$\lambda\rho$-calculus

Yuichi Komori
Arato Cho

abstract. In [K02], one of the authors introduced the system $\lambda\rho$-calculus and stated without proof that the strong normalization theorem hold. We have discovered an elegant lemma(Lemma 4.10). Here we prove the strong normalization theorem by the lemma. While the typed λ-calculus gives a natural deduction for intuitionistic implicational logic (cf. [H97]), the typed $\lambda\rho$-calculus gives a natural deduction for classical implicational logic. Our system is simpler than Parigot's $\lambda\mu$-calculus (cf. [P92]).
Keywords: λ-calculus, typed λ-calculus, normalization theorem, classical logic, $\lambda\rho$-calculus, $\lambda\mu$-calculus, LK.

1 The type free $\lambda\rho$-calculus

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

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

λ-variables are denoted by "u", "v", "w", "x", "y", "z". ρ-variables are denoted by "a", "b", "c", "d". A *term-variable* is a λ-variable or a ρ-variable. Term-variables are denoted by "f", "g", "h". Distinct letters denotes distinct variables unless stated otherwise.

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

$$\begin{array}{rcl}
(aM)N & \rhd_{1a} & (aM), \\
(\lambda x.M)N & \rhd_{1\beta} & [N/x]M, \\
(\lambda a.M)N & \rhd_{1\rho} & \lambda b.([\lambda x.b(xN)/a]M)N, \text{ where } b \text{ is the first } \rho\text{-variable} \\
 & & \text{and } x \text{ is the first } \lambda\text{-variable such that } b \text{ and } x \text{ do} \\
 & & \text{not occur in } aMN, \\
M & \rhd_{1\rho\beta} & N \quad \text{if } M \rhd_{1a} N, M \rhd_{1\beta} N \text{ or } M \rhd_{1\rho} N.
\end{array}$$

If P contains a $\rho\beta$-redex-occurence $\underline{R}$ and Q is the result of replacing this by its contractum, we say that P *$\rho\beta$-contracts to Q* ($P \rhd_{1\rho\beta} Q$), 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 possibly empty sequence of $\rho\beta$-contractions with form

$$\langle P_1, \underline{R_1}, Q_1\rangle, \langle P_2, \underline{R_2}, Q_2\rangle, \ldots$$

where $P_1 \equiv_\alpha P$ and $Q_i \equiv_\alpha P_{i+1}$ for $i = 1, 2, \ldots$. We say a finite reduction is from P to Q if either it has $n \geq 1$ 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

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

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

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

$$P \rhd_{\rho\beta} T \textit{ and } Q \rhd_{\rho\beta} T.$$

Proof. Similar to the case of β-reduction, see [HS86]. ■

2 Typed $\lambda\rho$-terms

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

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

Aribitrary types are denoted by lower-case Greek letters except "λ" and "ρ".

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-variables is assumed to have one type. For any type τ, an infinite sequence of λ-variables with type τ and an infinite sequence of ρ-variable with type τ are assumed to exist.

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

1. A λ-variable x with type τ is a typed $\lambda\rho$-term, called an *atom*, and $x : \tau$.
2. if M and N are typed $\lambda\rho$-terms and $M : \sigma \to \tau$ and $N : \sigma$, then the expression (MN) is a typed $\lambda\rho$-term called an *application* and $(MN) : \tau$,
3. if M is a typed $\lambda\rho$-term and $M : \tau$ and a is a ρ-variable with type τ, 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 λ-variable with a type σ, then the expression $(\lambda x.M)$ is a typed $\lambda\rho$-term called a *λ-abstract* and $(\lambda x.M) : \sigma \to \tau$,
5. if M is a typed $\lambda\rho$-term and $M : \tau$ and a is a ρ-variable with the type τ, then the expression $(\lambda a.M)$ is a typed $\lambda\rho$-term called a *ρ-abstract* and $(\lambda a.M) : \tau$.

Typed $\lambda\rho$-terms will be abbriviated 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. $FV(M)$, which is the set of all term variables with a type free in M, is defined thus:

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 λ-variables in $FV(M)$ and the set of all ρ-variables in $FV(M)$, respectively.

EXAMPLE 2.4 (Peirce's Law).

