.*

---

## Abstract

This paper proves strong normalization of classical natural deduction with disjunction and permutative conversions, by using CPS-translation and augmentations. By them, this paper also proves strong normalization of classical natural deduction with general elimination rules for implication and conjunction, and their permutative conversions. This paper also proves natural deduction can be embedded into natural deduction with general elimination rules, strictly preserving proof normalization.

*Key words:* Classical natural deduction, Strong normalization, CPS-translation, Permutative conversion, General elimination

---

## 1 Introduction

There have been many studies on the computational aspect of classical natural deduction proofs, since Griffin [8] pointed out relationship between classical logic and type systems for control operators and Parigot [17] proposed the  $\lambda\mu$ -calculus, which corresponds to classical natural deduction. Normalization of classical proofs is an important subject in the research area, and there are many papers on the issue [4,18,7,5,23,1,3,11,12,15].

---

\* Corresponding author.

*Email addresses:* `knak@kuis.kyoto-u.ac.jp` (Koji Nakazawa),  
`tatsuta@nii.ac.jp` (Makoto Tatsuta).

The subformula property is important in natural deduction. In the intuitionistic natural deduction, normal proofs can be obtained by the proof normalization, which corresponds to the  $\beta$ -reduction in the simply typed  $\lambda$ -calculus. The subformula property states that any formula in a normal proof is some subformula in its end sequent. It holds for the following reason. In any normal proof, every major premise of any application of elimination rules is either an assumption or a consequence of another elimination rule, where a premise of an elimination rule is called major if it contains the logical connective which is eliminated by the rule. By this property, any formula in a normal proof is a subformula of its end sequent.

Permutative conversions are necessary for a natural deduction system with disjunction or existential quantification in order to have the subformula property. If we consider disjunction as a logical connective,  $\beta$ -normal proofs may fail to enjoy the subformula property for the following reason. The form of the disjunction elimination rule is

$$\frac{\Gamma \vdash \alpha \vee \beta \quad \Gamma, \alpha \vdash \gamma \quad \Gamma, \beta \vdash \gamma}{\Gamma \vdash \gamma} (\vee E),$$

so the conclusion  $\gamma$  may not be a subformula of the major premise  $\alpha \vee \beta$ . When the premise  $\gamma$  is produced by an introduction rule, and the conclusion  $\gamma$  is consumed by an elimination rule, the formula  $\gamma$  is not a subterm of its end sequent. In order to have the subformula property for normal proofs with disjunctions, we need another structural proof normalization, that is, permutative conversions. *Permutative conversions* are reduction rules which permute an application of the disjunction elimination and its successive elimination rule. For example, the proof

$$\frac{\frac{\frac{\vdots M}{\Gamma \vdash \alpha \vee \beta} \quad \frac{\frac{\vdots P_1}{\Gamma, \alpha \vdash \gamma_1 \rightarrow \gamma_2} \quad \frac{\vdots P_2}{\Gamma, \beta \vdash \gamma_1 \rightarrow \gamma_2}}{\Gamma \vdash \gamma_1 \rightarrow \gamma_2} (\vee E) \quad \frac{\vdots N}{\Gamma \vdash \gamma_1} (\rightarrow E)}{\Gamma \vdash \gamma_2} (\rightarrow E)$$

reduces to the proof

$$\frac{\frac{\vdots M}{\Gamma \vdash \alpha \vee \beta} \quad \frac{\frac{\vdots P_1}{\Gamma, \alpha \vdash \gamma_1 \rightarrow \gamma_2} \quad \frac{\vdots N}{\Gamma \vdash \gamma_1} (\rightarrow E)}{\Gamma, \alpha \vdash \gamma_2} (\rightarrow E) \quad \frac{\frac{\vdots P_2}{\Gamma, \beta \vdash \gamma_1 \rightarrow \gamma_2} \quad \frac{\vdots N}{\Gamma \vdash \gamma_1} (\rightarrow E)}{\Gamma, \beta \vdash \gamma_2} (\rightarrow E)}{\Gamma \vdash \gamma_2} (\vee E).$$

By permutative conversions, every major premise of any application of an elimination rule in a normal proof is either an assumption or the consequence of an elimination rule other than the disjunction elimination. Therefore, the subformula property holds in normal proofs with respect to the  $\beta$ -reduction and the permutative conversions. This discussion can be extended to the classical natural deduction with the  $\mu$ -rule, as de Groote showed in [5].

In [5], de Groote introduced the calculus  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ , which is an extension of Parigot's  $\lambda\mu$ -calculus and corresponds to the classical natural deduction with implication, conjunction and disjunction. He also gave a proof of the strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ . The proof, however, contains an error as Matthes pointed out in [11].

In this paper, we prove the strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  by means of a CPS-translation and augmentations. The notion of augmentations for  $\lambda\mu$ -calculus was introduced in [13] to correct the error of the strong normalization proof by a CPS-translation in [18]. The error in the proof of [5] is due to the same sort of problem, which is called *erasing-continuations* in [13]. We will show the augmentations can work also for conjunctions, disjunctions, and their permutative conversions. This paper corrects the error of the proof in [5] by applying the strong normalization proof by CPS-translation and augmentations to  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

We will explain erasing-continuation. In CPS-translating  $(\mu a.M)N$ ,  $N$  is passed to  $M$  by passing a continuation containing  $N$  to  $M$ . However, this continuation is erased and not actually passed when the variable  $a$  does not occur in  $M$ . We call this phenomenon *erasing-continuation*. Erasing-continuation disturbs strict preservation of reduction in the translation. A CPS-translation maps both a term  $(\mu a.M)N$  and its reduct  $\mu a.M'$  by  $\mu$ -reduction to the same term  $P$ . If  $a$  does not occur in  $M$ , then  $N$  does not occur in  $M'$  nor  $P$ . Therefore, even if  $N$  has some  $\beta$ -redex and  $N'$  is its  $\beta$ -reduct, a CPS-translation maps  $(\mu a.M)N$  and  $(\mu a.M)N'$  to the same term. Hence  $\beta$ -reduction is not strictly preserved by this CPS-translation.

De Groote tried to solve this problem by defining his CPS-translation of  $(\mu a.M)N$  by cases according to whether  $\mu a.M$  is vacuous or not, where  $\mu a.M$  is called *vacuous* when the variable  $a$  does not occur in  $M$ . However, this definition did not work in fact since it failed to preserve  $\beta$ -reduction.

In order to recover his proof, we use augmentations. For each term  $M$ , we define a set  $\text{Aug}(M)$  of its augmentations where the augmentation is a  $\beta$ -expansion of  $M$  and simulates reduction of  $M$ , and every subterm of the augmentation is not vacuous. We will show that if  $M$  reduces to  $N$  and  $M^+$  is in  $\text{Aug}(M)$ , then there exists some  $N^+$  in  $\text{Aug}(N)$  such that  $M^+$  reduces to  $N^+$ . Since an augmentation does not have vacuous subterms, the strong normalization of typed augmentations is proved by a CPS-translation. Combining augmentations and a CPS-translation, consequently we can prove the strong normalization of  $\lambda^{\rightarrow\wedge\vee\perp}$ .

General elimination rules have been studied [9,21,10,14]. For conjunction and implication, we can consider the elimination rules similar to the disjunction elimination. Von Plato [21] called such rules *general elimination rules* and gave a proof system for an intuitionistic natural deduction with the general elimi-

nation rules. General elimination rules have permutative conversions. He [21] proved that there is a one-to-one correspondence between normal proofs with respect to the  $\beta$ -reduction and the permutative conversions in his system and cut-free proofs in the intuitionistic sequent calculus. Permutative conversions also play an important role in relationship between natural deduction and sequent calculus. Furthermore, Nakazawa [14] showed an isomorphism between the proof normalization in the natural deduction with the general elimination and a cut-elimination procedure in the sequent calculus.

This paper defines the classical natural deduction  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  with general elimination rules, and proves the strong normalization of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  by means of a CPS-translation and the idea of augmentations. We show the augmentations can work also for general elimination rules of implication, conjunction, and their permutative conversions. We also show the strong normalization of untyped  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  with respect to  $\mu$ -reduction and permutative conversions by extending the norm defined in [5]. The strong normalization of the natural deduction with general elimination rules is proved by Joachimski and Matthes [10] for the implicational fragment, and by Nakazawa [14]. These proofs are only for the intuitionistic case and it has not been proved for the classical case yet.

This paper also gives an embedding from  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ , which preserves the typability and strictly preserves steps of proof normalization. This embedding and the strong normalization of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  give another proof of the strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

Section 2 defines the classical natural deduction system  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ . We discuss erasing-continuation in Section 3. The strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  is proved in Section 4. Section 5 defines the system  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  with general elimination rules and proved its strong normalization. The embedding from  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  is given in Section 6.

## 2 Definition of the system $\lambda\mu^{\rightarrow\wedge\vee\perp}$

In this section, we give the definition of the classical natural deduction system  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  with disjunction and permutative conversions [5].  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  is an extension of Parigot's  $\lambda\mu$ -calculus [18] with conjunction, disjunction and permutative conversions.

**Definition 1** ( $\lambda\mu^{\rightarrow\wedge\vee\perp}$ ) *We define  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .*

(1) Types, which are also called formulas, are defined as follows:

$$\alpha, \beta ::= \perp \mid \alpha \rightarrow \beta \mid \alpha \wedge \beta \mid \alpha \vee \beta.$$

