In [K02], one of the authors has posed a new system $\lambda\rho$-calculus and stated without proof that the strong normalization theorem holds. We will give a proof of it. While the type assignment system $TA_\lambda$ gives a natural deduction for intuitionistic implicational logic (cf. [H97]), $TA_{\lambda\rho}$ gives a natural deduction for classical implicational logic. Our system is simpler than Parigot’s $\lambda\mu$-calculus (cf. [P92]).

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

Definition 1.1 ($\lambda\rho$-terms). An infinite sequence of $\lambda$-variables is assumed to be given, and an infinite sequence of $\rho$-variables is assumed to be given. Then linguistic expressions called $\lambda\rho$-terms are defined thus:

1. each $\lambda$-variable is a $\lambda\rho$-term, called an atom or atomic term;
2. if $M$ and $N$ are $\lambda\rho$-term 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 an 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 it called a $\lambda$-abstract or a $\rho$-abstract respectively.)

$\lambda$-variables are denoted by $u, v, w, x, y, z$, with or without number-subscripts. $\rho$-variables are denoted by $a, b, c, d$, with or without number-subscripts. A term-variable means a $\lambda$-variable or a $\rho$-variable. Term-variables are denoted by $f, g$.

*This article is an abstract and details will be published elsewhere.
h, with or without number-subscripts. Distinct letters denotes distinct variables unless otherwise stated.

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 \( L, M, N, P, Q, R, S, T \), with or without number-subscripts. For \( \lambda \rho \)-term we shall say just term.

\( FV(M) \) is the set of all variables free in \( M \). For example, \( FV(\lambda x.b.a(x(by))) = \{a, y\} \).

**Definition 1.2 (\( \beta \rho \)-contraction).** A \( \beta \rho \)-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 \( \lambda a. ([\lambda x.a(xN)/a]M)N \) respectively. The re-write rules are

\[
\begin{align*}
(aM)N & \triangleright_{1\beta \rho} (aM), \\
(\lambda x.M)N & \triangleright_{1\beta \rho} [N/x]M, \\
(\lambda a.M)N & \triangleright_{1\beta \rho} \lambda a.([\lambda x.a(xN)/a]M)N.
\end{align*}
\]

If \( P \) containd a \( \beta \rho \)-redex-occurence \( R \) and \( Q \) is the result of replacing this by its contractum, we say \( P \beta \rho \)-contracts to \( Q \) (\( P \triangleright_{1\beta \rho} Q \)).

The notion of \( \beta \rho \)-reduction and the notation \( P \triangleright_{\beta \rho} Q \) are defined as usual.

**Theorem 1.3 (Church-Rosser threorem for \( \beta \rho \)-reduction).** If \( M \triangleright_{\beta \rho} P \) and \( M \triangleright_{\beta \rho} Q \), then there exists \( T \) such that

\( P \triangleright_{\beta \rho} T \) and \( Q \triangleright_{\beta \rho} T \).

**Proof.** Similar to the case of \( \beta \)-reduction, see [HS86].

2 Assigning types to 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 \( \sigma \) 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.

**Definition 2.2 (Type-assignment).** A type-assignment is any expression

\[ M : \tau \]

where \( M \) is a \( \lambda \rho \)-term or a \( \rho \)-variable and \( \tau \) is a type; we call \( M \) its subject and \( \tau \) is its predicate.
Definition 2.3 (The system $TA_{\lambda\rho}$). $TA_{\lambda\rho}$ is a Natural Deduction system. Its formulas are type-assignments. $TA_{\lambda\rho}$ has no axioms and has four rules called ($\rightarrow E$), ($\rightarrow I$), (Absurd) and (Rati), as follows.

Deduction rules of $TA_{\lambda\rho}$:

$$
\frac{P : \sigma \rightarrow \tau \quad Q : \sigma}{PQ : \tau} (\rightarrow E), \quad \frac{\Pi}{\lambda x.P : \sigma \rightarrow \tau} (\rightarrow I),
$$

$$
\frac{a : \tau \quad P : \tau}{aP : \sigma} \text{ (Absurd)}, \quad \frac{P : \tau}{\lambda a.P : \tau} \text{ (Rati)}.
$$

Example 2.4 (Peirce’s Law).

$$
\frac{\alpha : \alpha \rightarrow \beta \quad y : (\alpha \rightarrow \beta) \rightarrow \alpha}{\lambda y.(\lambda x.ax) : ((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow \alpha} (\rightarrow I)
$$

The above $TA_{\lambda\rho}$-deduction is written in more compact style:

$$
\frac{\alpha : \alpha \quad x : \alpha}{ax : \beta} \text{ (Absurd)}
$$

$$
\frac{y : (\alpha \rightarrow \beta) \rightarrow \alpha \quad \lambda x.ax : \alpha \rightarrow \beta}{\lambda y.(\lambda x.ax) : ((\alpha \rightarrow \beta) \rightarrow \alpha) \rightarrow \alpha} (\rightarrow I)
$$

3 Strong Normalization Theorem for $TA_{\lambda\rho}$

$\beta\rho$-reductions of deductions of $TA_{\lambda\rho}$ correspond to $\beta\rho$-reductions of $\lambda\rho$-terms. We prove the strong normalization theorem for deductions of $TA_{\lambda\rho}$, that is, for every deduction of $TA_{\lambda\rho}$ $\Pi$, all reductions starting at $\Pi$ are finite. To prove the theorem, we introduce $\circ$-expansion and use the strong normalization theorem for deductions of $TA_{\lambda}$.