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

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

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

$$\cfrac{\cfrac{\cfrac{x:(\alpha\to\beta)\to\alpha \qquad \cfrac{\cfrac{a:\alpha \quad y:\alpha}{\beta}}{\alpha\to\beta}\,\lambda y}{\alpha}}{\alpha}\,\lambda a}{((\alpha\to\beta)\to\alpha)\to\alpha}\,\lambda x \qquad ,$$

or in more a redundant form as follows:

$$\cfrac{\cfrac{\cfrac{x:(\alpha\to\beta)\to\alpha \qquad \cfrac{\cfrac{a:\alpha \quad y:\alpha}{ay:\beta}}{\lambda y.ay:\alpha\to\beta}}{x(\lambda y.ay):\alpha}}{\lambda a.x(\lambda y.ay):\alpha}}{\lambda xa.x(\lambda y.ay):((\alpha\to\beta)\to\alpha)\to\alpha}$$

Definition 2.5 (Type-erasure and typability). We assume the existance of two mappings j and k such that j is an one-to-one onto mapping from the set of all λ-variables and g is one-to-one onto mapping from the set of all ρ-variables with a type to the set of all ρ-variables. For the 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. $er(M)$ is defined as follows:

1. $er(x) \equiv x$,
2. $er((MN)) \equiv (er(M)er(N))$,
3. $er((aM)^\sigma) \equiv (aer(M))$,
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 λ-variable x with the type $Type(N)$, the substitution of N for x in M $[N/x]M$ is defined as the usual. For a typed $\lambda\rho$-term M and ρ-variables a, b such that $Type(a) = Type(b)$, the substitution of b for a in M $[b/a]M$ is defined also as the 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 ρ-variable. Remark that the expression $\lambda x.b(xN)$ is not a typed $\lambda\rho$-term.

Definition 2.6 (Substitution an expression $\lambda x.b(xN)$ for a ρ-variable). For typed $\lambda\rho$-terms M, N, a ρ-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 $Type(x) = Type(a) = \alpha\to\beta$, $b:\beta$ and $N:\alpha$.

1. $[\lambda x.b(xN)/a]M \equiv M \quad$ if $a \notin FV(M)$,
2. $[\lambda x.b(xN)/a](MR) \equiv ([\lambda x.b(xN)/a]M[\lambda x.b(xN)/a]R) \quad$ if $a \in FV(MR)$,
3. $[\lambda x.b(xN)/a](\lambda y.M) \equiv \lambda y.[\lambda x.b(xN)/a]M \quad$ if $a \in FV(M)$ and $y \notin FV(\lambda x.b(xN))$,
4. $[\lambda x.b(xN)/a](\lambda y.M) \equiv \lambda z.[\lambda x.b(xN)/a][z/y]M \quad$ if $a \in FV(M)$ and $y \in FV(\lambda x.b(xN))$,
5. $[\lambda x.b(xN)/a](cM)^\sigma \equiv (b[\lambda x.b(xN)/a]M)^\sigma \quad$ if $a \in FV(M)$ and $c \not\equiv a$,
6. $[\lambda x.b(xN)/a](aM)^\sigma \equiv (\lambda x.(b(xN))^\sigma)[\lambda x.b(xN)/a]M$,
7. $[\lambda x.b(xN)/a](\lambda c.M) \equiv \lambda c.[\lambda x.b(xN)/a]M \quad$ if $a \in FV(\lambda c.M)$ and $c \notin FV(bN)$,
8. $[\lambda x.b(xN)/a](\lambda c.M) \equiv \lambda d.[\lambda x.b(xN)/a][d/c]M \quad$ if $a \in FV(\lambda c.M)$ and $c \in FV(bN)$.

(In 4 z is the first λ-variable with the type $Type(y)$ which does not occur in xNM. In 8 d is the first ρ-variable with the 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 $(aM)^{\sigma\to\tau}N$, $(\lambda x.M)N$ 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

$$\begin{array}{rcl}
(aM)^{\sigma\to\tau}N & \rhd_{1a} & (aM)^\tau, \\
(\lambda x.M)N & \rhd_{1\beta} & [N/x]M, \\
(\lambda a.M)N & \rhd_{1\rho} & \lambda b.([\lambda x.b(xN)/a]M)N, \text{ where } b \text{ is the first } \rho\text{-variable} \\
 & & \text{and } x \text{ is the first } \lambda\text{-variable such that } b : Type(MN), \\
 & & x : Type(a) \text{ and } b \text{ and } x \text{ do not occur in } aMN, \\
M & \rhd_{1\rho\beta} & N \quad \text{if } M \rhd_{1a} N, M \rhd_{1\beta} N \text{ or } M \rhd_{1\rho} N.
\end{array}$$

If P contains a $\rho\beta$-redex-occurence $\underline{R}$ and Q is the result of replacing this by its contractum, we say P *$\rho\beta$-contracts to* Q ($P \rhd_{1\rho\beta} Q$), 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 threorem for typed $\lambda\rho$-terms). *Let M, P and Q be typed $\lambda\rho$-terms. If $M \rhd_{\rho\beta} P$ and $M \rhd_{\rho\beta} Q$, then there exists a typed $\lambda\rho$-term T such that*

