Available online at www.sciencedirect.com

ScienceDirect

Annals of Pure and Applied Logic 153 (2008) 21–37

ANNALS OF PURE AND APPLIED LOGIC

www.elsevier.com/locate/apal

Strong normalization of classical natural deduction with disjunctions

Koji Nakazawa[a,*], Makoto Tatsuta[b]

[a] Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan
[b] National Institute of Informatics, 2-1-2 Hitotsubashi, Tokyo, 101-8430, Japan

Available online 8 February 2008

Abstract

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

MSC: 03B05; 03F05

Keywords: 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 the 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].

The subformula property is important in natural deduction. In the intuitionistic natural deduction, normal proofs can be obtained by proof normalization, which corresponds to the β-reduction in the simply typed λ-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 said to be 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, β-normal proofs may fail

* Corresponding author.
E-mail addresses: knak@kuis.kyoto-u.ac.jp (K. Nakazawa), tatsuta@nii.ac.jp (M. Tatsuta).

0168-0072/$ - see front matter
doi:10.1016/j.apal.2008.01.003

to enjoy the subformula property for the following reason. The form of the disjunction elimination rule is

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

so the conclusion γ may not be a subformula of the major premise $\alpha \vee \beta$. When the premise γ is produced by an introduction rule, and the conclusion γ is consumed by an elimination rule, the formula γ 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

$$\frac{\dfrac{\begin{matrix}\vdots\ M \\ \Gamma \vdash \alpha \vee \beta\end{matrix} \quad \begin{matrix}\vdots\ P_1 \\ \Gamma, \alpha \vdash \gamma_1 \to \gamma_2\end{matrix} \quad \begin{matrix}\vdots\ P_2 \\ \Gamma, \beta \vdash \gamma_1 \to \gamma_2\end{matrix}}{\Gamma \vdash \gamma_1 \to \gamma_2}\ (\vee \mathrm{E}) \qquad \begin{matrix}\vdots\ N \\ \Gamma \vdash \gamma_1\end{matrix}}{\Gamma \vdash \gamma_2}\ (\to \mathrm{E})$$

reduces to the proof

$$\frac{\begin{matrix}\vdots\ M \\ \Gamma \vdash \alpha \vee \beta\end{matrix} \quad \dfrac{\begin{matrix}\vdots\ P_1 \\ \Gamma, \alpha \vdash \gamma_1 \to \gamma_2\end{matrix} \quad \begin{matrix}\vdots\ N \\ \Gamma \vdash \gamma_1\end{matrix}}{\Gamma, \alpha \vdash \gamma_2}\ (\to \mathrm{E}) \quad \dfrac{\begin{matrix}\vdots\ P_2 \\ \Gamma, \beta \vdash \gamma_1 \to \gamma_2\end{matrix} \quad \begin{matrix}\vdots\ N \\ \Gamma \vdash \gamma_1\end{matrix}}{\Gamma, \beta \vdash \gamma_2}\ (\to \mathrm{E})}{\Gamma \vdash \gamma_2}\ (\vee \mathrm{E}).$$

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 β-reduction and the permutative conversions. This discussion can be extended to the classical natural deduction with the μ-rule, as de Groote showed in [5].

In [5], de Groote introduced the calculus $\lambda\mu^{\to\wedge\vee\perp}$, 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 $\lambda\mu^{\to\wedge\vee\perp}$. The proof, however, contains an error as Matthes pointed out in [11].

In this paper, we prove the strong normalization of $\lambda\mu^{\to\wedge\vee\perp}$ 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 that 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 $\lambda\mu^{\to\wedge\vee\perp}$.

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 μ-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 β-redex and N' is its β-reduct, a CPS-translation maps $(\mu a.M)N$ and $(\mu a.M)N'$ to the same term. Hence β-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 said to be *vacuous* when the variable a does not occur in M. However, this definition did not work in fact since it failed to preserve β-reduction.

In order to recover his proof, we use augmentations. For each term M, we define a set $\mathrm{Aug}(M)$ of its augmentations where the augmentation is a β-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 $\mathrm{Aug}(M)$, then there exists some N^+ in $\mathrm{Aug}(N)$ 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 $\lambda^{\to\wedge\vee\perp}$.