Definition 3.1 ($\circ$-translation). For every deduction of $TA_{\lambda\rho}$, of which the last rule is $\text{Rati}$, we define $\circ$-translation as follows:

1. $\Pi^\circ \equiv \Pi$ , where $\Pi \equiv \Pi^1 \quad M : \alpha$ and $\alpha$ is an atomic type;

2. $\Pi^\circ \equiv \Pi^2$ , where $\Pi^2 \equiv (\Pi_2)^\circ \quad N : \gamma$, and

$$
\frac{\lambda y.N : \beta \rightarrow \gamma}{\lambda a.M : \beta \rightarrow \gamma}
$$
\[ \begin{array}{c}
\frac{x : \beta \rightarrow \gamma \quad y : \beta}{\bar{a} : \gamma} \\
xy : \gamma \\
a(xy) : \delta
\end{array} \]
\[ \Pi_2 \equiv \frac{\lambda x.a(xy) : (\beta \rightarrow \gamma) \rightarrow \delta}{[\lambda x.a(xy)/a] \Pi_1} \]
\[ \frac{[\lambda x.a(xy)/a] M : \beta \rightarrow \gamma \quad y : \beta}{[\lambda x.a(xy)/a] M y : \gamma} \]
\[ \lambda a.[\lambda x.a(xy)/a] M y : \gamma \]

**Definition 3.2 (\text{*}-expansion).** For every deduction of \( TA_{\lambda \rho} \), we define \( \text{*}-\)expansion as follows:

1. \((x : \alpha)^* \equiv x : \alpha ;\)
2. \(\Pi^* \equiv M^* : \alpha \rightarrow \beta \quad N^* : \alpha \quad \text{where} \quad \Pi \equiv M : \alpha \rightarrow \beta \quad N : \alpha ;\)
3. \(\Pi^* \equiv \frac{x : \alpha}{(\Pi_1)^*} \quad M^* : \beta \quad \text{where} \quad \Pi \equiv \frac{x : \alpha}{\Pi_1} \quad M : \beta ;\)
4. \(\Pi^* \equiv \frac{a : \alpha}{(\Pi_1)^*} \quad M^* : \beta \quad \text{where} \quad \Pi \equiv \frac{a : \alpha}{\Pi_1} \quad M : \beta ;\)
5. \(\Pi^* \equiv (\Pi_1)^* \quad \text{where} \quad \Pi \equiv \frac{a : \alpha}{\Pi_2} \quad M^* : \alpha \quad \text{and} \quad \Pi_1 \equiv \frac{a : \alpha}{M : \alpha} \quad \text{and} \quad \Pi_2 \equiv \frac{M^* : \alpha}{a M : \beta} .\)

**Definition 3.3 (\(\beta \alpha\)-reduction).** A \(\beta \alpha\)-reduction is a \(\beta \rho\)-reduction which does not allow a contraction \((\lambda a.M)N \triangleright_{1\beta \rho} \lambda a.(([\lambda x.a(xy)/a] M) N)\).

**Theorem 3.4 (Strong normalization theorem for \(\beta \alpha\)-reduction).** For every deduction \( \Pi \) of \( TA_{\lambda \rho} \), all \(\beta \alpha\)-reductions starting at \( \Pi \) are finite.

**Proof.** Similar to the case of \( TA_{\lambda} \), see [HS86]. \hfill \blacksquare

**Theorem 3.5 (Strong normalization theorem for \( TA_{\lambda \rho} \)).** For every deduction \( \Pi \) of \( TA_{\lambda \rho} \), all \(\beta \rho\)-reductions starting at \( \Pi \) are finite.

**Proof.** We can prove that \((\Pi_1)^* \triangleright_{1\beta \alpha} (\Pi_2)^* \) if \( \Pi_1 \triangleright_{1\beta \rho} \Pi_2 \). So we get an infinite sequence of \(\beta \alpha\)-reductions from a infinite sequence of \(\beta \rho\)-reductions. That is, strong normalization theorem for \(\beta \alpha\)-reduction leads strong normalization theorem for \( TA_{\lambda \rho} \). \hfill \blacksquare
References


