



## -calculus II

著者(英)	Yuichi KOMORI
journal or publication title	Tsukuba Journal of Mathematics
volume	37
number	2
page range	307-320
year	2013-12
URL	<a href="http://hdl.handle.net/2241/00154311">http://hdl.handle.net/2241/00154311</a>

## $\lambda\rho$ -CALCULUS II

By

Yuichi KOMORI

**Abstract.** In [4], the author introduced the system  $\lambda\rho$ -calculus and stated without proof that the strong normalization theorem holds. Here we introduce a lemma (Lemma 4.10) and use it to prove the strong normalization theorem. While a typed  $\lambda$ -term itself is a derivation of the natural deduction for intuitionistic implicational logic (cf. [2]), a typed  $\lambda\rho$ -term itself is a derivation of the natural deduction for classical implicational logic. Our system is simpler than the implicational fragment of Parigot’s  $\lambda\mu$ -calculus (cf. [5]).

### 1 The Type Free $\lambda\rho$ -Calculus

DEFINITION 1.1 ( $\lambda\rho$ -terms). Assume to have an infinite sequence of  $\lambda$ -variables and an infinite sequence of  $\rho$ -variables. Then the linguistic expressions called  $\lambda\rho$ -terms are defined as:

1. each  $\lambda$ -variable is a  $\lambda\rho$ -term, called an *atom* or *atomic term*,
2. if  $M$  and  $N$  are  $\lambda\rho$ -terms then  $(MN)$  is a  $\lambda\rho$ -term called an *application*,
3. if  $M$  is a  $\lambda\rho$ -term and  $a$  is a  $\rho$ -variable then  $(aM)$  is a  $\lambda\rho$ -term called *absurd*,
4. if  $M$  is a  $\lambda\rho$ -term and  $f$  is a  $\lambda$ -variable or a  $\rho$ -variable then  $(\lambda f.M)$  is a  $\lambda\rho$ -term called an *abstract*. (If  $f$  is a  $\lambda$ -variable or a  $\rho$ -variable, then  $(\lambda f.M)$  is a  $\lambda$ -abstract or a  $\rho$ -abstract, respectively.)

Note that  $\rho$ -variables are not terms.  $\lambda$ -variables are denoted by “ $u$ ”, “ $v$ ”, “ $w$ ”, “ $x$ ”, “ $y$ ”, “ $z$ ”.  $\rho$ -variables are denoted by “ $a$ ”, “ $b$ ”, “ $c$ ”, “ $d$ ”. A *term*-

---

2000 AMS subject classification: 03B05, 03B40.

Key words and phrases:  $\lambda$ -calculus, typed  $\lambda$ -calculus, normalization theorem, classical logic,  $\lambda\rho$ -calculus,  $\lambda\mu$ -calculus, LK.

Received December 18, 2012.

Revised March 12, 2013.

variable is a  $\lambda$ -variable or a  $\rho$ -variable. Term-variables are denoted by “ $f$ ”, “ $g$ ”, “ $h$ ”. Distinct letters denote distinct variables unless stated otherwise.

A term  $\lambda a.M$  is sometimes denoted by  $\rho a.M$  if the variable  $a$  is a  $\rho$ -variable.

Arbitrary  $\lambda\rho$ -terms are denoted by “ $L$ ”, “ $M$ ”, “ $N$ ”, “ $P$ ”, “ $Q$ ”, “ $R$ ”, “ $S$ ”, “ $T$ ”.

DEFINITION 1.2 (Free variables). The set  $FV(M)$  of all term variables free in  $M$ , is defined as:

1.  $FV(x) = \{x\}$ ,
2.  $FV((MN)) = FV(M) \cup FV(N)$ ,
3.  $FV((aM)) = FV(M) \cup \{a\}$ ,
4.  $FV((\lambda f.M)) = FV(M) - \{f\}$ .

DEFINITION 1.3 ( $\rho\beta$ -contraction). A  $\rho\beta$ -redex is any  $\lambda\rho$ -term of form  $(aM)N$ ,  $(\lambda x.M)N$  or  $(\lambda a.M)N$ ; its contractum is  $(aM)$ ,  $[N/x]M$  or  $\lambda b.([\lambda x.b(xN)/a]M)N$  respectively. The re-write rules are

$$(aM)N \triangleright_{1\alpha} (aM),$$

$$(\lambda x.M)N \triangleright_{1\beta} [N/x]M,$$

$$(\lambda a.M)N \triangleright_{1\rho} \lambda b.([\lambda x.b(xN)/a]M)N, \text{ where } b \text{ is the first } \rho\text{-variable} \\ \text{and } x \text{ is the first } \lambda\text{-variable such that } b \text{ and } x \text{ do} \\ \text{not occur in } aMN,$$

$$M \triangleright_{1\rho\beta} N \text{ if } M \triangleright_{1\alpha} N, M \triangleright_{1\beta} N \text{ or } M \triangleright_{1\rho} N.$$

We call a  $\lambda\rho$ -term of form  $(aM)N$  an  $a$ -redex,  $(\lambda x.M)N$  a  $\beta$ -redex and  $(\lambda a.M)N$  a  $\rho$ -redex. If  $P$  contains a  $\rho\beta$ -redex-occurrence  $\underline{R}$  and  $Q$  is the result of replacing this by its contractum, we say that  $P$   $\rho\beta$ -contracts to  $Q$  ( $P \triangleright_{1\rho\beta} Q$ ), and we call the triple  $\langle P, \underline{R}, Q \rangle$  a  $\rho\beta$ -contraction of  $P$ .