General elimination rules have been studied [9,21,10,14]. For conjunction and implication, we can consider the elimination rules to be 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 elimination 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 β-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 the 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 $\lambda\mu_g^{\to\wedge\vee\bot}$ with general elimination rules, and proves the strong normalization of $\lambda\mu_g^{\to\wedge\vee\bot}$ by means of a CPS-translation and the idea of augmentations. We show that the augmentations can work also for general elimination rules of implication, conjunction, and their permutative conversions. We also show the strong normalization of untyped $\lambda\mu_g^{\to\wedge\vee\bot}$ with respect to μ-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 they have not been proved for the classical case yet.

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

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

2. Definition of the system $\lambda\mu^{\to\wedge\vee\bot}$

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

Definition 1 ($\lambda\mu^{\to\wedge\vee\bot}$). We define $\lambda\mu^{\to\wedge\vee\bot}$.

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

$$\alpha, \beta \ ::= \bot \mid \alpha \to \beta \mid \alpha \wedge \beta \mid \alpha \vee \beta.$$

(2) We have two kinds of variables, one is the λ-variables, denoted by $x, y, \ldots$, and the other is the μ-variables, denoted by $a, b, \ldots$. The 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 $\overline{\epsilon}$ for a finite sequence of eliminators, and we use $M\overline{\epsilon}$ to denote $(\ldots(M\epsilon_1)\epsilon_2\ldots)\epsilon_n$ for $\overline{\epsilon} \equiv \epsilon_1, \ldots \epsilon_n$. If $\overline{\epsilon}$ is empty, $M\overline{\epsilon}$ denotes M.

(3) The form of typing judgments is $\Gamma \vdash M : \alpha; \Delta$, where Γ is a set of type assignments for λ-variables in the form of $x : \alpha$ and Δ is a set of type assignments for μ-variables in the form of $a : \alpha$. We write $\Gamma, x : \alpha$ for the set $\Gamma \cup \{x : \alpha\}$. Typing rules are in Fig. 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 Fig. 2, where ϵ is an arbitrary eliminator. One-step reduction $\to$ is defined as their congruence closure. Strict reduction relation $\to^+$ is the transitive closure of the relation $\to$, and reduction relation

$$\frac{}{\Gamma, x : \alpha \vdash x : \alpha; \Delta}\ (\mathrm{Ax})$$

$$\frac{\Gamma \vdash M : \alpha; \Delta, a : \alpha}{\Gamma \vdash aM : \bot; \Delta, a : \alpha}\ (\bot\mathrm{I}) \qquad \frac{\Gamma \vdash M : \bot; \Delta, a : \alpha}{\Gamma \vdash \mu a.M : \alpha; \Delta}\ (\bot\mathrm{E})$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta; \Delta}{\Gamma \vdash \lambda x.M : \alpha \to \beta; \Delta}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash M : \alpha \to \beta; \Delta \quad \Gamma \vdash N : \alpha; \Delta}{\Gamma \vdash MN : \beta; \Delta}\ (\to\mathrm{E})$$

$$\frac{\Gamma \vdash M : \alpha; \Delta \quad \Gamma \vdash N : \beta; \Delta}{\Gamma \vdash \langle M, N\rangle : \alpha\wedge\beta; \Delta}\ (\wedge\mathrm{I}) \qquad \frac{\Gamma \vdash M : \alpha_1\wedge\alpha_2; \Delta}{\Gamma \vdash M\pi_j : \alpha_j; \Delta}\ (\wedge\mathrm{E})$$

$$\frac{\Gamma \vdash M : \alpha_j; \Delta}{\Gamma \vdash \iota_j M : \alpha_1 \vee \alpha_2; \Delta}\ (\vee\mathrm{I})$$

$$\frac{\Gamma \vdash M : \alpha \vee \beta; \Delta \quad \Gamma, x : \alpha \vdash N : \gamma; \Delta \quad \Gamma, y : \beta \vdash L : \gamma; \Delta}{\Gamma \vdash M[x.N, y.L] : \gamma; \Delta}\ (\vee\mathrm{E})$$

Fig. 1. Typing rules of $\lambda\mu^{\to\wedge\vee\bot}$.

$$\begin{array}{llcl}
(\beta_{\to}) & (\lambda x.M)\ N & \to_\beta & M[x:=N] \\
(\beta_{\wedge}) & \langle M_1, M_2\rangle\pi_j & \to_\beta & M_j \\
(\beta_{\vee}) & (\iota_j M)[x_1.N_1, x_2.N_2] & \to_\beta & N_j[x_j:=M] \\
 & & & \\
(\delta) & M[x.N, y.L]\epsilon & \to_\delta & M[x.N\epsilon, y.L\epsilon] \\
 & & & \\
(\mu) & (\mu a.M)\epsilon & \to_\mu & \mu a.M[a \Leftarrow \epsilon]
\end{array}$$

Fig. 2. Reduction rules of $\lambda\mu^{\to\wedge\vee\bot}$.