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

Proof. Simmlar to the case of β-reduction, see [HS86]. ■

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 λ-variable with the 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, $Type(x) = Type(a) = \sigma \to \tau$, $b : \tau$, $Q : \sigma$ 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 $P \equiv (aP_1)^\gamma$. Then $P_1 : \sigma \to \tau$ and $[\lambda x.b(xQ)/a](aP_1)^\gamma \equiv (\lambda x.(b(xQ))^\gamma)[\lambda x.b(xQ)/a]P_1$. Now we have $Type([\lambda x.b(xQ)/a]P) =$
$Type(P) = \gamma$ and $FV([\lambda x.b(xQ)/a]P) = FV([\lambda x.b(xQ)/a]P_1) \cup FV(Q) \cup \{b\} \subseteq (FV(P) - \{a\}) \cup FV(Q) \cup \{b\}$. ■

THEOREM 3.3 (Subject-reduction theorem). *If $P \rhd_{\rho\beta} Q$, 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 that P is a redex and Q is its contractum. It is enough to prove that if $P \rhd_{1\rho\beta} Q$, then $Type(Q) = Type(P)$ and $FV(Q) \subseteq FV(P)$.
Case 1: $P \equiv (aP_1)^{\sigma\to\tau} P_2$ 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 $FV(Q) \subseteq FV(P)$. ■

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 $*$-expansion and use the strong normalization theorem for typed λ-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 τ is an atomic type, then $(\lambda a.M)^\circ \equiv (\lambda a.M)$,

2. if $\tau \equiv \alpha \to \beta$, 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 λ-variable with the type $\alpha \to \beta$, the second λ-variable with the type α and the first ρ-variable with the type β which do not occur in aM.

By the above definition, if $M : \sigma_1 \to \cdots \to \sigma_n \to p$, then $(\lambda a.M)^\circ \rhd_\beta \lambda y_1 \cdots y_n b.[\lambda x.b(xy_1 \cdots y_n)/a]My_1 \cdots y_n$ where $x : \sigma_1 \to \cdots \to \sigma_n \to p$, $y_1 : \sigma_1 \cdots y_n : \sigma_n$ and $b : p$.

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 $\lambda a.M : \alpha \to \beta$, then

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

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

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 $FV(M^*) = FV(N^*) - \{a\} = FV(N) - \{a\} = FV(N)$. Let $Type(N)$ is a composite type $\alpha \to \beta$. Since $Type(N^*) = \alpha \to \beta$, $Type([\lambda x.b(xy)/a]N^*) = \alpha \to \beta$ by Lemma 3.2 where $x : \alpha \to \beta$, $y : \alpha$ and $b : \beta$. Hence

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

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 λ-variable with the type $Type(N)$, then*

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

Proof. By induction on the length of $Type(\lambda a.M)$. ■

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

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

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

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

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

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

■

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

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

Proof. Similar to that of Lemma 4.6. ■

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

$$M \quad \rhd_{1a\beta} \quad N \quad \text{if } M \rhd_{1a} N \text{ or } M \rhd_{1\beta} N.$$

If P contains an $a\beta$-redex-occurence $\underline{R}$ and Q is the result of replacing $\underline{R}$ by its contractum, we say P *$a\beta$-contracts to* Q ($P \rhd_{1a\beta} Q$), 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 λ-calculus, see [HS86]. ∎

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 $M \rhd_{1\rho\beta} N$ then $M^* \rhd_{1a\beta} N^*$.*

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

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

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

$$\begin{array}{rcl} ((aP)^{\sigma\to\tau}Q)^* & \equiv & (aP^*)^{\sigma\to\tau}Q^* \\ & \rhd_{1a\beta} & (aP^*)^{\tau} \\ & \equiv & ((aP)^{\tau})^*. \end{array}$$

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

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

∎

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 $M_1, M_2, \ldots$ be an infinite $\rho\beta$-reduction. By Lemma 4.10, we can get an infinite $a\beta$-reduction $M_1^*, M_2^*, \ldots$. It contradicts Theorem 4.9. ∎

5 Subformula property for normal typed $\lambda\rho$-terms

DEFINITION 5.1 (Subterms). The set $Subt(M)$ of all *subterm* 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. $Subt((\lambda f.P)) = Subt(P) \cup \{(\lambda f.P)\}$.