(2) We have two kinds of variables, one is the  $\lambda$ -variables, denoted by  $x, y, \dots$ , and the other is the  $\mu$ -variables, denoted by  $a, b, \dots$ . Terms are defined by:

$$\begin{aligned} M, N ::= & x \mid \lambda x.M \mid \langle M, N \rangle \mid \iota_j M \mid MN \mid M\pi_j \mid M[x_1.N_1, x_2.N_2] \\ & \mid aM \mid \mu a.M, \end{aligned}$$

where  $j$  is either 1 or 2. We will use the meta-variable  $j$  to denote the index 1 or 2.  $x$  in  $M$  is bound in  $\lambda x.M$ ,  $x_j$  in  $N_j$  is bound in  $M[x_1.N_1, x_2.N_2]$ , and  $a$  in  $M$  is bound in  $\mu a.M$ . Renaming bound variables is admitted as usual. Eliminators are defined by

$$\epsilon ::= M \mid \pi_j \mid [x.N, y.L].$$

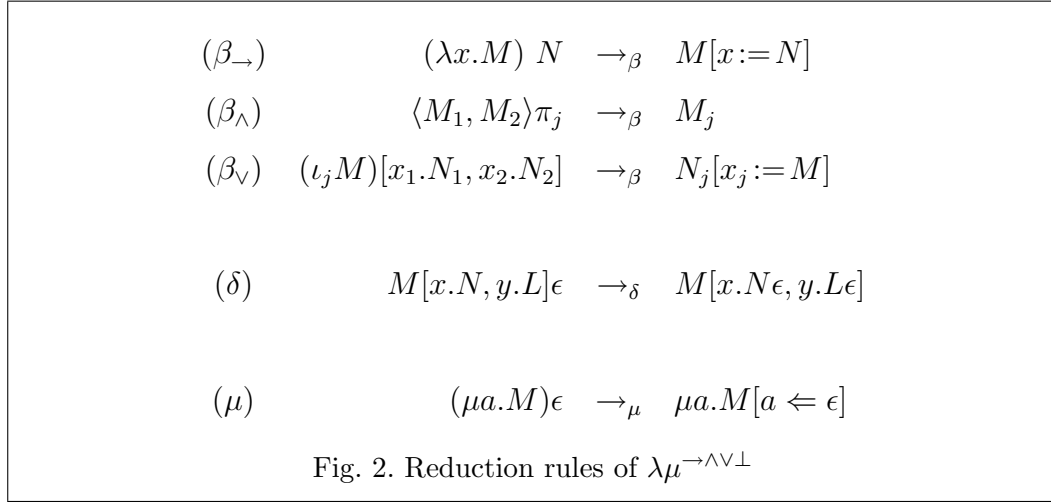
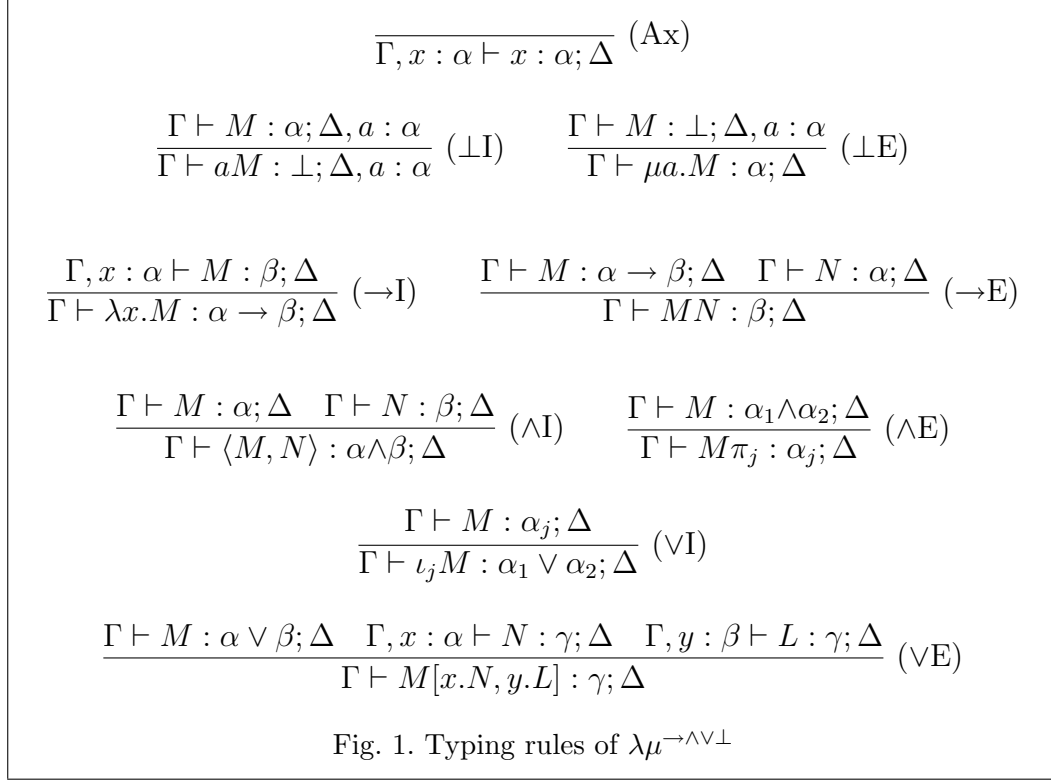
By these, we can treat instances of elimination rules for three logical connectives in a uniform manner such as  $M\epsilon$ . We write  $\bar{\epsilon}$  for a finite sequence of eliminators, and we use  $M\bar{\epsilon}$  to denote  $(\dots(M\epsilon_1)\epsilon_2\dots)\epsilon_n$  for  $\bar{\epsilon} \equiv \epsilon_1, \dots, \epsilon_n$ . If  $\bar{\epsilon}$  is empty,  $M\bar{\epsilon}$  denotes  $M$ .

(3) The form of typing judgments is  $\Gamma \vdash M : \alpha; \Delta$ , where  $\Gamma$  is a set of type assignments for  $\lambda$ -variables in the form of  $x : \alpha$  and  $\Delta$  is a set of type assignments for  $\mu$ -variables in the form of  $a : \alpha$ . We write  $\Gamma, x : \alpha$  for the set  $\Gamma \cup \{x : \alpha\}$ . Typing rules are in Figure 1.

(4) Substitution  $M[x := N]$  is defined as usual. For any eliminator, the structural substitution  $M[a \Leftarrow \epsilon]$  is defined as the term obtained from  $M$  by replacing each subterm  $aN$  by  $a(N[a \Leftarrow \epsilon]\epsilon)$ .  $\epsilon[x := N]$  and  $\epsilon[a \Leftarrow \epsilon']$  are similarly defined. Reduction rules are in Figure 2, where  $\epsilon$  is an arbitrary eliminator. One-step reduction  $\rightarrow$  is defined as their congruence closure. Strict reduction relation  $\rightarrow^+$  is the transitive closure of the relation  $\rightarrow$ , and reduction relation  $\rightarrow^*$  is the reflexive transitive closure of the relation  $\rightarrow$ . We will use the symbol  $\rightarrow_\beta$  to denote the one-step reduction relation defined by  $\beta$ -rules only. Relations  $\rightarrow_\mu$  and  $\rightarrow_\delta$  are similarly defined. A term  $M$  is said to be normal if there is no term  $N$  such that  $M \rightarrow N$ . We call the reduction defined by  $\delta$ -rule permutative conversion.

**Theorem 2 (Subject reduction property [5])** *If  $\Gamma \vdash M : \alpha; \Delta$  is derivable in  $\lambda\mu^{\rightarrow \wedge \vee \perp}$  and  $M \rightarrow N$  holds,  $\Gamma \vdash N : \alpha; \Delta$  is derivable in  $\lambda\mu^{\rightarrow \wedge \vee \perp}$ .*

The subformula property states that any formula occurring in a normal proof is a subformula of the end sequent in the proof. In order to have the property,



we have to introduce additional rules such as permutative conversions, as discussed in [5].  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  enjoys the subformula property due to the  $\mu$ -reduction and the permutative conversions. Permutative conversions are indispensable for the following reason. In the disjunction elimination rule, the conclusion  $\gamma$  may not be a subformula of the major premise  $\alpha \vee \beta$ . When the premise  $\gamma$  is produced by some introduction rule of some connective, and the conclusion  $\gamma$  is consumed by some elimination rule of the same connective, the formula  $\gamma$  may not be a subformula of the end sequent in the proof. If we considered only  $\beta$ -rules, the subformula property would not hold.

**Theorem 3 (Subformula property [5])** *Any formula  $\beta$  occurring in a normal proof of  $\Gamma \vdash M : \alpha; \Delta$  in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  is some subformula of some formula in  $\Gamma \vdash M : \alpha; \Delta$ .*

### 3 Erasing-Continuation and Augmentations

We discuss a strong normalization proof by a CPS-translation, erasing-continuation phenomena, and augmentations.

There are many attempts to prove strong normalization of calculi with control operators by reducing them to that of the simply typed  $\lambda$ -calculus by CPS-translations, such as [4,18,7,5]. We say that a translation *preserves reduction* when the translation of  $M$  reduces to the translation of  $N$  with some steps if  $M$  reduces to  $N$  with one step. We say that a translation *strictly preserves reduction* when the translation of  $M$  reduces to the translation of  $N$  with more than zero steps if  $M$  reduces to  $N$  with one step. In order to prove the strong normalization of some system, we can use a CPS-translation from that system into the simply typed  $\lambda$ -calculus such that the CPS-translation preserves reduction and strictly preserves  $\beta$ -reduction steps. Then we can reduce the strong normalization of the system to that of the simply typed  $\lambda$ -calculus. In the  $\lambda\mu$ -calculus, the CPS-translation defined by  $\overline{M} \equiv \lambda k.(M : k)$  preserves the reduction, where  $M : K$  is defined by:

$$\begin{aligned} x : K &\equiv xK, \\ \lambda x.M : K &\equiv K(\lambda x.\overline{M}), \\ MN : K &\equiv M : \lambda m.m\overline{N}K, \\ aM : K &\equiv M : k_a \quad (k_a \text{ is a } \lambda\text{-variable}), \\ \mu a.M : K &\equiv (M : \lambda x.x)[k_a := K]. \end{aligned}$$

*Erasing-continuation* is a phenomenon that some continuation passed to a term is erased in a translation because the continuation is passed by substituting it for a continuation variable in the term, and the term does not contain that variable. Erasing-continuation may occur in the above CPS-translation as follows. In the CPS-translation  $\mu a.M : K$  of the  $\mu$ -abstraction, a continuation is passed by substituting a continuation  $K$  for the continuation variable  $k_a$ . When  $M$  has no free  $a$ ,  $M : \lambda x.x$  does not contain  $k_a$  and the continuation  $K$  is erased in the CPS-translation  $\mu a.M : K$ . Then any redex in  $K$  is erased. For example,  $\overline{(\mu a.x)N}$  is identical with  $\lambda k.x(\lambda x.x)$  for any  $N$ , and any redex in  $N$  does not remain in the CPS-translation of  $(\mu a.x)N$ .

Erasing-continuation disturbs strong normalization proof by a translation. To

prove the strong normalization of one system, we can define a translation from that system to another system for which the strong normalization is already proved. If the translation strictly preserves reduction, we can prove the strong normalization of the first system by translating a reduction sequence to another reduction sequence in the second system. We expect that the CPS-translation strictly preserves reduction steps. However, it is not the case if erasing-continuation occurs, because some redex is erased in the translation, so the translation of  $M$  syntactically equals the translation of  $N$  even if  $M$  reduces to  $N$  with one step.

Several strong normalization proofs by CPS-translations contained errors due to the erasing-continuation problem [13]. For example, in [18], Parigot claimed the following three lemmas: (1) if  $M \rightarrow_\beta N$  holds in  $\lambda\mu$ , then we have  $\overline{\overline{M}} \rightarrow^+ \overline{\overline{N}}$ , (2) if  $M \rightarrow_\mu N$  holds, then we have  $\overline{\overline{M}} \equiv \overline{\overline{N}}$ , and (3) the  $\mu$ -reduction is strongly normalizing. The strong normalization of  $\lambda\mu$ -calculus would follow from these lemmas and the strong normalization of the simply typed  $\lambda$ -calculus if the claim (1) held. However the claim (1) does not hold indeed due to the erasing-continuation.