$\to^*$ is the reflexive transitive closure of the relation $\to$. We will use the symbol $\to_\beta$ to denote the one-step reduction relation defined by β-rules only. Relations $\to_\mu$ and $\to_\delta$ are similarly defined. A term M is said to be normal if there is no term N such that $M \to N$. We call the reduction defined by δ-rule permutative conversion.

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

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

Theorem 3 (*Subformula Property [5]*). *Any formula β occurring in a normal proof of $\Gamma \vdash M : \alpha; \Delta$ in $\lambda\mu^{\to\wedge\vee\bot}$ 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 have been many attempts to prove strong normalization of calculi with control operators by reducing them to that of the simply typed λ-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 λ-calculus such that the CPS-translation preserves reduction and strictly preserves β-reduction steps. Then we can reduce the strong normalization of the system to that of the simply typed λ-calculus. In the $\lambda\mu$-calculus, the CPS-translation defined by $\overline{\overline{M}} \equiv \lambda k.(M : k)$ preserves the reduction, where $M : K$ is defined by:

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

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 μ-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, $\overline{\overline{(\mu a.x)N}}$ 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 the 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 has been 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, this 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 $M \to_\beta N$ holds in $\lambda\mu$, then we have $\overline{\overline{M}} \to^+ \overline{\overline{N}}$, (2) if $M \to_\mu N$ holds, then we have $\overline{\overline{M}} \equiv \overline{\overline{N}}$, and (3) the μ-reduction is strongly normalizing. The strong normalization of $\lambda\mu$-calculus would follow from these lemmas and the strong normalization of the simply typed λ-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 $\lambda\mu^{\to\wedge\vee\perp}$. Matthes [11], however, pointed out that the proof contained an error. [5] defined the CPS-translation of μ-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 $M \to_\beta N$ holds, and on the other hand, for their CPS-translations $\overline{\overline{M}} \equiv \lambda k.(\lambda m.m(\lambda k_0.yk)(\lambda x.x))(\lambda z k_1.xk_1)$ 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 the strong normalization proof of $\lambda\mu^{\to\wedge\vee\perp}$ 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 the 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 μ-abstractions $\mu a.M$ where M does not contain any free variable a. We call such a μ-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 β-expansions of the original terms, have the same type, and contain no vacuous μ-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 λ-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 the 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 μ-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 λ-variable k_a for each μ-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 λ-term K as follows:

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

where the operation $\epsilon \mathbin{@} K$ is defined as

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

Definition 5 (*Double Negation Translation*). *Double negation translation* $\overline{\overline{\alpha}}$ of a formula α is defined as $\neg\neg\alpha^*$, where $\neg\alpha$ is an abbreviation of $\alpha \rightarrow \perp$, and α^* 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 Γ of type assignments for λ-variables, $\overline{\overline{\Gamma}}$ denotes the set $\{x : \overline{\overline{\alpha}} \mid x : \alpha \in \Gamma\}$. For a set Δ of type assignments for μ-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 λ-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 λ-calculus.*

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

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

Lemma 8. *For any terms P, K, K', M and any eliminator ϵ, we have*
(1) $(P : K)[k := K'] \equiv P : K[k := K'] \quad (k \notin FV(P))$,
(2) $(P : K)[x := \overline{\overline{M}}] \to^* P[x := M] : K[x := \overline{\overline{M}}]$,
(3) $(P : K)[k_a := \epsilon \mathbin{@} k_a] \equiv P[a \Leftarrow \epsilon] : K[k_a := \epsilon \mathbin{@} k_a]$.