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

NOTATION 5.2. Let Γ be a set of types. If a type δ has an occurrence in α, or in a type in Γ, 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 δ 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 σ in $Type(Subt(P))$ and every type τ 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 δ in $Type(Subt(M))$, $\delta \leq Type(\{x\} \cup FV(M))$. Therefore for every type δ in $Type(Subt(M))$, $\delta \leq Type(FV(M) \cup \{M\})$. ■

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

In this section, we shall prove that a typed $\lambda\rho$-term is a proof of the 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 Γ, Θ, Δ and Λ be sets of types. Γ, Δ denotes the set $\Gamma \cup \Delta$ and $\Gamma \backslash \alpha$ denotes the set $\Gamma - \{\alpha\}$.

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

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

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

$$\frac{\Gamma \Rightarrow \Theta, \alpha \quad \beta, \Delta \Rightarrow \Lambda}{\alpha \to \beta, \Gamma, \Delta \Rightarrow \Theta, \Lambda}\ (\to\Rightarrow), \qquad \frac{\Gamma \Rightarrow \Theta, \beta}{\Gamma \backslash \alpha \Rightarrow \Theta, \alpha \to \beta}\ (\Rightarrow\to).$$

THEOREM 6.2. *If* $\Gamma \Rightarrow \Theta$ *is provable the system* $LK_{\to}$*, then 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 induction on the length of the $LK_{\to}$ proof of $\Gamma \Rightarrow \Theta$. ■

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

Proof. By induction on the length of M. The only nontrivial case is when M is of the form (PQ). Since M is normal, $P \equiv yP_1 \cdots P_n$ for some λ-variable y and normal $\lambda\rho$-terms $P_1, \ldots, P_n$. Let $Type(x)$ be $\sigma_1 \to \cdots \to \sigma_n \to \tau \to \gamma$. Then we have $Type(P_1) = \sigma_1$. By the induction hypothesis, there exists a cut free deduction in $LK_{\to}$ proving $Type(FV_\lambda(P_1)) \Rightarrow Type(FV_\rho(P_1)), \sigma_1$. Let z be a new λ-variable with a type $\sigma_2 \to \cdots \to \sigma_n \to \tau \to \gamma$. 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 $\sigma_2 \to \cdots \to \sigma_n \to \tau \to \gamma, Type(FV_\lambda(P_2 \cdots P_nQ)) \Rightarrow Type(FV_\rho(P_2 \cdots P_nQ)), \gamma$. By the rule $(\to\Rightarrow)$, we get a a cut free deduction of LK proving $\sigma_1 \to \cdots \to \sigma_n \to \tau \to \gamma, Type(FV_\lambda(P_1 \cdots P_nQ)) \Rightarrow Type(FV_\rho(P_1 \cdots P_nQ)), \gamma$. As $Type(FV_\lambda(M)) \equiv \sigma_1 \to \cdots \to \sigma_n \to \tau \to \gamma, Type(FV_\lambda(P_1 \cdots P_nQ))$ 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 $Type(FV_\lambda(M)) \Rightarrow Type(FV_\rho(M) \cup \{M\})$. ■

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

Proof. By Theorem 4.11, there exists a $\rho\beta$-normal form M^* of M. By Lemma 6.3, $Type(FV_\lambda(M^*)) \Rightarrow Type(FV_\rho(M^*) \cup \{M^*\})$ is provable without cut in the system $LK_{\to}$. 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 $Type(FV_\lambda(M)) \Rightarrow Type(FV_\rho(M) \cup \{M\})$. ■

THEOREM 6.5. $\Gamma \Rightarrow \Theta$ *is provable the system* $LK_{\to}$ *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. ■

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

Proof. By Lemma 6.2 and Lemma 6.4. ■

References

[H97] J. Roser Hindley. *Basic Simple Type Theory*, Vol. 42 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 1997.

[HS86] J. Roger Hindley and Jonathan P. Seldin. *Introduction to combinators and λ-calculus*, Vol. 1 of *London Mathematical Society Student Texts*, Cambridge University Press, 1986.

[P92] Michel Parigot. *$\lambda\mu$-CALCULUS: AN ALGORITHMIC INTERPRETATION OF CLASSICAL NATURAL DEDUCTION*, Lecture Notes in Computer Science 624, 190-201, 1992.

[K02] Yuichi Komori. *$\lambda\rho$-Calculus: A Natural Deduction for Classical Logic*, BULLETIN OF THE SECTION OF LOGIC, VOL. 31, No. 2, 65-70, 2002.

Yuichi KOMORI
Chiba University
Inaga-ku Chiba
263-8522 JAPAN
komori@math.s.chiba-u.ac.jp
Arato CHO
aratoc@g.math.s.chiba-u.ac.jp