DEFINITION 1.4 ( $\rho\beta$ -reduction). A  $\rho\beta$ -reduction of a term  $P$  is a finite (perhaps empty) or infinite sequence of  $\rho\beta$ -contractions with form

$$\langle P_1, \underline{R}_1, Q_1 \rangle, \langle P_2, \underline{R}_2, Q_2 \rangle, \dots$$

where  $P_1 \equiv_\alpha P$  and  $Q_i \equiv_\alpha P_{i+1}$  for  $i = 1, 2, \dots$ . We say a finite  $\rho\beta$ -reduction is from  $P$  to  $Q$  iff either it has  $n \geq 1$   $\rho\beta$ -contractions and  $Q_n \equiv_\alpha Q$  or it is empty and  $P \equiv_\alpha Q$ . A reduction from  $P$  to  $Q$  is said to terminate or end to  $Q$ . If there

is a reduction from  $P$  to  $Q$  we say that  $P$   $\rho\beta$ -reduces to  $Q$ , in symbols

$$P \triangleright_{\rho\beta} Q.$$

Note that  $\alpha$ -conversions are allowed in a  $\rho\beta$ -reduction.

**THEOREM 1.5** (Church-Rosser theorem for  $\rho\beta$ -reduction). *If  $M \triangleright_{\rho\beta} P$  and  $M \triangleright_{\rho\beta} Q$ , then there exists  $T$  such that*

$$P \triangleright_{\rho\beta} T \quad \text{and} \quad Q \triangleright_{\rho\beta} T.$$

**PROOF.** Similar to the case of  $\beta$ -reduction, see [3]. □

## 2 Typed $\lambda\rho$ -Terms

**DEFINITION 2.1** (Types). An infinite sequence of *type-variables*, distinct from the term-variables, is assumed to be given. *Types* are linguistic expressions defined as:

1. each type-variable is a type called an *atom*;
2. if  $\sigma$  and  $\tau$  are types then  $(\sigma \rightarrow \tau)$  is a type called a *composite type*.

Type-variables are denoted by “ $p$ ”, “ $q$ ”, “ $r$ ” with or without number-subscripts, and distinct letters denote distinct variables unless otherwise stated.

Arbitrary types are denoted by lower-case Greek letters except “ $\lambda$ ” and “ $\rho$ ”.

Parentheses will often (but not always) be omitted from types, and the reader should restore omitted ones in the way of association to the right.

Any term-variable is assumed to have one type. For any type  $\tau$ , an infinite sequence of  $\lambda$ -variables with type  $\tau$  and an infinite sequence of  $\rho$ -variables with type  $\tau$  are assumed to exist.

**DEFINITION 2.2** (Typed  $\lambda\rho$ -terms). We shall define typed  $\lambda\rho$ -terms and  $Type(M)$  (assertion  $type(M) = \tau$  is denoted by  $M : \tau$ ) simultaneously.

1. A  $\lambda$ -variable  $x$  with type  $\tau$  is a typed  $\lambda\rho$ -term, called an *atom*, and  $x : \tau$ .
2. If  $M$  and  $N$  are typed  $\lambda\rho$ -terms and  $M : \sigma \rightarrow \tau$  and  $N : \sigma$ , then the expression  $(MN)$  is a typed  $\lambda\rho$ -term called an *application* and  $(MN) : \tau$ .
3. Let  $\sigma$  be any type. If  $M$  is a typed  $\lambda\rho$ -term and  $M : \tau$  and  $a$  is a  $\rho$ -variable with type  $\tau$ , then the expression  $(aM)^\sigma$  is a typed  $\lambda\rho$ -term called an *absurd* and  $(aM)^\sigma : \sigma$ .

4. If  $M$  is a typed  $\lambda\rho$ -term and  $M : \tau$  and  $x$  is a  $\lambda$ -variable with type  $\sigma$ , then the expression  $(\lambda x.M)$  is a typed  $\lambda\rho$ -term called a  $\lambda$ -abstract and  $(\lambda x.M) : \sigma \rightarrow \tau$ .
5. If  $M$  is a typed  $\lambda\rho$ -term and  $M : \tau$  and  $a$  is a  $\rho$ -variable with type  $\tau$ , then the expression  $(\lambda a.M)$  is a typed  $\lambda\rho$ -term called a  $\rho$ -abstract and  $(\lambda a.M) : \tau$ .

Typed  $\lambda\rho$ -terms will be abbreviated using the same conventions as for  $\lambda\rho$ -terms.

DEFINITION 2.3 (Free variables in a typed  $\lambda\rho$ -term). Let  $M$  be a typed  $\lambda\rho$ -term. The set  $FV(M)$  of all the free term-variables in  $M$ , is defined as:

1.  $FV(x) = \{x\}$ ,
2.  $FV((MN)) = FV(M) \cup FV(N)$ ,
3.  $FV((aM)^\sigma) = FV(M) \cup \{a\}$ ,
4.  $FV((\lambda f.M)) = FV(M) - \{f\}$ ,

$FV_\lambda(M)$  and  $FV_\rho(M)$  denote the set of all  $\lambda$ -variables in  $FV(M)$  and the set of all  $\rho$ -variables in  $FV(M)$ , respectively.

EXAMPLE 2.4 (Peirce's Law).

$$\lambda xa.x(\lambda y.(ay)^\beta), \quad \text{where } x : (\alpha \rightarrow \beta) \rightarrow \alpha, \ y : \alpha \text{ and } a : \alpha.$$

On the other hand, the proof of Peirce's Law is  $\lambda xa.[a](x(\lambda yb.[a]y))$  in Parigot's system. We think that proofs in our system are generally simpler than those in the implicational fragment of Parigot's system.