Proof. By induction on P. $\square$

Lemma 9. *Suppose that a $\lambda\mu^{\to\wedge\vee\bot}$-term M is typable and not vacuous, and K is a λ-term. Then we have the following.*
(1) *If M contains a free μ-variable a and the type of a is not $\bot$, then the free λ-variable k_a appears in $M : K$.*
(2) *If the type of M is not $\bot$, 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 λ- and μ-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 $\beta \to \alpha \not\equiv \bot$, 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 $M \equiv aM_1$. Since the type of a is not $\bot$, the type of M_1 is not $\bot$ either. By the induction hypothesis for (2), we have $k_a \in FV(M_1 : k_a) = FV(aM_1 : K)$. 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 $\bot$, that of P_j is not $\bot$ 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 $\bot$, the type of a is not $\bot$ 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 $\lambda\mu^{\to\wedge\vee\bot}$-terms M and N, if $M \to N$ holds, we have $\overline{\overline{M}} \to^* \overline{\overline{N}}$. In particular, if M is typable and not vacuous, then we have the following.*
(1) *$M \to_\beta N$ implies $\overline{\overline{M}} \to^+ \overline{\overline{N}}$,*
(2) *$M \to_{\delta\mu} N$ implies $\overline{\overline{M}} \equiv \overline{\overline{N}}$.*

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

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

Other cases of β-rules are similarly proved.

The case of (δ). 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 $M : K \equiv Q : ([x_1.P_1, x_2.P_2] \mathbin{@} (\epsilon \mathbin{@} K)) \equiv Q : (\lambda m.mK_1K_2)$, where $K_j \equiv \lambda x_j.(P_j : (\epsilon \mathbin{@} K))$, which is identical to $\lambda x_j.(P_j\epsilon : K)$. Hence, we have $M : K \equiv Q : ([x_1.P_1\epsilon, x_2.P_2\epsilon] \mathbin{@} K) \equiv Q[x_1.P_1\epsilon, x_2.P_2\epsilon] : K$, which is identical to $N : K$.

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

Other cases are proved by the induction hypothesis and Lemma 9. For example, in the case of $M_1 M_2 \to_\beta M_1 N_2$ where $M_2 \to_\beta N_2$ and $M_1 M_2$ is typable and not vacuous, we have $\overline{\overline{M_2}} \to^+ \overline{\overline{N_2}}$ by the induction hypothesis. Furthermore, the type of M_1 is a functional type, which is not $\bot$, so we have $k \in FV(M_1 : k)$ by Lemma 9(2).

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

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

Then we can prove this case by $M_1M_2 : K \equiv M_1 : \lambda m.m\overline{\overline{M_2}}K \equiv (M_1 : k)[k := \lambda m.m\overline{\overline{M_2}}K] \to^+ (M_1 : k)[k := \lambda m.m\overline{\overline{N_2}}K] \equiv M_1N_2 : K$. □

4.2. Augmentations

For a $\lambda\mu^{\to\wedge\vee\bot}$-term M, an *augmentation* of M will be defined as a non-vacuous $\lambda\mu^{\to\wedge\vee\bot}$-term such that it is a β-expansion of M and any reduction step in $\lambda\mu^{\to\wedge\vee\bot}$ 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 λ-variable c_α for each type α. For each $\lambda\mu^{\to\wedge\vee\bot}$-term M, we define the set $\mathrm{Aug}(M)$ of *augmentations* in Fig. 3. For each eliminator ϵ, the set $\mathrm{Aug}(\epsilon)$ of its augmentations is defined by: $\mathrm{Aug}(\pi_j) = \{\pi_j\}$ and $\mathrm{Aug}([x.N, y.L]) = \{[x.N^+, y.L^+] | N^+ \in \mathrm{Aug}(N), L^+ \in \mathrm{Aug}(L)\}$.

The augmentation of a term M is defined so that (1) it is a β-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 $\mu a.M$ β-reduces to $\mu a.M$. We need to have some flexibility in the part $c_\alpha\bar{\epsilon}$ in $\mathrm{Aug}(\mu a.M)$ 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 $(\mu a.x)w \to \mu a.x$ and choose the augmentation $(\mu a.(\lambda z.x)(a(cy)))w$ of $(\mu a.x)w$, we should 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 augmentations instead of a single augmentation. c_α is used to adjust a type according to the variable a and the sequence $\bar{\epsilon}$ of eliminators so that some augmentations are typable. For example, when we think $\mu a.x$ of type α under $x : \bot, a : \alpha$, we have the typable augmentation $\mu a.(\lambda z.x)(ac_\alpha)$ of type α 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* $\mathrm{Aug}(M)$*, then N is not vacuous.*

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

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 exists a type context Π which consists of type assignments of the form $c_\beta : \beta$ and an augmentation $M^+ \in \mathrm{Aug}(M)$ such that $\Gamma, \Pi \vdash M^+ : \alpha; \Delta$ is derivable.*

(2) *If $M \to_\bullet N$ ($\bullet = \beta$, δ or μ) and $M^+ \in \mathrm{Aug}(M)$ hold, then there exists $N^+ \in \mathrm{Aug}(N)$ such that $M^+ \to_\bullet N^+$ holds.*

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

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

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

Other cases of β-rules are similarly proved.