In [5], de Groote tried to prove the strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ . Matthes [11], however, pointed out that the proof contained an error. [5] defined the CPS-translation of  $\mu$ -abstraction  $\mu a.M$  by  $\mu a.M : K \equiv (\lambda k_a.(M : \lambda x.x))K$  when  $M$  has no free  $a$ . This translation, however, did not preserve reduction, as Matthes gave a counterexample in [11]. If we suppose  $M \equiv \mu a.(\lambda z.x)(ay)$  and  $N \equiv \mu a.x$ , then  $M \rightarrow_\beta N$  holds, and on the other hand, for their CPS-translations  $\overline{\overline{M}} \equiv \lambda k.(\lambda m.m(\lambda k_0.yk)(\lambda x.x))(\lambda zk_1.xk_1)$  and  $\overline{\overline{N}} \equiv \lambda k.(\lambda k_a.x(\lambda x.x))k$ , the former cannot reduce to the latter. Therefore, Lemma 10 of [5] does not hold, and strong normalization proof of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  of [5] has not been finished.

In order to use such a CPS-translation with erasing-continuation for proving strong normalization, augmentations were developed [13]. An augmentation of a term  $M$  is obtained from  $M$  by recursively replacing  $\mu a.N$  by  $\mu a.(\lambda z.N)(aP)$  with a fresh variable  $z$  and an appropriate term  $P$ . Since  $a$  always appears in  $L$  for every subterm  $\mu a.L$  in an augmentation, erasing-continuation does not occur when CPS-translating an augmentation. Hence a CPS-translation can be used to prove strong normalization of augmentations. Moreover augmentations can simulate reduction of the original terms. Hence we can prove strong normalization of the original terms by combining the CPS-translation and augmentations. Augmentations will be defined and more explained in Section 4.2.



## 4 Strong normalization of $\lambda\mu^{\rightarrow\wedge\vee\perp}$

In this section, we correct the proof in [5], by applying a CPS-translation and augmentations to  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

As we discussed in the previous section, the erasing-continuation problem happens for  $\mu$ -abstractions  $\mu a.M$  where  $M$  does not contain any free variable  $a$ . We call such a  $\mu$ -abstraction *vacuous*. The key idea of our proof is translating any reduction sequence of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to another sequence of augmentations, which are  $\beta$ -expansions of the original terms, have the same type, and contain no vacuous  $\mu$ -abstraction. Then we can apply a CPS-translation to the sequence of augmentations to prove its termination, since they are not vacuous. So we can reduce the strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to that of the simply typed  $\lambda$ -calculus.

First, we will define a CPS-translation and prove its preservation of typability and reduction. Secondly, we will define augmentations and explain them. Finally, we will prove strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

### 4.1 CPS-translation

The CPS-translation defined here is standard. It is essentially the same as that of [5], except we do not need the case analysis of the  $\mu$ -abstraction. It will work in our proof since we combine it with augmentations to avoid the erasing-continuation problem.

**Definition 4 (CPS-translation)** Suppose that we have some fixed  $\lambda$ -variable  $k_a$  for each  $\mu$ -variable  $a$ . CPS-translation of a  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term  $M$  is defined as  $\overline{\overline{M}} \equiv \lambda k.(M : k)$ , where  $M : K$  is defined for a  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term  $M$  and a  $\lambda$ -term  $K$  as follows:

$$\begin{aligned} x : K &\equiv xK, \\ \lambda x.M : K &\equiv K(\lambda x.\overline{\overline{M}}), \\ \langle M, N \rangle : K &\equiv K(\lambda x.x\overline{\overline{M}}\overline{\overline{N}}), \\ \iota_j M : K &\equiv K(\lambda y_1 y_2.y_j\overline{\overline{M}}), \\ M\epsilon : K &\equiv M : (\epsilon @ K), \\ aM : K &\equiv M : k_a, \\ \mu a.M : K &\equiv (M : \lambda x.x)[k_a := K], \end{aligned}$$

where the operation  $\epsilon @ K$  is defined as

$$\begin{aligned} N @ K &\equiv \lambda m.m \overline{\overline{N}} K, \\ \pi_j @ K &\equiv \lambda x.x(\lambda y_1 y_2.y_j K), \\ [x_1.P_1, x_2.P_2] @ K &\equiv \lambda m.m(\lambda x_1.(P_1 : K))(\lambda x_2.(P_2 : K)). \end{aligned}$$

**Definition 5 (Double negation translation)** *Double negation translation  $\overline{\overline{\alpha}}$  of a formula  $\alpha$  is defined as  $\neg\neg\alpha^*$ , where  $\neg\alpha$  is an abbreviation of  $\alpha \rightarrow \perp$ , and  $\alpha^*$  is defined by:*

$$\begin{aligned} \perp^* &\equiv \perp, \\ (\alpha \rightarrow \beta)^* &\equiv \overline{\overline{\alpha}} \rightarrow \overline{\overline{\beta}}, \\ (\alpha \wedge \beta)^* &\equiv \neg(\overline{\overline{\alpha}} \rightarrow \neg\overline{\overline{\beta}}), \\ (\alpha \vee \beta)^* &\equiv \neg\overline{\overline{\alpha}} \rightarrow \neg\neg\overline{\overline{\beta}}. \end{aligned}$$

For a set  $\Gamma$  of type assignments for  $\lambda$ -variables,  $\overline{\overline{\Gamma}}$  denotes the set  $\{x : \overline{\overline{\alpha}} \mid x : \alpha \in \Gamma\}$ . For a set  $\Delta$  of type assignments for  $\mu$ -variables,  $\neg\Delta^*$  denotes the set  $\{k_a : \neg\alpha^* \mid a : \alpha \in \Delta\}$ .

**Proposition 6** *If  $M$  is typable in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ ,  $\overline{\overline{M}}$  is typable in the simply typed  $\lambda$ -calculus. More precisely, if  $\Gamma \vdash M : \alpha; \Delta$  is derivable in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ , then  $\overline{\overline{\Gamma}}, \neg\Delta^* \vdash \overline{\overline{M}} : \overline{\overline{\alpha}}$  is derivable in the simply typed  $\lambda$ -calculus.*

*Proof.* By induction on the derivation of  $\Gamma \vdash M : \alpha; \Delta$ .  $\square$

**Definition 7** *A  $\mu$ -abstraction  $\mu a.M$  is said to be vacuous if any free  $\mu$ -variable  $a$  does not occur in  $M$ . A  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term  $M$  is said to be vacuous if  $M$  contains some vacuous  $\mu$ -abstraction term as its subterm.*

**Lemma 8** *For any terms  $P, K, K', M$  and any eliminator  $\epsilon$ , we have*

- (1)  $(P : K)[k := K'] \equiv P : K[k := K'] \quad (k \notin FV(P)),$
- (2)  $(P : K)[x := \overline{\overline{M}}] \rightarrow^* P[x := M] : K[x := \overline{\overline{M}}],$
- (3)  $(P : K)[k_a := \epsilon @ k_a] \equiv P[a \leftarrow \epsilon] : K[k_a := \epsilon @ k_a].$

*Proof.* By induction on  $P$ .  $\square$

**Lemma 9** *Suppose that a  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term  $M$  is typable and not vacuous, and  $K$  is a  $\lambda$ -term. Then we have the following.*

- (1) *If  $M$  contains a free  $\mu$ -variable  $a$  and the type of  $a$  is not  $\perp$ , then the free  $\lambda$ -variable  $k_a$  appears in  $M : K$ .*
- (2) *If the type of  $M$  is not  $\perp$ , then all the free variables of  $K$  appear in  $M : K$ .*

*Proof.* They are proved simultaneously by induction on the derivation of  $\Gamma \vdash M : \alpha; \Delta$ . We will use  $FV(M)$  to denote the set of free  $\lambda$ - and  $\mu$ -variables.

(1) The case where  $M \equiv M_1 M_2$  and  $a \in FV(M_2)$ . By the induction hypothesis for (1),  $\overline{\overline{M_2}} \equiv \lambda k. M_2 : k$  contains a free occurrence of  $k_a$ . Since the type of  $M_1$  is  $\beta \rightarrow \alpha \not\equiv \perp$ , by the induction hypothesis for (2), we have  $FV(M_1 : \lambda m. m \overline{\overline{M_2}} K) \supseteq FV(\lambda m. m \overline{\overline{M_2}} K) \supseteq FV(\overline{\overline{M_2}})$ , where the left hand side is equal to  $FV(M_1 M_2 : K)$ . Therefore,  $M_1 M_2 : K$  contains a free occurrence of  $k_a$ .

The case where  $M \equiv a M_1$ . Since the type of  $a$  is not  $\perp$ , the type of  $M_1$  is not  $\perp$  either. By the induction hypothesis for (2), we have  $k_a \in FV(M_1 : k_a) = FV(a M_1 : K)$ . Other cases are similarly proved.

(2) The case where  $M \equiv N[x_1. P_1, x_2. P_2]$ . Since the type of  $M$  is not  $\perp$ , that of  $P_j$  is not  $\perp$  either. By the induction hypothesis for (2), we have  $FV(K) \subseteq FV(P_j : K)$ . Furthermore, since the type of  $N$  is of the form  $\alpha_1 \vee \alpha_2$ , by the induction hypothesis for (2), we have  $FV(K) \subseteq FV(P_1 : K) \subseteq FV(N : \lambda m. m (\lambda x_1. (P_1 : K)) (\lambda x_2. (P_2 : K))) = FV(M : K)$ .

The case where  $M \equiv \mu a. N$ . Since the type of  $M$  is not  $\perp$ , the type of  $a$  is not  $\perp$  either. Since  $M$  is not vacuous, we have  $a \in FV(N)$ . By the induction hypothesis for (1), we have  $k_a \in FV(N : \lambda x. x)$ . Therefore, we have  $FV(K) \subseteq FV((N : \lambda x. x)[k_a := K]) = FV(M : K)$ . Other cases are similarly proved.  $\square$

**Proposition 10** *For any  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -terms  $M$  and  $N$ , if  $M \rightarrow N$  holds, we have  $\overline{\overline{M}} \rightarrow^* \overline{\overline{N}}$ . In particular, if  $M$  is typable and not vacuous, then we have the following.*

(1)  $M \rightarrow_\beta N$  implies  $\overline{\overline{M}} \rightarrow^+ \overline{\overline{N}}$ ,

(2)  $M \rightarrow_{\delta\mu} N$  implies  $\overline{\overline{M}} \equiv \overline{\overline{N}}$ .