The above typed  $\lambda\rho$ -term is written in a tree form as follows:

$$\frac{\frac{\frac{x : (\alpha \rightarrow \beta) \rightarrow \alpha}{\frac{\alpha}{\lambda a}}}{((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow \alpha} \lambda x, \quad \frac{\frac{a : \alpha \quad y : \alpha}{\beta}}{\alpha \rightarrow \beta} \lambda y}{\alpha \rightarrow \beta} \lambda y$$

or in a more redundant form as follows:

$$\frac{\frac{\frac{x : (\alpha \rightarrow \beta) \rightarrow \alpha}{\lambda y.ay : \alpha \rightarrow \beta}}{x(\lambda y.ay) : \alpha}}{\lambda a.x(\lambda y.ay) : \alpha}}{\lambda xa.x(\lambda y.ay) : ((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow \alpha}$$

DEFINITION 2.5 (Type-erasure and typability). We assume the existence of two mappings  $j$  and  $k$  such that  $j$  is a one-to-one onto mapping from the set of all  $\lambda$ -variables with type to the set of all  $\lambda$ -variables and  $k$  is a one-to-one onto mapping from the set of all  $\rho$ -variables with type to the set of all  $\rho$ -variables. For simplicity, we write  $x$  and  $a$  for  $j(x)$  and  $k(a)$ , respectively. The *type-erasure*  $er(M)$  of a typed  $\lambda\rho$ -term  $M$  is the  $\lambda\rho$ -term obtained by erasing all types from  $M$ . Namely, type-erasure  $er(M)$  is defined as follows:

1.  $er(x) \equiv x$ ,
2.  $er((MN)) \equiv (er(M) er(N))$ ,
3.  $er((aM)^\sigma) \equiv (a er(M))$ ,
4.  $er((\lambda x.M)) \equiv (\lambda x.er(M))$ ,
5.  $er((\lambda a.M)) \equiv (\lambda a.er(M))$ .

A  $\lambda\rho$ -term  $M$  is called *typable* iff there exists a typed  $\lambda\rho$ -term  $N$  such that  $er(N) \equiv_\alpha M$ .

For typed  $\lambda\rho$ -terms  $M, N$  and a  $\lambda$ -variable  $x$  with type  $Type(x)$ , the substitution of  $N$  for  $x$  in  $M$   $[N/x]M$  is defined as usual. For a typed  $\lambda\rho$ -term  $M$  and  $\rho$ -variables  $a, b$  such that  $Type(a) = Type(b)$ , the substitution of  $b$  for  $a$  in  $M$   $[b/a]M$  is also defined as usual.

To define  $\rho\beta$ -contraction for typed  $\lambda\rho$ -terms, we have to define the substitution of an expression  $\lambda x.b(xN)$  for a  $\rho$ -variable. Notice that the expression  $\lambda x.b(xN)$  is not a typed  $\lambda\rho$ -term.

DEFINITION 2.6 (Substitution of an expression  $\lambda x.b(xN)$  for a  $\rho$ -variable). For typed  $\lambda\rho$ -terms  $M, N$ , a  $\rho$ -variable  $b$ , we define  $[\lambda x.b(xN)/a]M$  to be the result of substituting  $\lambda x.b(xN)$  for every free occurrence of  $a$  in  $M$ , where  $Type(x) = Type(a) = \alpha \rightarrow \beta$ ,  $b : \beta$  and  $N : \alpha$ .

1.  $[\lambda x.b(xN)/a]M \equiv M$  if  $a \notin FV(M)$ ,
2.  $[\lambda x.b(xN)/a](MR) \equiv ([\lambda x.b(xN)/a]M [\lambda x.b(xN)/a]R)$  if  $a \in FV(MR)$ ,
3.  $[\lambda x.b(xN)/a](\lambda y.M) \equiv \lambda y.[\lambda x.b(xN)/a]M$  if  $a \in FV(M)$  and  $y \notin FV(\lambda x.b(xN))$ ,
4.  $[\lambda x.b(xN)/a](\lambda y.M) \equiv \lambda z.[\lambda x.b(xN)/a][z/y]M$  if  $a \in FV(M)$  and  $y \in FV(\lambda x.b(xN))$ ,
5.  $[\lambda x.b(xN)/a](cM)^\sigma \equiv (c[\lambda x.b(xN)/a]M)^\sigma$  if  $a \in FV(M)$  and  $c \neq a$ ,
6.  $[\lambda x.b(xN)/a](aM)^\sigma \equiv (\lambda x.(b(xN))^\sigma)[\lambda x.b(xN)/a]M$ ,
7.  $[\lambda x.b(xN)/a](\lambda c.M) \equiv \lambda c.[\lambda x.b(xN)/a]M$  if  $a \in FV(\lambda c.M)$  and  $c \notin FV(bN)$ ,

8.  $[\lambda x.b(xN)/a](\lambda c.M) \equiv \lambda d.[\lambda x.b(xN)/a][d/c]M$  if  $a \in FV(\lambda c.M)$  and  $c \in FV(bN)$ .

(In 4  $z$  is the first  $\lambda$ -variable with type  $Type(y)$  which does not occur in  $xNM$ . In 8  $d$  is the first  $\rho$ -variable with type  $Type(c)$  which does not occur in  $bNM$ .)