The case of (δ). 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 $M^+ \in \mathrm{Aug}(M)$ is of the form $M^+ \equiv Q^+[x_1.P_1^+, x_2.P_2^+]\epsilon^+$. Then we have $M^+ \to_\delta Q^+[x_1.P_1^+\epsilon^+, x_2.P_2^+\epsilon^+]$, where the right-hand side is an augmentation of N.

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

Other cases are similarly proved. □

4.3. Strong normalization

We prove the strong normalization of $\lambda\mu^{\to\wedge\vee\bot}$ 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 $\lambda\mu^{\to\wedge\vee\bot}$-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 $M \to_{\delta\mu} N$ implies $|M| > |N|$. □

Theorem 15 (*Strong Normalization*). *Any typable $\lambda\mu^{\to\wedge\vee\bot}$-term is strongly normalizing.*

Proof. Assume that there is an infinite $\beta\delta\mu$-reduction sequence $M_0 \to M_1 \to M_2 \to \cdots$, where M_0 is a typable $\lambda\mu^{\to\wedge\vee\bot}$-term, and we will show contradiction. First, by Proposition 13, we have the infinite reduction sequence $M_0^+ \to M_1^+ \to M_2^+ \to \cdots$ 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 $\overline{\overline{M_0^+}} \to^* \overline{\overline{M_1^+}} \to^* \overline{\overline{M_2^+}} \to^* \cdots$ by Proposition 10. If the reduction step $M_i \to M_{i+1}$ is a β-step, we have $\overline{\overline{M_i^+}} \to^+ \overline{\overline{M_{i+1}^+}}$ 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 $M_i \to M_{i+1}$ is a β-step. Therefore, $\overline{\overline{M_0^+}} \to^* \overline{\overline{M_1^+}} \to^* \overline{\overline{M_2^+}} \to^* \cdots$ is an infinite β-reduction sequence, which contradicts the strong normalization of the simply typed λ-calculus. □

5. General elimination rules

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

5.1. The system $\lambda\mu_g^{\to\wedge\vee\bot}$

The disjunction elimination rule of $\lambda\mu^{\to\wedge\vee\bot}$ 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

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

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

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

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

$$[N, x.P] \star \epsilon' \equiv [N, x.P\epsilon']$$
$$[(x_1, x_2).P] \star \epsilon' \equiv [(x_1, x_2).P\epsilon']$$
$$[x_1.P_1, x_2.P_2] \star \epsilon' \equiv [x_1.P_1\epsilon', x_2.P_2\epsilon']$$

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

calculus [14]. Joachimski and Matthes [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 Fig. 4. They are a generalization of the ordinary elimination rules. The elimination rules in $\lambda\mu^{\to\wedge\vee\perp}$ are derivable in $\lambda\mu_g^{\to\wedge\vee\perp}$ by the general elimination rules as follows:

$$\frac{\begin{array}{c}\vdots\\ \Gamma \vdash A \to B; \Delta\end{array} \quad \begin{array}{c}\vdots\\ \Gamma \vdash A; \Delta\end{array} \quad \overline{\Gamma, B \vdash B; \Delta}}{\Gamma \vdash B; \Delta} (\to\mathrm{E}),$$

$$\frac{\begin{array}{c}\vdots\\ \Gamma \vdash A_1 \wedge A_2; \Delta\end{array} \quad \overline{\Gamma, A_1, A_2 \vdash A_j; \Delta}}{\Gamma \vdash A_j; \Delta} (\wedge\mathrm{E}).$$

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

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

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

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

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 $\lambda\mu^{\to\wedge\vee\perp}$ except elimination rules for implication and conjunction, which are given in Fig. 4.