*Proof.* The proposition is proved together with the claim  $M : K \rightarrow^* N : K$  for any  $K$  by induction on the definition of  $M \rightarrow N$ . We will prove only  $M : K \rightarrow^* N : K$ , from which  $\overline{\overline{M}} \rightarrow^* \overline{\overline{N}}$  follows.

The case of  $(\beta_\rightarrow)$ . Suppose that  $M \equiv (\lambda x. P)Q$  and  $N \equiv P[x := Q]$ . We have  $M : K \equiv (\lambda m. m \overline{\overline{Q}} K) (\lambda x. \overline{\overline{P}}) \rightarrow^+ \overline{\overline{P}}[x := \overline{\overline{Q}}] K \equiv (\lambda k. (P : k)[x := \overline{\overline{Q}}]) K \rightarrow^* (\lambda k. (P[x := Q] : k)) K$  (by Lemma 8)  $\rightarrow P[x := Q] : K$ .

Other cases of  $\beta$ -rules are similarly proved.

The case of  $(\delta)$ . Suppose that,  $M \equiv Q[x_1. P_1, x_2. P_2]\epsilon$  and  $N \equiv Q[x_1. P_1\epsilon, x_2. P_2\epsilon]$ . Then we have  $M : K \equiv Q : ([x_1. P_1, x_2. P_2] @ (\epsilon @ K)) \equiv Q : (\lambda m. m K_1 K_2)$ , where  $K_j \equiv \lambda x_j. (P_j : (\epsilon @ K))$ , which is identical with

$\lambda x_j.(P_j \epsilon : K)$ . Hence, we have  $M : K \equiv Q : ([x_1.P_1 \epsilon, x_2.P_2 \epsilon] @ K) \equiv Q[x_1.P_1 \epsilon, x_2.P_2 \epsilon] : K$ , which is identical to  $N : K$ .

The case of  $(\mu)$ . Suppose that  $M \equiv (\mu a.P)\epsilon$  and  $N \equiv \mu a.P[a \Leftarrow \epsilon]$ . This case is proved as follows:  $M : K \equiv \mu a.P : (\epsilon @ K) \equiv (P : \lambda x.x)[k_a := \epsilon @ k_a][k_a := K] \equiv (P[a \Leftarrow \epsilon] : \lambda x.x)[k_a := K]$  (by Lemma 8)  $\equiv \mu a.P[a \Leftarrow \epsilon] : K$ , where it should be noted that  $k_a \notin FV(\epsilon)$  and  $(\epsilon @ k_a)[k_a := K] \equiv \epsilon @ K$  hold.

Other cases are proved by the induction hypothesis and Lemma 9. For example, in the case of  $M_1 M_2 \rightarrow_\beta \overline{M_1} \overline{M_2}$  where  $M_2 \rightarrow_\beta N_2$  and  $M_1 M_2$  is typable and not vacuous, we have  $\overline{M_2} \rightarrow^+ \overline{N_2}$  by the induction hypothesis. Furthermore, the type of  $M_1$  is a functional type, which is not  $\perp$ , so we have  $k \in FV(M_1 : k)$  by Lemma 9 (2). Then we can prove this case by  $M_1 M_2 : K \equiv M_1 : \lambda m.m \overline{M_2} K \equiv (M_1 : k)[k := \lambda m.m \overline{M_2} K] \rightarrow^+ (M_1 : k)[k := \lambda m.m \overline{N_2} K] \equiv M_1 N_2 : K$ .  $\square$

## 4.2 Augmentations

For a  $\lambda\mu^{\rightarrow^{\wedge\vee\perp}}$ -term  $M$ , an *augmentation* of  $M$  will be defined as a non-vacuous  $\lambda\mu^{\rightarrow^{\wedge\vee\perp}}$ -term such that it is a  $\beta$ -expansion of  $M$  and any reduction step in  $\lambda\mu^{\rightarrow^{\wedge\vee\perp}}$  can be simulated by their augmentations. There is no erasing-continuation problem for the CPS-translation of augmentations to prove their strong normalization, since they are not vacuous.

**Definition 11 (Augmentations)** Fix a  $\lambda$ -variable  $c_\alpha$  for each type  $\alpha$ . For each  $\lambda\mu^{\rightarrow^{\wedge\vee\perp}}$ -term  $M$ , we define the set  $\text{Aug}(M)$  of augmentations in Figure 3. For each eliminator  $\epsilon$ , the set  $\text{Aug}(\epsilon)$  of its augmentations is defined by:  $\text{Aug}(\pi_j) = \{\pi_j\}$  and  $\text{Aug}([x.N, y.L]) = \{[x.N^+, y.L^+]\mid N^+ \in \text{Aug}(N), L^+ \in \text{Aug}(L)\}$ .

The augmentation of a term  $M$  is defined so that (1) it is a  $\beta$ -expansion of  $M$ , (2) it is not vacuous, and (3) augmentations simulate reductions of the original terms.

An augmentation of  $\mu a.M$  is not vacuous even if  $\mu a.M$  is vacuous, because we always have  $a$  in  $(\lambda z.M^+)(a(c_\alpha \bar{\epsilon}))$ . Since  $z$  is fresh, the augmentation of  $\mu a.M$   $\beta$ -reduces to  $\mu a.M$ . We need to have some flexibility in the part  $c_\alpha \bar{\epsilon}$  in  $\text{Aug}(\mu a.M)$  in order for augmentations to simulate reduction of the original terms, since  $(\mu a.(\lambda z.M^+)(a(c_\alpha \bar{\epsilon})))\epsilon$  reduces to  $\mu a.(\lambda z.(M^+)')(a(c_\alpha \bar{\epsilon}\epsilon))$ . For example,  $(\mu a.(\lambda z.x)(a(cy)))w$  reduces to  $\mu a.(\lambda z.x)(a(cyw))$ . When we think  $(\mu a.x)w \rightarrow \mu a.x$  and choose the augmentation  $(\mu a.(\lambda z.x)(a(cy)))w$  of  $(\mu a.x)w$ , we have to have the corresponding augmentation  $\mu a.(\lambda z.x)(a(cyw))$  of  $\mu a.x$  since  $(\mu a.(\lambda z.x)(a(cy)))w$  reduces to  $\mu a.(\lambda z.x)(a(cyw))$ . So we need the set of

$$\begin{aligned}
\text{Aug}(x) &= \{x\} \\
\text{Aug}(\lambda x.M) &= \{\lambda x.M^+ \mid M^+ \in \text{Aug}(M)\} \\
\text{Aug}(MN) &= \{M^+N^+ \mid M^+ \in \text{Aug}(M), N^+ \in \text{Aug}(N)\} \\
\text{Aug}(\langle M_1, N_2 \rangle) &= \{\langle M_1^+, M_2^+ \rangle \mid M_j^+ \in \text{Aug}(M_j)\} \\
\text{Aug}(M\pi_j) &= \{M^+\pi_j \mid M^+ \in \text{Aug}(M)\} \\
\text{Aug}(\iota_j M) &= \{\iota_j M^+ \mid M^+ \in \text{Aug}(M)\} \\
\text{Aug}(M[x_1.P_1, x_2.P_2]) &= \{M^+[x_1.P_1^+, x_2.P_2^+] \mid \\
&\quad M^+ \in \text{Aug}(M), P_j^+ \in \text{Aug}(P_j)\} \\
\text{Aug}(aM) &= \{aM^+ \mid M^+ \in \text{Aug}(M)\} \\
\text{Aug}(\mu a.M) &= \{\mu a.(\lambda z.M^+)(a(c_\alpha \bar{e})) \mid M^+ \in \text{Aug}(M), \\
&\quad z \text{ is not free in } M^+, c_\alpha \bar{e} \text{ is not vacuous}\}
\end{aligned}$$

Fig. 3. Augmentations of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -terms

augmentations instead of a single augmentation.  $c_\alpha$  is used to adjust a type according to the variable  $a$  and the sequence  $\bar{e}$  of eliminators so that some augmentations are typable. For example, when we think  $\mu a.x$  of type  $\alpha$  under  $x : \perp, a : \alpha$ , we have the typable augmentation  $\mu a.(\lambda z.x)(ac_\alpha)$  of type  $\alpha$  among its augmentations.

The logical effects of augmentations are as follows. (1) Some augmentations of a term are not typable even if the term is typed. We chose this definition for simplicity. We could define augmentations so that the augmentation of  $M$  always has the same type as that of  $M$ . (2) If an augmentation is typable, then the proof corresponding to a term  $M$  and the proof corresponding to the augmentation of  $M$  are equal up to proof normalization. (3) For any proof corresponding to  $M$ , there exists an augmentation of  $M$  such that the proof corresponding to the augmentation has the same end sequent as that of  $M$ .

**Lemma 12** *For any  $M$ , if  $N$  is in  $\text{Aug}(M)$ , then  $N$  is not vacuous.*

*Proof.* It is proved easily by induction on  $M$  from the definition of  $\text{Aug}(\mu a.M)$ .  $\square$

**Proposition 13** (1) *For any typable term, there exists a typable augmentation of the term. More precisely, if we have  $\Gamma \vdash M : \alpha; \Delta$ , there exist a type context  $\Pi$  which consists of type assignments of the form  $c_\beta : \beta$  and an augmentation  $M^+ \in \text{Aug}(M)$  such that  $\Gamma, \Pi \vdash M^+ : \alpha; \Delta$  is derivable.*

(2) *If  $M \rightarrow_\bullet N$  ( $\bullet = \beta, \delta$  or  $\mu$ ) and  $M^+ \in \text{Aug}(M)$  hold, then there exists  $N^+ \in \text{Aug}(N)$  such that  $M^+ \rightarrow_\bullet N^+$  holds.*

*Proof.* (1) By induction on the derivation of  $\Gamma \vdash M : \alpha; \Delta$ .

(2) We use the following lemmas: for any  $M_i, \epsilon, M_i^+ \in \text{Aug}(M_i)$  and  $\epsilon^+ \in \text{Aug}(\epsilon)$ , we have (i)  $M_1^+[x := M_2^+] \in \text{Aug}(M_1[x := M_2])$  and (ii)  $M_1^+[a \Leftarrow \epsilon^+] \in \text{Aug}(M_1[a \Leftarrow \epsilon])$ . These are easily proved by induction on  $M_1$ . Then the claim is proved by induction on  $M \rightarrow_\bullet N$ .

The case of  $(\beta_\rightarrow)$ . Suppose that  $M \equiv (\lambda x.P_1)P_2$  and  $N \equiv P_1[x := P_2]$ . Arbitrary  $M^+ \in \text{Aug}(M)$  is of the form  $(\lambda x.P_1^+)P_2^+$  for some  $P_i^+ \in \text{Aug}(P_i)$ . Then, we have  $M^+ \rightarrow_\beta P_1^+[x := P_2^+]$  and the right hand side is an augmentation of  $N \equiv P_1[x := P_2]$ .