DEFINITION 2.7 ( $\rho\beta$ -contraction for typed  $\lambda\rho$ -terms). A  $\rho\beta$ -redex is any typed  $\lambda\rho$ -term of form  $(aM)^{\sigma \rightarrow \tau}N$ ,  $(\lambda x.M)N$  or  $(\lambda a.M)N$ ; its contractum is  $(aM)^\tau$ ,  $[N/x]M$  or  $\lambda b.([\lambda x.b(xN)/a]M)N$  respectively. The re-write rules are

$$(aM)^{\sigma \rightarrow \tau}N \triangleright_{1a} (aM)^\tau,$$

$$(\lambda x.M)N \triangleright_{1\beta} [N/x]M,$$

$$(\lambda a.M)N \triangleright_{1\rho} \lambda b.([\lambda x.b(xN)/a]M)N, \text{ where } b \text{ is the first } \rho\text{-variable} \\ \text{and } x \text{ is the first } \lambda\text{-variable such that } b : Type(MN), \\ x : Type(a) \text{ and } b \text{ and } x \text{ do not occur in } aMN,$$

$$M \triangleright_{1\rho\beta} N \text{ if } M \triangleright_{1a} N, M \triangleright_{1\beta} N \text{ or } M \triangleright_{1\rho} N.$$

We call a  $\lambda\rho$ -term of form  $(aM)^{\sigma \rightarrow \tau}N$  an  $a$ -redex,  $(\lambda x.M)N$  a  $\beta$ -redex and  $(\lambda a.M)N$  a  $\rho$ -redex. If  $P$  contains a  $\rho\beta$ -redex-occurrence  $\underline{R}$  and  $Q$  is the result of replacing this by its contractum, we say that  $P$   $\rho\beta$ -contracts to  $Q$  ( $P \triangleright_{1\rho\beta} Q$ ), and we call the triple  $\langle P, \underline{R}, Q \rangle$  a  $\rho\beta$ -contraction of  $P$ .

A  $\rho\beta$ -reduction for typed  $\lambda\rho$ -terms is defined in the same way as a  $\rho\beta$ -reduction for type free  $\lambda\rho$ -terms.

THEOREM 2.8 (Church-Rosser theorem for typed  $\lambda\rho$ -terms). Let  $M$ ,  $P$  and  $Q$  be typed  $\lambda\rho$ -terms. If  $M \triangleright_{\rho\beta} P$  and  $M \triangleright_{\rho\beta} Q$ , then there exists a typed  $\lambda\rho$ -term  $T$  such that

$$P \triangleright_{\rho\beta} T \text{ and } Q \triangleright_{\rho\beta} T.$$

PROOF. Similar to the case of  $\beta$ -reduction, see [3]. □

### 3 Subject-Reduction Theorem for Typed $\lambda\rho$ -Calculus

LEMMA 3.1. If  $P$  and  $Q$  are typed  $\lambda\rho$ -terms and  $x$  is a  $\lambda$ -variable with type  $Type(Q)$ , then  $[Q/x]P$  is a typed  $\lambda\rho$ -term and  $Type([Q/x]P) = Type(P)$  and  $FV([Q/x]P) \subseteq (FV(P) - \{x\}) \cup FV(Q)$ .

PROOF. By induction on the length of  $P$ . □

LEMMA 3.2. *If  $P$  and  $Q$  are typed  $\lambda\rho$ -terms,  $Type(x) = Type(a) = \sigma \rightarrow \tau$ ,  $b : \tau$ ,  $Q : \sigma$  and  $x \notin FV(Q)$ , then  $[\lambda x.b(xQ)/a]P$  is a typed  $\lambda\rho$ -term and  $Type([\lambda x.b(xQ)/a]P) = Type(P)$  and  $FV([\lambda x.b(xQ)/a]P) \subseteq (FV(P) - \{a\}) \cup FV(Q) \cup \{b\}$ .*

PROOF. By induction on the length of  $P$ . The only nontrivial case is  $P \equiv (aP_1)^\gamma$ . Then  $P_1 : \sigma \rightarrow \tau$  and  $[\lambda x.b(xQ)/a](aP_1)^\gamma \equiv (\lambda x.(b(xQ))^\gamma) \cdot [\lambda x.b(xQ)/a]P_1$ . Now we have  $Type([\lambda x.b(xQ)/a]P) = Type(P) = \gamma$  and  $FV([\lambda x.b(xQ)/a]P) = FV([\lambda x.b(xQ)/a]P_1) \cup FV(Q) \cup \{b\} \subseteq (FV(P) - \{a\}) \cup FV(Q) \cup \{b\}$ .  $\square$

THEOREM 3.3 (Subject-reduction theorem). *If  $P \triangleright_{\rho\beta} Q$ , then  $Type(Q) = Type(P)$  and  $FV(Q) \subseteq FV(P)$ .*

PROOF. By Lemma 3.1, it is enough to take care of the case in which  $P$  is a redex and  $Q$  is its contractum. It is enough to prove that if  $P \triangleright_{1\rho\beta} Q$ , then  $Type(Q) = Type(P)$  and  $FV(Q) \subseteq FV(P)$ .

Case 1:  $P \equiv (aP_1)^{\sigma \rightarrow \tau} P_2$  and  $Q \equiv (aP_1)^\tau$ . It is obvious that  $Type(P) = Type(Q) = \tau$ . Then we have  $FV(Q) = FV(P_1) \cup \{a\} \subseteq FV(P_1) \cup \{a\} \cup FV(P_2) = FV(P)$ .

Case 2:  $P \equiv (\lambda x.P_1)P_2$  and  $Q \equiv [P_2/x]P_1$ . By Lemma 3.1, we have  $Type(Q) = Type(P)$  and  $FV(Q) \subseteq FV(P)$ .

Case 3:  $P \equiv (\lambda a.P_1)P_2$  and  $Q \equiv \lambda b.([\lambda x.b(xP_2)/a]P_1)P_2$ . By Lemma 3.2, we have  $Type(Q) = Type(P)$  and  $FV(Q) \subseteq FV(P)$ .  $\square$

#### 4 Strong Normalization Theorem for Typed $\lambda\rho$ -Terms

We prove the strong normalization theorem for typed  $\lambda\rho$ -terms, that is, for every typed  $\lambda\rho$ -term  $M$ , all reductions starting at  $M$  are finite. To prove the theorem, we introduce the concept of  $*$ -expansion and use the strong normalization theorem for typed  $\lambda$ -terms.