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

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

5.2. CPS-translation

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

Definition 17 (*CPS-Translation*). Suppose that we have some fixed λ-variable k_a for each μ-variable a. The *CPS-translation* of a $\lambda\mu_g^{\to\wedge\vee\bot}$-term M is defined as a λ-term $\overline{\overline{M}} \equiv \lambda k.(M : k)$, where $M : K$ is defined by:

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

where the composition $\epsilon \mathbin{@} K$ is defined by

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

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

Proposition 18. *For any typable $\lambda\mu_g^{\to\wedge\vee\bot}$-term M, the CPS-translation $\overline{\overline{M}}$ is typable in the simply typed λ-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 $\Gamma \vdash M : \alpha; \Delta$. □

Intuitively, we can consider an eliminator ϵ as a part of a continuation, so $\epsilon \mathbin{@} K$ can be seen as a composition of ϵ 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 ϵ and ϵ', and any λ-term K, we have $(\epsilon \star \epsilon') \mathbin{@} K \equiv \epsilon \mathbin{@}(\epsilon' \mathbin{@} K)$.*

Proof. By case analysis with respect to the form of ϵ. □

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

Lemma 20. *For any terms P, K, K', M and any eliminator ϵ, we have*
(1) $(P : K)[k := K'] \to^* P : K[k := K']$,
(2) $(P : K)[x := \overline{\overline{M}}] \to^* P[x := M] : K[x := \overline{\overline{M}}]$,
(3) $(P : K)[k_a := \epsilon \mathbin{@} k_a] \equiv P[a \Leftarrow \epsilon] : K[k_a := \epsilon \mathbin{@} k_a]$.

Proof. By induction on P. □

Lemma 21. *Let a $\lambda\mu_g^{\to\wedge\vee\bot}$-term M be typable and not vacuous. For any λ-term K, we have the following.*
(1) *If M contains a free μ-variable a and the type of a is not $\bot$, then the free λ-variable k_a appears in $M : K$.*
(2) *If the type of M is not $\bot$, then all free variables of K appear in $M : K$.*

Proof. This lemma is proved by induction on M similarly to Lemma 9. □

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

(1) $M \to_\beta N$ *implies* $\overline{\overline{M}} \to^+ \overline{\overline{N}}$,

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

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

5.3. Augmentations

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

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

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

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

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 exists a context* Π*, 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* $M \to_\bullet N$ *(*$\bullet = \beta, \delta$ *or* μ*) and* $M^+ \in \mathrm{Aug}(M)$*, there exists* $N^+ \in \mathrm{Aug}(N)$ *such that* $M^+ \to_\bullet N^+$ *holds.*

Proof. They are proved similarly to Proposition 13. □

5.4. Strong normalization

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

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

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

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

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

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

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

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

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

$$\begin{array}{rlrll} \lfloor x \rfloor_a & = 0, & \lfloor \lambda x.M \rfloor_a & = \lfloor M \rfloor_a, & \\ \lfloor \langle M, N \rangle \rfloor_a & = \lfloor M \rfloor_a + \lfloor N \rfloor_a, & \lfloor \iota_j M \rfloor_a & = \lfloor M \rfloor_a, & \\ \lfloor \mu b.M \rfloor_a & = \lfloor M \rfloor_a, & \lfloor bM \rfloor_a & = \lfloor M \rfloor_a & (a \not\equiv b), \\ \lfloor M\epsilon \rfloor_a & = \lfloor M \rfloor_a + \#M \lfloor \epsilon \rfloor_a, & \lfloor aM \rfloor_a & = \lfloor M \rfloor_a + \#M, & \end{array}$$

$$\lfloor [N, x.P] \rfloor_a = \lfloor N \rfloor_a + \lfloor P \rfloor_a, \qquad \lfloor [(x, y).P] \rfloor_a = \lfloor P \rfloor_a,$$
$$\lfloor [x_1.P_1, x_2.P_2] \rfloor_a = \lfloor P_1 \rfloor_a + \lfloor P_2 \rfloor_a.$$

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. □

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* $a \not\equiv b$,
(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 $a \not\equiv b$ 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. $|(aN)[a \Leftarrow \epsilon]| = |a(N[a \Leftarrow \epsilon]\epsilon)| = |N[a \Leftarrow \epsilon]\epsilon| = |N[a \Leftarrow \epsilon]| + \#N[a \Leftarrow \epsilon]|\epsilon| = |N| + \lfloor N \rfloor_a |\epsilon| + \#N|\epsilon|$ (by (1) and the induction hypothesis) $= |aN| + \lfloor aN \rfloor_a |\epsilon|$. Other cases are similarly proved. □

Proposition 29. *If $M \to_{\delta\mu} N$, 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 ϵ_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 $M \to_{\delta\mu} N$. Note that we have to prove $\#M \geq \#N$ and $\lfloor M \rfloor_a \geq \lfloor N \rfloor_a$ simultaneously.