Other cases of  $\beta$ -rules are similarly proved.

The case of  $(\delta)$ . Suppose that  $M \equiv Q[x_1.P_1, x_2.P_2]\epsilon$  and  $N \equiv Q[x_1.P_1\epsilon, x_2.P_2\epsilon]$ . Any  $M^+ \in \text{Aug}(M)$  is of the form  $M^+ \equiv Q^+[x_1.P_1^+, x_2.P_2^+]\epsilon^+$ . Then we have  $M^+ \rightarrow_\delta Q^+[x_1.P_1^+\epsilon^+, x_2.P_2^+\epsilon^+]$ , where the right hand side is an augmentation of  $N$ .

The case of  $(\mu)$ . Suppose that  $M \equiv (\mu a.P)\epsilon$  and  $N \equiv \mu a.P[a \Leftarrow \epsilon]$ . Any  $M^+ \in \text{Aug}(M)$  is of the form  $(\mu a.(\lambda z.P^+)(a(c\bar{\epsilon}_1)))\epsilon^+$ , then we have  $M^+ \rightarrow_\mu \mu a.(\lambda z.P^+[a \Leftarrow \epsilon^+])(a(c\bar{\epsilon}_1\epsilon^+))$ . Since  $c\bar{\epsilon}_1\epsilon^+$  is not vacuous, the right hand side is an augmentation of  $N$  by the above lemma.

Other cases are similarly proved.  $\square$

### 4.3 Strong normalization

We prove strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  by means of the CPS-translation and the augmentations. We use the strong normalization of the  $\delta\mu$ -reduction, which was proved in [5] by de Groote.

**Proposition 14 ([5])** *Any untyped  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term is strongly normalizing with respect to  $\delta\mu$ -reduction.*

*Proof.* As [5] proved, we can define some norm  $|M|$  as a positive integer, and prove that  $M \rightarrow_{\delta\mu} N$  implies  $|M| > |N|$ .  $\square$

**Theorem 15 (Strong normalization)** *Any typable  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term is strongly normalizing.*

*Proof.* Assume that there is an infinite  $\beta\delta\mu$ -reduction sequence  $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow \dots$ , where  $M_0$  is a typable  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term, and we will show contradiction. First, by Proposition 13, we have the infinite reduction sequence  $M_0^+ \rightarrow M_1^+ \rightarrow M_2^+ \rightarrow \dots$  of augmentations, where they are not vacuous and  $M_0^+$  is

typed. By Theorem 2, they are typable. Next, by taking the CPS-translation of each augmentation, we have the sequence  $\overline{\overline{M_0^+}} \rightarrow^* \overline{\overline{M_1^+}} \rightarrow^* \overline{\overline{M_2^+}} \rightarrow^* \dots$  by Proposition 10. If the reduction step  $M_i \rightarrow M_{i+1}$  is a  $\beta$ -step, we have  $\overline{\overline{M_i^+}} \rightarrow^+ \overline{\overline{M_{i+1}^+}}$  by Proposition 10 (1), and otherwise we have  $\overline{\overline{M_i^+}} \equiv \overline{\overline{M_{i+1}^+}}$  by Proposition 10 (2). Since the  $\delta\mu$ -reduction is strongly normalizing by Proposition 14, there are infinitely many  $i$ 's such that  $M_i \rightarrow M_{i+1}$  is a  $\beta$ -step. Therefore,  $\overline{\overline{M_0^+}} \rightarrow^* \overline{\overline{M_1^+}} \rightarrow^* \overline{\overline{M_2^+}} \rightarrow^* \dots$  is an infinite  $\beta$ -reduction sequence, which contradicts the strong normalization of the simply typed  $\lambda$ -calculus.  $\square$

## 5 General elimination rules

In this section, we define a system  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ , which is classical natural deduction with general elimination rules for conjunction and implication. We show the strong normalization of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  by using the CPS-translation and the augmentations.

### 5.1 The system $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$

The disjunction elimination rule of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  is similar to the left rule of sequent calculus. We can define elimination rules of conjunction and implication in the same manner. Such rules are called *general elimination rules* in [21]. The natural deduction with general elimination rules has more direct relationship with the sequent calculus than ordinary natural deduction. Von Plato gave a natural deduction system with general elimination rules, and a one-to-one correspondence between it and the sequent calculus. Furthermore, there is an isomorphism between the proof normalization in the natural deduction with the general elimination and a cut-elimination procedure in the sequent calculus [14]. [9,10] also studied normalization of an intuitionistic natural deduction with general elimination rule for implication.

The general elimination rules for conjunction and implication are given in Figure 4. They are a generalization of the ordinary elimination rules. The elimination rules in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  are derivable in  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  by the general elimination rules as follows:

$$\frac{\Gamma \vdash A \rightarrow B; \Delta \quad \Gamma \vdash A; \Delta \quad \overline{\Gamma, B \vdash B; \Delta}}{\Gamma \vdash B; \Delta} (\rightarrow E),$$

$$\begin{array}{c}
\frac{\Gamma \vdash M : \alpha \rightarrow \beta; \Delta \quad \Gamma \vdash N : \alpha; \Delta \quad \Gamma, x : \beta \vdash P : \gamma; \Delta}{\Gamma \vdash M[N, x.P] : \gamma; \Delta} (\rightarrow E) \\
\frac{\Gamma \vdash M : \alpha_1 \wedge \alpha_2; \Delta \quad \Gamma, x_1 : \alpha_1, x_2 : \alpha_2 \vdash P : \gamma; \Delta}{\Gamma \vdash M[(x_1, x_2).P] : \gamma; \Delta} (\wedge E)
\end{array}$$

Fig. 4. New Elimination rules of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$

$$\frac{\Gamma \vdash A_1 \wedge A_2; \Delta \quad \overline{\Gamma, A_1, A_2 \vdash A_j; \Delta}}{\Gamma \vdash A_j; \Delta} (\wedge E).$$

On the other hand, the general elimination rules are admissible in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

**Definition 16** ( $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ ) (1) Types of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  are the same as those of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

(2) Terms  $M, N, P$ , and eliminators  $\epsilon$  of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  are defined as follows:

$$\begin{aligned}
M, N, P &::= x \mid \lambda x.M \mid \langle M, N \rangle \mid \iota_j M \mid M\epsilon \mid aM \mid \mu a.M, \\
\epsilon &::= [N, x.P] \mid [(x_1, x_2).P] \mid [x_1.P_1, x_2.P_2].
\end{aligned}$$

$x$  in  $P$  is bound in  $[N, x.P]$ , and  $x_1$  and  $x_2$  in  $P$  are bound in  $[(x_1, x_2).P]$ .

(3) Typing rules are the same as  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  except elimination rules for implication and conjunction, which are given in Figure 4.

(4) Reduction rules of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  are given in Figure 5, where  $P[x_1 := M_1, x_2 := M_2]$  denotes a standard simultaneous substitution.

Note that  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  enjoys the subformula property and the subject reduction property.

## 5.2 CPS-translation

Now, we define a CPS-translation of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ , which is the same as that of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  except for the general elimination rules.

**Definition 17 (CPS-translation)** Suppose that we have some fixed  $\lambda$ -variable  $k_a$  for each  $\mu$ -variable  $a$ . The CPS-translation of a  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  term  $M$



$$\begin{array}{lll}
(\beta_{\rightarrow}) & (\lambda y.M)[N, x.P] & \rightarrow_{\beta} P[x := M[y := N]] \\
(\beta_{\wedge}) & \langle M_1, M_2 \rangle [(x_1, x_2).P] & \rightarrow_{\beta} P[x_1 := M_1, x_2 := M_2] \\
(\beta_{\vee}) & \iota_j M[x_1.P_1, x_2.P_2] & \rightarrow_{\beta} P_j[x_j := M] \\
\\ 
(\delta) & M\epsilon\epsilon' & \rightarrow_{\delta} M(\epsilon \star \epsilon') \\
\\ 
(\mu) & (\mu a.M)\epsilon & \rightarrow_{\mu} \mu a.M[a \Leftarrow \epsilon]
\end{array}$$

The composition  $\epsilon \star \epsilon'$  of eliminators is defined by

$$\begin{aligned}
[N, x.P] \star \epsilon' &\equiv [N, x.P\epsilon'] \\
[(x_1, x_2).P] \star \epsilon' &\equiv [(x_1, x_2).P\epsilon'] \\
[x_1.P_1, x_2.P_2] \star \epsilon' &\equiv [x_1.P_1\epsilon', x_2.P_2\epsilon']
\end{aligned}$$

Fig. 5. Reduction rules of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$

is defined as a  $\lambda$ -term  $\overline{\overline{M}} \equiv \lambda k.(M : k)$ , where  $M : K$  is defined by:

$$\begin{aligned}
x : K &\equiv xK, \\
\lambda x.M : K &\equiv K(\lambda x.\overline{\overline{M}}), \\
\langle M, N \rangle : K &\equiv K(\lambda x.x\overline{\overline{M}}\overline{\overline{N}}), \\
\iota_j M : K &\equiv K(\lambda y_1 y_2.y_j\overline{\overline{M}}), \\
M\epsilon : K &\equiv M : (\epsilon @ K), \\
aM : K &\equiv M : k_a, \\
\mu a.M : K &\equiv (M : \lambda x.x)[k_a := K],
\end{aligned}$$

where the composition  $\epsilon @ K$  is defined by

$$\begin{aligned}
[M, x.P] @ K &\equiv \lambda y.(\lambda x.(P : K))(y\overline{\overline{M}}), \\
[(x_1, x_2).P] @ K &\equiv \lambda y.y(\lambda x_1 x_2.(P : K)), \\
[x_1.P_1, x_2.P_2] @ K &\equiv \lambda y.y(\lambda x_1.(P_1 : K))(\lambda x_2.(P_2 : K)).
\end{aligned}$$

The definition of the double negation translation and vacuous terms are the same as Definition 5 and 7 respectively.

**Proposition 18** *For any typable  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  term  $M$ , the CPS-translation  $\overline{\overline{M}}$  is typable in the simply typed  $\lambda$ -calculus. More precisely, if  $\Gamma \vdash M : \alpha; \Delta$  is provable, then  $\overline{\overline{\Gamma}}, \neg\Delta^* \vdash \overline{\overline{M}} : \overline{\overline{\alpha}}$  is provable.*

*Proof.* By induction on the derivation of  $\Gamma \vdash M : \alpha; \Delta$ .  $\square$

Intuitively, we can consider an eliminator  $\epsilon$  as a part of a continuation, so  $\epsilon @ K$  can be seen as a composition of  $\epsilon$  and the continuation  $K$ . There is a relationship between the composition  $@$  of continuations and the composition  $\star$  of eliminators as follows.