DEFINITION 4.1 ( $\circ$ -translation). For every typed  $\lambda\rho$ -term  $(\lambda a.M)$ , where  $M : \tau$ , we define  $\circ$ -translation as follows:

1. if  $\tau$  is an atomic type, then  $(\lambda a.M)^\circ \equiv (\lambda a.M)$ ,
2. if  $\tau \equiv \alpha \rightarrow \beta$ , then  $(\lambda a.M)^\circ \equiv (\lambda y.(\lambda b.([\lambda x.b(xy)/a]My)^\circ))$ , where  $x$ ,  $y$  and  $b$  are the first  $\lambda$ -variable with the type  $\alpha \rightarrow \beta$ , the second  $\lambda$ -variable with the type  $\alpha$  and the first  $\rho$ -variable with the type  $\beta$  which do not occur in  $aM$ .



By the above definition, if  $M : \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow p$ , then  $(\lambda a.M)^\circ \triangleright_\beta \lambda y_1 \cdots y_n b. [\lambda x. b(xy_1 \cdots y_n)/a] M y_1 \cdots y_n$  where  $x : \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow p$ ,  $y_1 : \sigma_1 \cdots y_n : \sigma_n$  and  $b : p$ .

Note that Parigot [6] proved the strong normalization of propositional typed  $\lambda\mu$ -calculus using Gödel translation. This translation is similar to  $\circ$ -translation.

LEMMA 4.2.  $Type((\lambda a.M)^\circ) = Type(\lambda a.M)$  and  $FV((\lambda a.M)^\circ) = FV(\lambda a.M)$ .

PROOF. By induction on the length of  $Type(\lambda a.M)$ . If  $Type(\lambda a.M)$  is an atom, then  $(\lambda a.M)^\circ \equiv \lambda a.M$ , so  $Type(\lambda a.M) = Type((\lambda a.M)^\circ)$  and  $FV(\lambda a.M) = FV((\lambda a.M)^\circ)$ . If  $\lambda a.M : \alpha \rightarrow \beta$ , then

$$(\lambda a.M)^\circ \equiv (\lambda y. (\lambda b. [\lambda x. b(xy)/a] M y)^\circ) \quad \text{where } x : \alpha \rightarrow \beta \text{ and } y : \alpha.$$

Since  $M : \alpha \rightarrow \beta$ ,  $[\lambda x. b(xy)/a] M y : \beta$  by Lemma 3.2 and  $\lambda b. [\lambda x. b(xy)/a] M y : \beta$ . Hence by the induction hypothesis,  $(\lambda b. [\lambda x. b(xy)/a] M y)^\circ : \beta$  and  $FV((\lambda b. [\lambda x. b(xy)/a] M y)^\circ) = FV(\lambda b. [\lambda x. b(xy)/a] M y) = (FV(M) - \{a\}) \cup \{y\}$ . Therefore we have  $Type(\lambda a.M) = Type((\lambda a.M)^\circ)$  and  $FV(\lambda a.M) = FV((\lambda a.M)^\circ)$ .  $\square$

DEFINITION 4.3 (\*-expansion). For every typed  $\lambda\rho$ -term, we define its \*-expansion as follows:

1.  $(x)^* \equiv x$ ,
2.  $(MN)^* \equiv (M^*N^*)$ ,
3.  $(\lambda x.M)^* \equiv \lambda x.M^*$ ,
4.  $((aM)^\tau)^* \equiv (aM^*)^\tau$ ,
5.  $(\lambda a.M)^* \equiv (\lambda a.M^*)^\circ$ .

LEMMA 4.4.  $Type(M^*) = Type(M)$  and  $FV(M^*) = FV(M)$ .

PROOF. By induction on the length of  $M$ . The only nontrivial case is  $M \equiv \lambda a.N$ . By the induction hypothesis,  $Type(N^*) = Type(N)$  and  $FV(N^*) = FV(N)$ . In this case we prove the claim by induction on the length of  $Type(N)$ . If  $Type(N)$  is an atom, then  $M^* \equiv \lambda a.N^*$ . Therefore we have  $Type(M^*) = Type(N^*) = Type(N) = Type(M)$  and  $FV(M^*) = FV(N^*) - \{a\} = FV(N) - \{a\} = FV(M)$ . Let  $Type(N)$  be a composite type  $\alpha \rightarrow \beta$ . Since  $Type(N^*) = \alpha \rightarrow \beta$ ,  $Type([\lambda x. b(xy)/a]N^*) = \alpha \rightarrow \beta$  by Lemma 3.2 where  $x : \alpha \rightarrow \beta$ ,  $y : \alpha$  and  $b : \beta$ .

Hence

$$\begin{aligned}
 \text{Type}(M^*) &= \text{Type}((\lambda a.N^*)^\circ) \\
 &= \text{Type}(\lambda y.(\lambda b.[\lambda x.b(xy)/a]N^*y)^\circ) \\
 &= \alpha \rightarrow \text{Type}((\lambda b.[\lambda x.b(xy)/a]N^*y)^\circ) \\
 &= \alpha \rightarrow \text{Type}(\lambda b.[\lambda x.b(xy)/a]N^*y) \quad (\text{by Lemma 4.2}) \\
 &= \alpha \rightarrow \text{Type}([\lambda x.b(xy)/a]N^*y) \\
 &= \alpha \rightarrow \beta = \text{Type}(M).
 \end{aligned}$$