The case of (δ). 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 (μ). 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. □

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

Proof. By Proposition 29. □

Finally we have the result.

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

Proof. It is proved similarly to Theorem 15 by Propositions 22 and 25, Theorem 30, and the subject reduction property. □

6. Embedding from $\lambda\mu^{\to\wedge\vee\bot}$ to $\lambda\mu_g^{\to\wedge\vee\bot}$

In this section, we will give an embedding from $\lambda\mu^{\to\wedge\vee\bot}$ to $\lambda\mu_g^{\to\wedge\vee\bot}$ 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 the strong normalization of $\lambda\mu^{\to\wedge\vee\bot}$ to that of $\lambda\mu_g^{\to\wedge\vee\bot}$.

Definition 32. We define a map M^* from $\lambda\mu^{\to\wedge\vee\bot}$ to $\lambda\mu_g^{\to\wedge\vee\bot}$ 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 $\lambda\mu^{\to\wedge\vee\bot}$ to eliminators in $\lambda\mu_g^{\to\wedge\vee\bot}$ 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 ϵ, 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 $((aM)[a \Leftarrow \epsilon])^* \equiv a((M[a \Leftarrow \epsilon])^*\epsilon^\dagger) \equiv a((M^*[a \Leftarrow \epsilon^\dagger])\epsilon^\dagger) \equiv (aM^*)[a \Leftarrow \epsilon^\dagger] \equiv (aM)^*[a \Leftarrow \epsilon^\dagger]$. □

Proposition 34. (1) *If $\Gamma \vdash M : \alpha; \Delta$ is derivable in $\lambda\mu^{\to\wedge\vee\bot}$, $\Gamma \vdash M^* : \alpha; \Delta$ is derivable in $\lambda\mu_g^{\to\wedge\vee\bot}$.*
(2) *If $M \to_\bullet N$ ($\bullet = \beta$, δ or μ) in $\lambda\mu^{\to\wedge\vee\bot}$, we have $M^* \to_\bullet N^*$ in $\lambda\mu_g^{\to\wedge\vee\bot}$.*

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

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

$M^*[x_1.P_1^*\epsilon^\dagger, x_2.P_2^*\epsilon^\dagger] \equiv M^*[x_1.(P_1\epsilon)^*, x_2.(P_2\epsilon)^*] \equiv (M[x_1.P_1\epsilon, x_2.P_2\epsilon])^*$. The case of μ-reduction is proved from Lemma 33(2) by $((\mu a.M)\epsilon)^* \equiv (\mu a.M^*)\epsilon^\dagger \to_\mu \mu a.M^*[a \Leftarrow \epsilon^\dagger] \equiv (\mu a.M[a \Leftarrow \epsilon])^*$. The induction steps are easily proved from the induction hypothesis. □

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

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 λ-calculus. This is based on our same idea 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 $\lambda\mu^{\to\wedge\vee\perp}$ 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 the strong normalization of classical natural deduction with permutative conversion. We hope our proof will contribute to a 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 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 $\overline{\lambda}\mu\tilde{\mu}$ 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 β-rule for implication $(\lambda x.M)[N, y.P] \to P[y := M[x := N]]$ 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 the 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 $\lambda\mu_g^{\to\wedge\vee\perp}$, 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 $\lambda\mu_g^{\to\wedge\vee\perp}$ 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 $M \to_\delta N[x_1.P_1, x_2, P_2][y_1.Q_1\epsilon, y_2, Q_2\epsilon]$ and $M \to_\delta N[x_1.P_1[y_1.Q_1, y_2, Q_2], x_2, P_2[y_1.Q_1, y_2, Q_2]]\epsilon$, 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 β-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 $\lambda\mu^{\to\wedge\vee\perp}$-term M. This map also preserves typability and the $\beta\delta$-reduction, but may fail to preserve the μ-reduction. If we restrict the domain of the map $(\cdot)_*$ to the image of $(\cdot)^*$, any reduction $\to_\bullet$ is preserved. That is, if we have $M^* \to_\bullet N^*$, then $M \to_\bullet N$. More detailed investigation of the relationship between general and ordinary elimination rules will be another future work.