**Lemma 19** *For any eliminators  $\epsilon$  and  $\epsilon'$ , and any  $\lambda$ -term  $K$ , we have  $(\epsilon \star \epsilon') @ K \equiv \epsilon @ (\epsilon' @ K)$ .*

*Proof.* By case analysis with respect to the form of  $\epsilon$ .  $\square$

By this lemma, we have  $M\epsilon\epsilon': K \equiv M:\epsilon @ (\epsilon' @ K) \equiv M:((\epsilon \star \epsilon') @ K) \equiv M(\epsilon \star \epsilon'): K$ , so the CPS-translation is invariant under permutative conversions. Indeed, as in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ , the translation preserves the  $\beta\delta\mu$ -reduction relation.

**Lemma 20** *For any terms  $P$ ,  $K$ ,  $K'$ ,  $M$  and any eliminator  $\epsilon$ , we have*

- (1)  $(P:K)[k := K'] \rightarrow^* P:K[k := K']$ ,
- (2)  $(P:K)[x := \overline{\overline{M}}] \rightarrow^* P[x := M]:K[x := \overline{\overline{M}}]$ ,
- (3)  $(P:K)[k_a := \epsilon @ k_a] \equiv P[a \Leftarrow \epsilon]:K[k_a := \epsilon @ k_a]$ .

*Proof.* By induction on  $P$ .  $\square$

**Lemma 21** *Let a  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ -term  $M$  be typable and not vacuous. For any  $\lambda$ -term  $K$ , we have the following.*

- (1) *If  $M$  contains a free  $\mu$ -variable  $a$  and the type of  $a$  is not  $\perp$ , then the free  $\lambda$ -variable  $k_a$  appears in  $M:K$ .*
- (2) *If the type of  $M$  is not  $\perp$ , then all free variables of  $K$  appear in  $M:K$ .*

*Proof.* This lemma is proved by induction on  $M$  similarly to Lemma 9.  $\square$

**Proposition 22** *For any  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ -term  $M$ ,  $M \rightarrow N$  implies  $\overline{\overline{M}} \rightarrow^* \overline{\overline{N}}$ . In particular, if  $M$  is typable and not vacuous, we have the following:*

- (1)  $M \rightarrow_\beta N$  implies  $\overline{\overline{M}} \rightarrow^+ \overline{\overline{N}}$ ,
- (2)  $M \rightarrow_{\delta\mu} N$  implies  $\overline{\overline{M}} \equiv \overline{\overline{N}}$ .

*Proof.* We can prove the proposition together with the claim  $M:K \rightarrow^* N:K$  for any  $K$  by induction on the definition of  $M \rightarrow N$ . They are proved similarly to Proposition 10 by using Lemma 20 and 21.  $\square$

### 5.3 Augmentations

Augmentations for  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  is defined similarly to  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

**Definition 23 (Augmentations)** For a  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ -term  $M$ , the definition of the set  $\text{Aug}(M)$  of augmentations of  $M$  is the same as that of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  except the additional definition for new terms and new eliminators given by:  $\text{Aug}(M\epsilon) = \{M^+\epsilon^+ \mid M^+ \in \text{Aug}(M), \epsilon^+ \in \text{Aug}(\epsilon)\}$ ,  $\text{Aug}([N.x.P]) = \{[N^+, x.P^+] \mid N^+ \in \text{Aug}(N), P^+ \in \text{Aug}(P)\}$ ,  $\text{Aug}([(x_1, x_2).P]) = \{[(x_1, x_2).P^+] \mid P^+ \in \text{Aug}(P)\}$ .

**Lemma 24** Any augmentation  $M^+ \in \text{Aug}(M)$  of any term  $M$  is not vacuous.

*Proof.* It is proved by induction on  $M$  from the definition of  $\text{Aug}(\mu a.M)$ .  $\square$

**Proposition 25** (1) For any typable term, there exists a typable augmentation of the term. More precisely, if  $\Gamma \vdash M : \alpha; \Delta$  holds, there exist a context  $\Pi$ , whose elements are of the form  $c_\beta : \beta$ , and an augmentation  $M^+$  of  $M$  such that  $\Gamma, \Pi \vdash M^+ : \alpha; \Delta$  is derivable.

(2) If we have  $M \rightarrow_\bullet N$  ( $\bullet = \beta, \delta$  or  $\mu$ ) and  $M^+ \in \text{Aug}(M)$ , there exists  $N^+ \in \text{Aug}(N)$  such that  $M^+ \rightarrow_\bullet N^+$  holds.

*Proof.* They are similarly proved to Proposition 13.  $\square$

### 5.4 Strong normalization

In this subsection, we prove the strong normalization of  $\delta\mu$ -reduction for untyped  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ -terms. In order to prove it, we extend the norm of terms, which is defined by de Groote in [5] for  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ . By this result, we prove the strong normalization of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ .

**Definition 26 (Norm)** For a  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ -term  $M$  and an eliminator  $\epsilon$ , we define the norm  $|M|$  and  $|\epsilon|$ . In order for that, we also define the auxiliary notions  $\#M$ ,  $[M]_a$ ,  $\#\epsilon$ , and  $[\epsilon]_a$  simultaneously.

(1)  $|M|$  and  $|\epsilon|$  are defined by:

$$\begin{aligned} |x| &= 1, & |\lambda x.M| &= |M|, \\ |\langle M, N \rangle| &= |M| + |N|, & |\iota_j M| &= |M|, \\ |\mu a.M| &= |M|, & |aM| &= |M|, \\ |M\epsilon| &= |M| + \#M|\epsilon|, \end{aligned}$$

$$|[N, x.P]| = |N| + |P|, \quad |[ (x, y).P ]| = |P|, \quad |[x_1.P_1, x_2.P_2]| = |P_1| + |P_2|.$$

(2)  $\#M$  and  $\#\epsilon$  are defined by:

$$\begin{aligned} \#x &= 1, & \#\lambda x.M &= 1, \\ \#\langle M, N \rangle &= 1, & \#\iota_j M &= 1, \\ \#\mu a.M &= \lfloor M \rfloor_a + 1, & \#(aM) &= 1, \\ \#(M\epsilon) &= 2\#M\#\epsilon, \end{aligned}$$

$$\#[N, x.P] = \#P, \quad \#[(x, y).P] = \#P, \quad \#[x_1.P_1, x_2.P_2] = \#P_1 + \#P_2.$$

(3)  $\lfloor M \rfloor_a$  and  $\lfloor \epsilon \rfloor_a$  are defined by:

$$\begin{aligned} \lfloor x \rfloor_a &= 0, & \lfloor \lambda x.M \rfloor_a &= \lfloor M \rfloor_a, \\ \lfloor \langle M, N \rangle \rfloor_a &= \lfloor M \rfloor_a + \lfloor N \rfloor_a, & \lfloor \iota_j M \rfloor_a &= \lfloor M \rfloor_a, \\ \lfloor \mu b.M \rfloor_a &= \lfloor M \rfloor_a, & \lfloor bM \rfloor_a &= \lfloor M \rfloor_a \quad (a \neq b), \\ \lfloor M\epsilon \rfloor_a &= \lfloor M \rfloor_a + \#M\lfloor \epsilon \rfloor_a, & \lfloor aM \rfloor_a &= \lfloor M \rfloor_a + \#M, \\ \lfloor [N, x.P] \rfloor_a &= \lfloor N \rfloor_a + \lfloor P \rfloor_a, & \lfloor [(x, y).P] \rfloor_a &= \lfloor P \rfloor_a, \\ \lfloor [x_1.P_1, x_2.P_2] \rfloor_a &= \lfloor P_1 \rfloor_a + \lfloor P_2 \rfloor_a. \end{aligned}$$

**Lemma 27** *Let a meta-variable  $A$  denote either a term or an eliminator. We have the following:*

- (1)  $|A| \geq 1$ ,  $\#A \geq 1$  and  $\lfloor A \rfloor_a \geq 0$ ,
- (2) if  $a \notin FV(A)$ ,  $\lfloor A \rfloor_a = 0$  holds.

*Proof.* They are easily proved from the definitions.  $\square$

**Lemma 28** *Let  $A$  denote either a term or an eliminator. If  $a \notin FV(\epsilon)$  holds, then we have the following:*

- (1)  $\#A[a \leftarrow \epsilon] = \#A$ ,
- (2)  $\lfloor A[a \leftarrow \epsilon] \rfloor_b = \lfloor A \rfloor_b + \lfloor A \rfloor_a \lfloor \epsilon \rfloor_b$  for  $a \neq b$ ,
- (3)  $\lfloor A[a \leftarrow \epsilon] \rfloor_a = 2\lfloor A \rfloor_a \#\epsilon$ ,
- (4)  $|A[a \leftarrow \epsilon]| = |A| + \lfloor A \rfloor_a |\epsilon|$ .

*Proof.* We will prove (1) and (2) simultaneously by induction on  $A$ .

(1) The case of  $\mu b.M$ . We can suppose  $a \neq b$  and  $b \notin FV(\epsilon)$  by renaming bound variables.  $\#(\mu b.M)[a \leftarrow \epsilon] = \lfloor M[a \leftarrow \epsilon] \rfloor_b + 1 = \lfloor M \rfloor_b + \lfloor M \rfloor_a \lfloor \epsilon \rfloor_b + 1$  (by the induction hypothesis for (2))  $= \lfloor M \rfloor_b + 1$  (by  $\lfloor \epsilon \rfloor_b = 0$ )  $= \# \mu b.M$ . Other cases are similarly proved.

(2) The case of  $aM$ .  $\lfloor (aN)[a \leftarrow \epsilon] \rfloor_b = \lfloor a(N[a \leftarrow \epsilon]\epsilon) \rfloor_b = \lfloor N[a \leftarrow \epsilon]\epsilon \rfloor_b = \lfloor N[a \leftarrow \epsilon] \rfloor_b + \#N[a \leftarrow \epsilon]\lfloor \epsilon \rfloor_b = (\lfloor N \rfloor_b + \lfloor N \rfloor_a \lfloor \epsilon \rfloor_b) + \#N\lfloor \epsilon \rfloor_b$  (by the induction hypotheses for (1) and (2))  $= \lfloor N \rfloor_b + (\lfloor N \rfloor_a + \#N)\lfloor \epsilon \rfloor_b = \lfloor aN \rfloor_b + \lfloor aN \rfloor_a \lfloor \epsilon \rfloor_b$ . Other cases are similarly proved.