Similarly, we can get  $FV(M^*) = FV(M)$ . □

LEMMA 4.5. *If  $\lambda a.M$  and  $N$  are typed  $\lambda\rho$ -terms and  $x$  is a  $\lambda$ -variable with type  $\text{Type}(N)$ , then*

$$[N/x](\lambda a.M)^\circ \equiv_x ([N/x](\lambda a.M))^\circ.$$

PROOF. By induction on the length of  $\text{Type}(\lambda a.M)$ . □

LEMMA 4.6. *If  $M$  and  $N$  are typed  $\lambda\rho$ -terms and  $\text{Type}(N) = \text{Type}(x)$ , then*

$$[N^*/x]M^* \equiv_x ([N/x]M)^*.$$

PROOF. By induction on the length of  $M$ . The only nontrivial case is  $M \equiv \lambda a.R$ . By the induction hypothesis,  $[N^*/x]R^* \equiv_x ([N/x]R)^*$ . We assume that  $a \notin FV(N)$ . If  $\text{Type}(R)$  is an atom, then

$$\begin{aligned}
 [N^*/x](\lambda a.R)^* &\equiv [N^*/x](\lambda a.R^*)^\circ \\
 &\equiv [N^*/x](\lambda a.R^*) \quad (\text{as } \text{Type}(R) \text{ is an atom}) \\
 &\equiv_x \lambda a.[N^*/x]R^* \\
 &\equiv_x \lambda a.([N/x]R)^* \quad (\text{by the induction hypothesis}) \\
 &\equiv (\lambda a.([N/x]R)^*)^\circ \quad (\text{as } \text{Type}(R) \text{ is an atom}) \\
 &\equiv (\lambda a.([N/x]R))^* \\
 &\equiv ([N/x](\lambda a.R))^*.
 \end{aligned}$$

Let  $Type(R)$  be a composite type  $\alpha \rightarrow \beta$ . Then

$$\begin{aligned}
[N^*/x](\lambda a.R)^* &\equiv [N^*/x](\lambda z.(\lambda b.[\lambda y.b(yz)/a]R^*z)^\circ) \\
&\equiv \lambda z.[N^*/x](\lambda b.[\lambda y.b(yz)/a]R^*z)^\circ \\
&\equiv_x \lambda z.([N^*/x](\lambda b.[\lambda y.b(yz)/a]R^*z)^\circ)^\circ \quad (\text{by Lemma 4.5}) \\
&\equiv \lambda z.(\lambda b.[\lambda y.b(yz)/a][N^*/x]R^*z)^\circ \\
&\equiv_x \lambda z.(\lambda b.[\lambda y.b(yz)/a]([N/x]R)^*z)^\circ \quad (\text{by the induction hypothesis}) \\
&\equiv (\lambda a.([N/x]R))^* \\
&\equiv ([N/x](\lambda a.R))^*. \quad \square
\end{aligned}$$

LEMMA 4.7. *If  $M$  and  $N$  are typed  $\lambda\rho$ -terms, then*

$$[\lambda x.a(xN^*)/a]M^* \equiv_x ([\lambda x.a(xN)/a]M)^*.$$

PROOF. Similar to that of Lemma 4.6.  $\square$

DEFINITION 4.8 ( $a\beta$ -contraction for typed  $\lambda\rho$ -terms). An  $a\beta$ -redex is an  $a$ -redex or a  $\beta$ -redex, that is

$$M \triangleright_{\lambda a\beta} N \quad \text{if } M \triangleright_{\lambda a} N \text{ or } M \triangleright_{\lambda\beta} N.$$

If  $P$  contains an  $a\beta$ -redex-occurrence  $\underline{R}$  and  $Q$  is the result of replacing  $\underline{R}$  by its contractum, we say that  $P$   $a\beta$ -contracts to  $Q$  ( $P \triangleright_{\lambda a\beta} Q$ ), and we call the triple  $\langle P, \underline{R}, Q \rangle$  an  $a\beta$ -contraction of  $P$ .

An  $a\beta$ -reduction for typed  $\lambda\rho$ -terms is defined in the same way as a  $\rho\beta$ -reduction for type free  $\lambda\rho$ -terms.

THEOREM 4.9 (Strong normalization theorem for  $a\beta$ -reduction). *For any typed  $\lambda\rho$ -term  $M$ , all  $a\beta$ -reductions starting at  $M$  are finite.*

PROOF. Similar to the case of typed  $\lambda$ -calculus, see [3].  $\square$

The following lemma is the key result to prove strong normalization for  $\rho\beta$ -reduction.

**LEMMA 4.10.** *For any typed  $\lambda\rho$ -terms  $M$  and  $N$ , if  $M \triangleright_{1\rho\beta} N$  then  $M^* \triangleright_{1a\beta} N^*$ .*

**PROOF.** Case 1: The redex is  $(\lambda x.P)Q$ .

$$\begin{aligned} ((\lambda x.P)Q)^* &\equiv (\lambda x.P^*)Q^* \\ &\triangleright_{1a\beta} [Q^*/x]P^* \\ &\equiv ([Q/x]P)^* \quad (\text{by Lemma 4.6}). \end{aligned}$$

Case 2: The redex is  $(aP)^{\sigma \rightarrow \tau} Q$ .

$$\begin{aligned} ((aP)^{\sigma \rightarrow \tau} Q)^* &\equiv (aP^*)^{\sigma \rightarrow \tau} Q^* \\ &\triangleright_{1a\beta} (aP^*)^\tau \\ &\equiv ((aP)^\tau)^*. \end{aligned}$$

Case 3: The redex is  $(\lambda a.P)Q$ .

$$\begin{aligned} ((\lambda a.P)Q)^* &\equiv (\lambda y.(\lambda b.[\lambda x.b(xy)/a]P^*y)^\circ)Q^* \\ &\triangleright_{1a\beta} [Q^*/y](\lambda b.[\lambda x.b(xy)/a]P^*y)^\circ \\ &\equiv ([Q^*/y]\lambda b.[\lambda x.b(xy)/a]P^*y)^\circ \quad (\text{by Lemma 4.5}) \\ &\equiv (\lambda b.[\lambda x.b(xQ^*)/a]P^*Q^*)^\circ \\ &\equiv (\lambda b.([\lambda x.b(xQ)/a]P)^*Q^*)^\circ \quad (\text{by Lemma 4.7}) \\ &\equiv (\lambda b.([\lambda x.b(xQ)/a]P)Q)^*\circ \\ &\equiv (\lambda b.([\lambda x.b(xQ)/a]P)Q)^*. \quad \square \end{aligned}$$