Acknowledgments

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 (2003) 225–237.

[2] P.-L. Curien, H. Herbelin, The duality of computation, in: Proc. 5th International Conference on Functional Programming, ICFP '00, 2000, pp. 233–243.

[3] R. David, K. Nour, A short proof of the strong normalization of classical natural deduction with disjunction, Journal of Symbolic Logic 68 (4) (2003) 1277–1288.

[4] P. de Groote, A simple calculus of exception handling, in: M. Dezani-Ciancaglini, G. Plotkin (Eds.), Typed Lambda Calculi and Applications, TLCA '95, in: LNCS, vol. 902, 1995, pp. 201–215.

[5] P. de Groote, Strong normalization of classical natural deduction with disjunction, in: S. Abramsky (Ed.), Typed Lambda Calculi and Applications, TLCA '01, in: LNCS, vol. 2044, 2001, pp. 182–196.

[6] P. de Groote, On the strong normalisation of intuitionistic natural deduction with permutation-conversions, Information and Computation 178 (2002) 441–464.

[7] K. Fujita, Domain-free $\lambda\mu$-calculus, Theoretical Informatics and Applications 34 (6) (2000) 433–466.

[8] T. Griffin, A formulae-as-types notion of control, in: 17th Ann. ACM Symp. on Principles of Programming Languages, POPL '90, 1990, pp. 47–58.

[9] F. Joachimski, R. Matthes, Standardization and confluence for a lambda calculus with generalized applications, in: L. Bachmair (Ed.), Rewriting Techniques and Applications, RTA '00, in: LNCS, vol. 1833, 2000, pp. 141–155.

[10] F. Joachimski, R. Matthes, Short proofs of normalization for the simply-typed λ-calculus, permutative conversions and Gödels's T, Archive for Mathematical Logic 42 (2003) 59–87.

[11] R. Matthes, Stabilization — an alternative to double-negation translation for classical natural deduction, in: V. Stoltenberg-Hansen, J. Väänänen (Eds.), Proc. Logic Colloquium '03, in: Lecture Notes in Logic, vol. 24, 2003, pp. 167–199.

[12] R. Matthes, Non-strictly positive fixed-points for classical natural deduction, Annals of Pure and Applied Logic 133 (2005) 205–230.

[13] K. Nakazawa, M. Tatsuta, Strong normalization proof with CPS-translation for second order classical natural deduction, Journal of Symbolic Logic 68 (3) (2003) 851–859;Corrigendum is available in Journal of Symbolic Logic 68 (4) (2003) 1415–1416.

[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, in: LNCS, vol. 4583, 2007, pp. 336–350.

[15] K. Nour, K. Saber, A Semantics of Realisability for the Classical Propositional Natural Deduction, in: Electronic Notes in Theoretical Computer Science, vol. 140, 2005, pp. 31–39.

[16] A. Ohori, The Logical Abstract Machine: A Curry–Howard isomorphism for machine code, in: A. Middeldorp, T. Sato (Eds.), Proc. Fuji International Symposium on Functional and Logic Programming, FLOPS '99, in: LNCS, vol. 1722, 1999, pp. 300–318.

[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, in: LNCS, vol. 624, 1992, pp. 190–201.

[18] M. Parigot, Strong normalization for second order classical natural deduction, Journal of Symbolic Logic 62 (4) (1997) 1461–1479.

[19] M. Takahashi, Parallel reduction in λ-calculus, Information and Computation 118 (1) (1995) 120–127.
[20] M. Tatsuta, G. Mints, A simple proof of second-order strong normalization with permutative conversions, Annals of Pure and Applied Logic 136 (2005) 134–155.
[21] J. von Plato, Natural deduction with general elimination rules, Annals of Mathematical Logic 40 (7) (2001) 541–567.
[22] P. Wadler, Call-by-value is dual to call-by-name, in: C. Runciman, O. Shivers (Eds.), in: Proc. International Conference on Functional Programming, ICFP '03, 2003, pp. 189–201.
[23] Y. Yamagata, Strong normalization of second order symmetric lambda-mu calculus, Information and Computation 193 (1) (2004) 1–20.