(3) The case of  $aM$ . Note that  $\lfloor \epsilon \rfloor_a = 0$  holds since  $a \notin FV(\epsilon)$ .  $\lfloor (aN)[a \leftarrow \epsilon] \rfloor_a = \lfloor a(N[a \leftarrow \epsilon]\epsilon) \rfloor_a = \lfloor N[a \leftarrow \epsilon]\epsilon \rfloor_a + \#(N[a \leftarrow \epsilon]\epsilon) = \lfloor N[a \leftarrow \epsilon] \rfloor_a + 2\#\epsilon\#N[a \leftarrow \epsilon]$  (by  $\lfloor \epsilon \rfloor_a = 0$ )  $= 2\lfloor N \rfloor_a \#\epsilon + 2\#\epsilon\#N$  (by (1) and the induction hypothesis)  $= 2\lfloor aN \rfloor_a \#\epsilon$ . Other cases are similarly proved.

(4) The case of  $aM$ .  $\lfloor (aN)[a \leftarrow \epsilon] \rfloor = \lfloor a(N[a \leftarrow \epsilon]\epsilon) \rfloor = \lfloor N[a \leftarrow \epsilon]\epsilon \rfloor = \lfloor N[a \leftarrow \epsilon] \rfloor + \#N[a \leftarrow \epsilon]\lfloor \epsilon \rfloor = \lfloor N \rfloor + \lfloor N \rfloor_a \lfloor \epsilon \rfloor + \#N\lfloor \epsilon \rfloor$  (by (1) and the induction hypothesis)  $= \lfloor aN \rfloor + \lfloor aN \rfloor_a \lfloor \epsilon \rfloor$ . Other cases are similarly proved.  $\square$

**Proposition 29** *If  $M \rightarrow_{\delta\mu} N$ , then  $|M| > |N|$  holds.*

*Proof.* We use the following lemmas: (1)  $|\epsilon_1 \star \epsilon_2| = |\epsilon_1| + \#\epsilon_1|\epsilon_2|$ , (2)  $\#(\epsilon_1 \star \epsilon_2) = 2\#\epsilon_1\#\epsilon_2$ , and (3)  $\lfloor \epsilon_1 \star \epsilon_2 \rfloor_a = \lfloor \epsilon_1 \rfloor_a + \#\epsilon_1\lfloor \epsilon_2 \rfloor_a$  hold. These are proved by case analysis with respect to the form of  $\epsilon_1$ . By these lemmas, we prove  $|M| > |N|$ ,  $\#M \geq \#N$  and  $\lfloor M \rfloor_a \geq \lfloor N \rfloor_a$  by induction on the definition of  $M \rightarrow_{\delta\mu} N$ . Note that we have to prove  $\#M \geq \#N$  and  $\lfloor M \rfloor_a \geq \lfloor N \rfloor_a$  simultaneously.

The case of  $(\delta)$ . Suppose that  $M \equiv P\epsilon\epsilon'$  and  $N \equiv P(\epsilon \star \epsilon')$ . We have  $|P\epsilon\epsilon'| = |P\epsilon| + \#(P\epsilon)|\epsilon'| = |P| + \#P|\epsilon| + 2\#P\#\epsilon|\epsilon'| > |P| + \#P(|\epsilon| + \#\epsilon|\epsilon'|)$  (by  $\#P\#\epsilon|\epsilon'| \geq 1$ )  $= |P| + \#P|\epsilon \star \epsilon'|$  (by the above lemma)  $= |P(\epsilon \star \epsilon')|$ . We also have  $\#(P\epsilon\epsilon') = 4\#P\#\epsilon\#\epsilon' = 2\#P\#(\epsilon \star \epsilon')$  (by the above lemma)  $= \#(P(\epsilon \star \epsilon'))$ . We also have  $\lfloor P\epsilon\epsilon' \rfloor_a = \lfloor P\epsilon \rfloor_a + \#(P\epsilon)\lfloor \epsilon' \rfloor_a = \lfloor P \rfloor_a + \#P\lfloor \epsilon \rfloor_a + 2\#P\#\epsilon\lfloor \epsilon' \rfloor_a \geq \lfloor P \rfloor_a + \#P(\lfloor \epsilon \rfloor_a + \#\epsilon\lfloor \epsilon' \rfloor_a)$  (by  $\#P\#\epsilon\lfloor \epsilon' \rfloor_a \geq 0$ )  $= \lfloor P \rfloor_a + \#P\lfloor \epsilon \star \epsilon' \rfloor_a$  (by the above lemma)  $= \lfloor P(\epsilon \star \epsilon') \rfloor_a$ .

The case of  $(\mu)$ . Suppose that  $M \equiv (\mu a.P)\epsilon$  and  $N \equiv \mu a.P[a \leftarrow \epsilon]$ . We have  $|(\mu a.P)\epsilon| = |\mu a.P| + \#\mu a.P|\epsilon| = |P| + (\lfloor P \rfloor_a + 1)|\epsilon| > |P| + \lfloor P \rfloor_a |\epsilon|$  (by  $|\epsilon| \geq 1$ )  $= |P[a \leftarrow \epsilon]|$  (by Lemma 28 (4))  $= |\mu a.P[a \leftarrow \epsilon]|$ . We also have  $\#((\mu a.P)\epsilon) = 2\#\mu a.P\#\epsilon = 2(\lfloor P \rfloor_a + 1)\#\epsilon > 2\lfloor P \rfloor_a \#\epsilon + 1$  (by  $\#\epsilon \geq 1$ )  $= \lfloor P[a \leftarrow \epsilon] \rfloor_a + 1$  (by Lemma 28 (3))  $=$

$\# \mu a.P[a \Leftarrow \epsilon]$ . We also have  $\lfloor (\mu a.P)\epsilon \rfloor_b = \lfloor \mu a.P \rfloor_b + \# \mu a.P \lfloor \epsilon \rfloor_b = \lfloor P \rfloor_b + (\lfloor P \rfloor_a + 1) \lfloor \epsilon \rfloor_b \geq \lfloor P \rfloor_b + \lfloor P \rfloor_a \lfloor \epsilon \rfloor_b$  (by  $\lfloor \epsilon \rfloor_b \geq 0$ )  $= \lfloor P[a \Leftarrow \epsilon] \rfloor_b$  (by Lemma 28 (2))  $= \lfloor \mu a.P[a \Leftarrow \epsilon] \rfloor_b$ .

Other cases are proved by the induction hypothesis.  $\square$

**Theorem 30** *Any untyped  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  term is strongly normalizing with respect to  $\delta\mu$ -reduction.*

*Proof.* By Proposition 29.  $\square$

Finally we have the result.

**Theorem 31** *Any typable  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ -term is strongly normalizing.*

*Proof.* It is proved similarly to Theorem 15 by Proposition 22 and 25, Theorem 30, and the subject reduction property.  $\square$

## 6 Embedding from $\lambda\mu^{\rightarrow\wedge\vee\perp}$ to $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$

In this section, we will give an embedding from  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  and show that this embedding strictly preserves reduction steps. This will show that general elimination rules are really generalization of ordinary elimination rules. Hence we can reduce strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to that of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ .

**Definition 32** *We define a map  $M^*$  from  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  by:*

$$\begin{aligned} x^* &\equiv x, \\ (\lambda x.M)^* &\equiv \lambda x.M^*, \\ \langle M, N \rangle^* &\equiv \langle M^*, N^* \rangle, \\ (\iota_j M)^* &\equiv \iota_j M^*, \\ (M\epsilon)^* &\equiv M^* \epsilon^\dagger, \\ (\mu a.M)^* &\equiv \mu a.M^*, \\ (aM)^* &\equiv aM^*, \end{aligned}$$

where the map  $\epsilon^\dagger$  from eliminators in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  to eliminators in  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  is defined by:

$$\begin{aligned} N^\dagger &\equiv [N^*, x.x], \\ \pi_j^\dagger &\equiv [(x_1, x_2).x_j], \\ [x_1.P_1, x_2.P_2]^\dagger &\equiv [x_1.P_1^*, x_2.P_2^*]. \end{aligned}$$

**Lemma 33** *For any terms  $M, N$  and any eliminator  $\epsilon$ , we have the following.*

$$(1) (M[x := N])^* \equiv M^*[x := N^*],$$

$$(2) (M[a \Leftarrow \epsilon])^* \equiv M^*[a \Leftarrow \epsilon^\dagger].$$

*Proof.* It is proved straightforwardly by induction on  $M$ . The case of  $aM$  for (2) is proved as follows. By the induction hypothesis, we have  $(M[a \Leftarrow \epsilon])^* \equiv M^*[a \Leftarrow \epsilon^\dagger]$ . Therefore we have  $((aM)[a \Leftarrow \epsilon])^* \equiv a((M[a \Leftarrow \epsilon])^* \epsilon^\dagger) \equiv a((M^*[a \Leftarrow \epsilon^\dagger]) \epsilon^\dagger) \equiv (aM^*)[a \Leftarrow \epsilon^\dagger] \equiv (aM)^*[a \Leftarrow \epsilon^\dagger]$ .  $\square$

**Proposition 34** (1) If  $\Gamma \vdash M : \alpha; \Delta$  is derivable in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ ,  $\Gamma \vdash M^* : \alpha; \Delta$  is derivable in  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ .

(2) If  $M \rightarrow_\bullet N$  ( $\bullet = \beta, \delta$  or  $\mu$ ) in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ , we have  $M^* \rightarrow_\bullet N^*$  in  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ .

*Proof.* (1) It is proved by induction on the derivation of  $\Gamma \vdash M : \alpha; \Delta$  in  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ .

(2) By induction on the definition of  $M \rightarrow_\bullet N$ . In the case where  $M$   $\beta$ -reduces to  $N$  and its redex is  $M$ , the claim is proved by Lemma 33 (1), as follows. For  $(\beta_\rightarrow)$ , we have  $((\lambda x.M)N)^* \equiv (\lambda x.M^*)[N^*, y.y] \rightarrow_\beta y[y := M^*[x := N^*]] \equiv (M[x := N])^*$ . For  $(\beta_\wedge)$ , we also have  $(\langle M_1, M_2 \rangle \pi_j)^* \equiv \langle M_1^*, M_2^* \rangle [(x_1, x_2).x_j] \rightarrow_\beta x_j[x_1 := M_1^*, x_2 := M_2^*] \equiv M_j^*$ . For  $(\beta_\vee)$ , we also have  $((\iota_j M)[x_1.P_1, x_2.P_2])^* \equiv (\iota_j M^*)[x_1.P_1^*, x_2.P_2^*] \rightarrow_\beta P_j^*[x_j := M^*] \equiv (P_j[x_j := M])^*$ . The case of  $\delta$ -reduction is proved by  $(M[x_1.P_1, x_2.P_2]\epsilon)^* \equiv M^*[x_1.P_1^*, x_2.P_2^*]\epsilon^\dagger \rightarrow_\delta M^*[x_1.P_1^*\epsilon^\dagger, x_2.P_2^*\epsilon^\dagger] \equiv M^*[x_1.(P_1\epsilon)^*, x_2.(P_2\epsilon)^*] \equiv (M[x_1.P_1\epsilon, x_2.P_2\epsilon])^*$ . The case of  $\mu$ -reduction is proved from Lemma 33 (2) by  $((\mu a.M)\epsilon)^* \equiv (\mu a.M^*)\epsilon^\dagger \rightarrow_\mu \mu a.M^*[a \Leftarrow \epsilon^\dagger] \equiv (\mu a.M[a \Leftarrow \epsilon])^*$ . The induction steps are easily proved from the induction hypothesis.  $\square$

From this proposition, an infinite reduction sequence of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  is embedded to an infinite sequence of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  by the map  $(\cdot)^*$ . Hence, the strong normalization of  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  is also proved as a corollary of the strong normalization of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ .

## 7 Concluding remarks

The strong normalization of the intuitionistic natural deduction with permutative conversions was proved in [6,12,3,20]. De Groote [6] gave a CPS-translation from the intuitionistic natural deduction with conjunction, disjunction and implication to that with only implication, which is the simply typed  $\lambda$ -calculus. This is based on the same idea of ours and [5]. There was no difficulty due to erasing-continuation, because his system has no control operators. Tatsuta and Mints [20] proved the strong normalization of the second-order natural deduction by using atomic disjunctions and extending saturated

sets.

The strong normalization of the classical natural deduction with permutative conversions was already proved in [3,12]. David and Nour [3] proved the strong normalization of a system logically equivalent to  $\lambda\mu^{\rightarrow\wedge\vee\perp}$  by using only simple notions. Matthes [12] proved strong normalization of classical natural deduction with second-order universal quantification and disjunction by extending the reducibility candidates. Our method used the CPS-translation to give another proof of strong normalization of classical natural deduction with permutative conversion. We hope our proof will contribute to better understanding of the calculus. Our method can also be applied to second-order universal and existential quantifiers.

For the strong normalization of the natural deduction with general elimination rules, Joachimski and Matthes [10] proved it for intuitionistic logic with only implication by giving an inductive definition of strongly normalizing terms. In this paper, we gave the first strong normalization proof for classical natural deduction with general elimination rules for conjunction and disjunction.

General elimination rules are useful in the study of the relationship between the natural deduction and the sequent calculus, because general elimination rules are similar to left rules of the sequent calculus. For computational meaning of the sequent calculus, some term assignment systems have been proposed such as  $\bar{\lambda}\mu\tilde{\mu}$  by Curien and Herbelin [2] and the dual calculus by Wadler [22]. Furthermore, a correspondence to a low-level machine code has been shown by Ohori in [16]. However, the computational aspect of the sequent calculus has not been fully clarified yet. On the other hand, computational meaning of the natural deduction is deeply studied through the Curry-Howard isomorphism. General elimination rules are also relatively easy to understand. For example, the  $\beta$ -rule for implication  $(\lambda x.M)[N, y.P] \rightarrow P[y := M[x := N]]$  can be understood as the computation in which the function  $\lambda x.M$  is applied to the argument  $N$ , and then the result of the application is passed to the continuation  $y.P$ . The CPS-translation given in this paper realizes this idea. As discussed above, the relationship between natural deduction and sequent calculus helps us to study the computational aspect of sequent calculus. For this purpose, we have to give correspondence not only for normal proofs, but also for non-normal proofs, and consider the relationship between proof normalization and cut-elimination procedure. [14] studied this for intuitionistic logic. This study for classical logic would be a future work.

Confluence is another important property of calculi. In order to prove the confluence of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$ , due to the strong normalization result in this paper, it is sufficient to prove weak confluence, which we can prove in a straightforward way. On the other hand, if we cannot use the strong normalization, the confluence of  $\lambda\mu_g^{\rightarrow\wedge\vee\perp}$  is not so easy



to prove, because it is not straightforward to extend the well-known parallel reduction method in [19] to calculi with permutative conversions. For example, let  $M$  be the term  $N[x_1.P_1, x_2, P_2][y_1.Q_1, y_2, Q_2]\epsilon$ , and then we have  $M \rightarrow_\delta N[x_1.P_1, x_2, P_2][y_1.Q_1\epsilon, y_2, Q_2\epsilon]$  and  $M \rightarrow_\delta N[x_1.P_1[y_1.Q_1, y_2, Q_2], x_2, P_2[y_1.Q_1, y_2, Q_2]]\epsilon$ , but there is no term that these terms are reduced to by one-step ordinary parallel reduction. Andou [1] overcame this difficulty for the classical natural deduction with disjunction by extending the parallel reduction with the notion of segment-trees. Joachimski and Matthes [9] proved confluence of the intuitionistic natural deduction with permutative conversions by commutativity of the permutative conversion and the parallel  $\beta$ -reduction.

An inverse map of the embedding defined in the previous section can be defined by:

$$\begin{aligned}
x_* &\equiv x, \\
(\lambda x.M)_* &\equiv \lambda x.M_*, \\
\langle M, N \rangle_* &\equiv \langle M_*, N_* \rangle, \\
(\iota_j M)_* &\equiv \iota_j M_*, \\
(M[N, x.P])_* &\equiv P_*[x := M_*N_*], \\
(M[(x_1, x_2).P])_* &\equiv P_*[x_1 := M_*\pi_1, x_2 := M_*\pi_2], \\
(M[x_1.P_1, x_2.P_2])_* &\equiv M_*[x_1.P_{1*}, x_2.P_{2*}], \\
(\mu a.M)_* &\equiv \mu a.M_*, \\
(aM)_* &\equiv aM_*.
\end{aligned}$$

In fact, this map is a left inverse of the embedding, that is, we have  $(M^*)_* \equiv M$  for any  $\lambda\mu^{\rightarrow\wedge\vee\perp}$ -term  $M$ . This map also preserves typability and the  $\beta\delta$ -reduction, but may fail to preserve the  $\mu$ -reduction. If we restrict the domain of the map  $(\cdot)_*$  to the image of  $(\cdot)^*$ , any reduction  $\rightarrow_\bullet$  is preserved. That is, if we have  $M^* \rightarrow_\bullet N^*$ , then  $M \rightarrow_\bullet N$ . More detailed investigation of relationship between general and ordinary elimination rules will be another future work.

## Acknowledgment

The authors are grateful to Professor Philippe de Groote and Tatsuya Abe for their fruitful discussions.

## References

- [1] Y. Andou. Church-Rosser Property of a simple reduction for full first-order classical natural deduction. *Annals of Pure and Applied Logic* 119, pp.225–237,

2003.

- [2] P.-L. Curien and H. Herbelin. The Duality of Computation. In *Proc. 5th International Conference on Functional Programming (ICFP '00)*, pp. 233–243, 2000.
- [3] R. David and K. Nour. A short proof of the strong normalization of classical natural deduction with disjunction. *Journal of Symbolic Logic* 68(4), pp. 1277–1288, 2003.
- [4] P. de Groote. A simple calculus of exception handling. In M. Dezani-Ciancaglini and G. Plotkin, Eds., *Typed Lambda Calculi and Applications (TLCA '95)*, LNCS 902, pp. 201–215, 1995.
- [5] P. de Groote. Strong normalization of classical natural deduction with disjunction. In S. Abramsky, Ed., *Typed Lambda Calculi and Applications (TLCA '01)*, LNCS 2044, pp. 182–196, 2001.
- [6] P. de Groote. On the Strong Normalisation of Intuitionistic Natural Deduction with Permutation-Conversions. *Information and Computation* 178, pp. 441–464, 2002.
- [7] K. Fujita. Domain-free  $\lambda\mu$ -calculus. *Theoretical Informatics and Applications* 34(6), pp. 433–466, 2000.
- [8] T. Griffin. A formulae-as-types notion of control. In *17th Ann. ACM Symp. on Principles of Programming Languages (POPL '90)*, pp. 47–58, 1990.
- [9] F. Joachimski and R. Matthes. Standardization and Confluence for a Lambda Calculus with Generalized Applications. In L. Bachmair, Ed., *Rewriting Techniques and Applications (RTA '00)*, LNCS 1833, pp. 141–155, 2000.
- [10] F. Joachimski and R. Matthes. Short proofs of normalization for the simply-typed  $\lambda$ -calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic* 42, pp. 59–87, 2003.
- [11] R. Matthes. Stabilization — an alternative to double-negation translation for classical natural deduction. In V. Stoltenberg-Hansen and J. Väänänen, Eds., *Proc. Logic Colloquium '03, Lecture Notes in Logic* 24, pp. 167–199, 2003.
- [12] R. Matthes. Non-strictly positive fixed-points for classical natural deduction. *Annals of Pure and Applied Logic* 133, pp. 205–130, 2005.
- [13] K. Nakazawa and M. Tatsuta. Strong normalization proof with CPS-translation for second order classical natural deduction. *Journal of Symbolic Logic* 68(3), pp. 851–859, 2003. Corrigendum is available in *Journal of Symbolic Logic* 68(4), pp. 1415–1416, 2003.
- [14] K. Nakazawa. An isomorphism between cut-elimination procedure and proof reduction. In S. Ronchi Della Rocca, Ed., *Typed Lambda Calculi and Applications (TLCA '07)*, LNCS 4583, pp. 336–350, 2007.

- [15] K. Nour and K. Saber. A semantics of realisability for the classical propositional natural deduction. *Electronic Notes in Theoretical Computer Science* 140, pp. 31–39, 2005.
- [16] A. Ohori. The Logical Abstract Machine: a Curry-Howard isomorphism for machine code. In A. Middeldorp and T. Sato, Eds., *Proc. Fuji International Symposium on Functional and Logic Programming (FLOPS '01)*, LNCS 1722, pp. 300–318, 1999.
- [17] M. Parigot.  $\lambda\mu$ -calculus: an algorithmic interpretation of classical natural deduction. In A. Voronkov, Ed., *Proc. International Conference on Logic Programming and Automated Reasoning (LPAR '92)*, LNCS 624, pp. 190–201, 1992.
- [18] M. Parigot. Strong normalization for second order classical natural deduction. *Journal of Symbolic Logic* 62(4), pp. 1461–1479, 1997.
- [19] M. Takahashi. Parallel reduction in  $\lambda$ -calculus. *Information and Computation* 118(1), pp. 120–127, 1995.
- [20] M. Tatsuta and G. Mints. A simple proof of second-order strong normalization with permutative conversions. *Annals of Pure and Applied Logic* 136, pp. 134–155, 2005.
- [21] J. von Plato. Natural deduction with general elimination rules. *Annals of Mathematical Logic* 40(7), pp. 541–567, 2001.
- [22] P. Wadler. Call-by-value is dual to call-by-name. In C. Runciman and O. Shivers, Eds., *Proc. International Conference on Functional Programming (ICFP '03)*, pp. 189–201, 2003.
- [23] Y. Yamagata. Strong Normalization of Second Order Symmetric Lambda-mu Calculus. *Information and Computation* 193(1), pp. 1–20, 2004.