**THEOREM 4.11** (Strong normalization theorem for  $\rho\beta$ -reduction). *For any typed  $\lambda\rho$ -term  $M$ , all  $\rho\beta$ -reductions starting at  $M$  are finite.*

**PROOF.** Let  $M_1, M_2, \dots$  be an infinite  $\rho\beta$ -reduction. By Lemma 4.10, we can get an infinite  $a\beta$ -reduction  $M_1^*, M_2^*, \dots$ . This contradicts Theorem 4.9.  $\square$

Y. Andou [1] proved the weak normalization theorem for  $\rho\beta$ -reduction, that is, every typed  $\lambda\rho$ -term  $M$  has a normal form. The cut-elimination proof for LK only needs the weak normalization theorem, though we use the strong normalization theorem in the section 6.

## 5 Subformula Property for Normal Typed $\lambda\rho$ -Terms

DEFINITION 5.1 (Subterms). The set  $Subt(M)$  of all *subterms* of a typed  $\lambda\rho$ -term  $M$  is defined by induction on the length of  $M$  as follows:

1. if  $M$  is an atom,  $Subt(M) = \{M\}$ ,
2.  $Subt((PQ)) = Subt(P) \cup Subt(Q) \cup \{(PQ)\}$ ,
3.  $Subt((aP)^\sigma) = Subt(P) \cup \{a\} \cup \{(aP)^\sigma\}$
4.  $Subt((\lambda f.P)) = Subt(P) \cup \{f\} \cup \{(\lambda f.P)\}$ .

$\rho$ -variables are not  $\lambda\rho$ -terms but  $\rho$ -variables may be in  $Subt(M)$ .  $Subt(M)$  is a set of  $\lambda\rho$ -terms and  $\rho$ -variables. Let  $S$  be a set of  $\lambda\rho$ -terms and  $\rho$ -variables.  $Type(S)$  denotes the set  $\{Type(M) \mid M \in S\}$ .

NOTATION 5.2. Let  $\Gamma$  be a set of types. If a type  $\delta$  has an occurrence in  $\alpha$ , or in a type in  $\Gamma$ , we write as  $\delta \leq \alpha$ , or  $\delta \leq \Gamma$  respectively.

THEOREM 5.3 (Subformula property for typed  $\lambda\rho$ -terms in the normal form). *Let a typed  $\lambda\rho$ -term  $M$  be a  $\rho\beta$ -normal form. Then for every type  $\delta$  in  $Type(Subt(M))$ ,  $\delta \leq Type(FV(M) \cup \{M\})$ .*

PROOF. By induction on the length of  $M$ . The only nontrivial case is when  $M$  is of the form  $PQ$ . Since  $PQ$  is a  $\rho\beta$ -normal form, so are  $P$  and  $Q$ , and hence by the induction hypothesis, for every type  $\sigma$  in  $Type(Subt(P))$  and every type  $\tau$  in  $Type(Subt(Q))$ ,  $\sigma \leq Type(FV(P) \cup \{P\})$  and  $\tau \leq Type(FV(Q) \cup \{Q\})$ . Now, since  $PQ$  is a  $\rho\beta$ -normal form,  $P$  must be in the form  $xP_1 \cdots P_n$ . Hence  $Type(P) \leq Type(x)$  and for every type  $\delta$  in  $Type(Subt(M))$ ,  $\delta \leq Type(\{x\} \cup FV(M))$ . Therefore for every type  $\delta$  in  $Type(Subt(M))$ ,  $\delta \leq Type(FV(M) \cup \{M\})$ .  $\square$

## 6 Gentzen's LK and Typed $\lambda\rho$ -Terms

In this section we prove that a typed  $\lambda\rho$ -term corresponds to a proof in classical implicational logic and prove simultaneously the cut elimination theorem for the implicational fragment  $LK_{\rightarrow}$  of LK by using the strong normalization theorem for typed  $\lambda\rho$ -terms.

The calculus  $LK_{\rightarrow}$ , that we use here is the following:

DEFINITION 6.1. Let  $\Gamma$ ,  $\Theta$ ,  $\Delta$  and  $\Lambda$  be sets of types.  $\Gamma$ ,  $\Delta$  denotes the set  $\Gamma \cup \Delta$  and  $\Gamma \setminus \alpha$  denotes the set  $\Gamma - \{\alpha\}$ .

1. axiom:  $(I) \alpha \Rightarrow \alpha$ .
2. rules:

$$\frac{\Gamma \Rightarrow \Theta}{\alpha, \Gamma \Rightarrow \Theta} (w \Rightarrow), \quad \frac{\Gamma \Rightarrow \Theta}{\Gamma \Rightarrow \Theta, \alpha} (\Rightarrow w),$$

$$\frac{\Gamma \Rightarrow \Theta, \alpha \quad \alpha, \Delta \Rightarrow \Lambda}{\Gamma, \Delta \Rightarrow \Theta, \Lambda} (cut),$$

$$\frac{\Gamma \Rightarrow \Theta, \alpha \quad \beta, \Delta \Rightarrow \Lambda}{\alpha \rightarrow \beta, \Gamma, \Delta \Rightarrow \Theta, \Lambda} (\rightarrow \Rightarrow), \quad \frac{\Gamma \Rightarrow \Theta, \beta}{\Gamma \setminus \alpha \Rightarrow \Theta, \alpha \rightarrow \beta} (\Rightarrow \rightarrow).$$

LEMMA 6.2. *If  $\Gamma \Rightarrow \Theta$  is provable the system  $LK_{\rightarrow}$ , then there exists a typed  $\lambda\rho$ -term  $M$  such that  $\Gamma \ni Type(FV_z(M))$  and  $\Theta \ni Type(FV_{\rho}(M) \cup \{M\})$ .*

PROOF. By induction on the length of the  $LK_{\rightarrow}$  proof of  $\Gamma \Rightarrow \Theta$ . □

LEMMA 6.3. *For any  $\rho\beta$ -normal typed  $\lambda\rho$ -term  $M$ ,  $Type(FV_z(M)) \Rightarrow Type(FV_{\rho}(M) \cup \{M\})$  is provable without cut in the system  $LK_{\rightarrow}$ .*

PROOF. By induction on the length of  $M$ . The only nontrivial case is when  $M$  is of the form  $(PQ)$ . Since  $M$  is normal,  $P \equiv xP_1 \cdots P_n$  for some  $\lambda$ -variable  $x$  and normal  $\lambda\rho$ -terms  $P_1, \dots, P_n$ . Let  $Type(x)$  be  $\sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau \rightarrow \gamma$ . Then we have  $Type(P_1) = \sigma_1$ . By the induction hypothesis, there exists a cut free deduction in  $LK_{\rightarrow}$  proving  $Type(FV_z(P_1)) \Rightarrow Type(FV_{\rho}(P_1)), \sigma_1$ . Let  $z$  be a new  $\lambda$ -variable with a type  $\sigma_2 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau \rightarrow \gamma$ . The  $\lambda\rho$ -term  $zP_2 \cdots P_nQ$  is normal. Hence, by the induction hypothesis, there exists a cut free deduction of  $LK$  proving  $\sigma_2 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau \rightarrow \gamma, Type(FV_z(P_2 \cdots P_nQ)) \Rightarrow Type(FV_{\rho}(P_2 \cdots P_nQ)), \gamma$ . By the rule  $(\rightarrow \Rightarrow)$ , we get a cut free deduction of  $LK$  proving  $\sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau \rightarrow \gamma, Type(FV_z(P_1 \cdots P_nQ)) \Rightarrow Type(FV_{\rho}(P_1 \cdots P_nQ)), \gamma$ . As  $Type(FV_z(M)) \equiv \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau \rightarrow \gamma, Type(FV_z(P_1 \cdots P_nQ))$  and  $Type(FV_{\rho}(M) \cup \{M\}) \equiv Type(FV_{\rho}(P_1 \cdots P_nQ)), \gamma$ , we get a cut free deduction of  $LK$  proving  $Type(FV_z(M)) \Rightarrow Type(FV_{\rho}(M) \cup \{M\})$ . □

LEMMA 6.4. *For any typed  $\lambda\rho$ -term  $M$ ,  $Type(FV_z(M)) \Rightarrow Type(FV_{\rho}(M) \cup \{M\})$  is provable without cut in the system  $LK_{\rightarrow}$ .*

PROOF. By Theorem 4.11, there exists a  $\rho\beta$ -normal form  $M^*$  of  $M$ . By Lemma 6.3,  $Type(FV_z(M^*)) \Rightarrow Type(FV_{\rho}(M^*) \cup \{M^*\})$  is provable without cut

in the system  $LK_{\rightarrow}$ . By Theorem 3.3,  $Type(FV(M) \cup \{M\}) \supseteq Type(FV(M^*) \cup \{M^*\})$ . Hence, by the weakening rules ( $w \Rightarrow$ ) and ( $\Rightarrow w$ ), we can get a cut free deduction of  $Type(FV_{\lambda}(M)) \Rightarrow Type(FV_{\rho}(M) \cup \{M\})$ .  $\square$

**THEOREM 6.5.**  $\Gamma \Rightarrow \Theta$  is provable in the system  $LK_{\rightarrow}$  if and only if there exists a typed  $\lambda\rho$ -term  $M$  such that  $\Gamma \supseteq Type(FV_{\lambda}(M))$  and  $\Theta \supseteq Type(FV_{\rho}(M) \cup \{M\})$ .

**PROOF.** By Lemma 6.2 and Lemma 6.4.  $\square$

**THEOREM 6.6.** If  $\Gamma \Rightarrow \Theta$  is provable in the system  $LK_{\rightarrow}$ , then  $\Gamma \Rightarrow \Theta$  is provable without cut in the system  $LK_{\rightarrow}$ .

**PROOF.** By Lemma 6.2 and Lemma 6.4.  $\square$

## References

- [ 1 ] Yuuki Andou. A proof of the normalization theorem for  $\lambda\rho$ -calculus. Reports of Faculty of Literature, Hosei Univ., 50:1–5, 2005.
- [ 2 ] J. Roger Hindley. Basic Simple Type Theory, volume 42 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.
- [ 3 ] J. Roger Hindley and Jonathan P. Seldin. Lambda-calculus and Combinators, an Introduction. Cambridge University Press, 2008.
- [ 4 ] Yuichi Komori.  $\lambda\rho$ -calculus: A natural deduction for classical logic. Bulletin of the Section of Logic, 31:65–70, 2002.
- [ 5 ] Michel Parigot.  $\lambda\mu$ -calculus: An algorithmic interpretation of classical natural deduction. Lecture Notes in Computer Science, 624:190–201, 1992.
- [ 6 ] Michel Parigot. Proofs of strong normalization for second order classical natural deduction. Journal of Symbolic Logic, 62(4):1461–1479, 1997.

Department of Mathematics  
Faculty of Science, Chiba University  
Inage-ku Chiba 263-8522 Japan  
e-mail: komori.yuichi@gmail.com