論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Lambda-Calculus: A Simplified Proof of the Church-Rosser Theorem and an Extension of the Curry-Howard Correspondence
著者(和文)	松田直祐
Author(English)	Naosuke Matsuda
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第10105号, 授与年月日:2016年3月26日, 学位の種別:課程博士, 審査員:鹿島 亮,小島 定吉,増原 英彦,渡辺 治,脇田 建
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第10105号, Conferred date:2016/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

LAMBDA-CALCULUS:

A SIMPLIFIED PROOF OF THE CHURCH-ROSSER THEOREM AND AN EXTENSION OF THE CURRY-HOWARD CORRESPONDENCE

A THESIS
SUBMITTED TO TOKYO INSTITUTE OF TECHNOLOGY

Adviser: KASHIMA Ryo

By
MATSUDA Naosuke
March, 2016

Abstract

This thesis consists of the following three chapters:

- **Chapter 1:** This chapter gives a brief introduction to the topics "lambda-calculus", "proof theory" and "Curry-Howard correspondence".
- **Chapter 2:** The Church-Rosser property of the lambda-beta-calculus is an important property which guarantees that the lambda-beta-calculus is well-behaved as a computation model. In this chapter, we give a new proof to the Church-Rosser theorem by improving the proof given in (Takahashi, 1989). Furthermore, we explain that our proof method can be applied to abstract term rewriting systems. The result in this chapter was given by Komori, Yamakawa and the author in (Komori, Matsuda, & Yamakawa, 2014).
- **Chapter 3:** In this chapter, we give a typed lambda-calculus, called the intuitionistic lambda-rho-calculus, which corresponds to the implicational fragment of intuitionistic logic and can capture the work of the operators **catch** and **throw** of functional programming language. Because the work of the operators cannot be captured with the lambda-beta-calculus, this result is regarded as an extension of the Curry-Howard correspondence. Furthermore, we show some important properties, such as the strong normalization theorem, of the system. The result of this chapter was given by Fujita, Kashima, Komori and the author in (Fujita, Kashima, Komori, & Matsuda, 2015; Matsuda, 2015c).

Acknowledgments

I would like to thank Prof. Ryo Kashima and Prof. Yuichi Komori, my mentors. I learned a lot of things from them and, of course, the result of this thesis is very influenced by their works. Furthermore, Prof. Kashima gave me a lot of advice when I wrote this thesis.

I would like to thank Prof. Ken-etsu Fujita, Prof. Toshihiko Kurata and Prof. Koji Nakazawa. I had some discussions with them and it developed the result of this thesis.

I also wish to thank Prof. Sadayoshi Kojima, Prof. Hidehiko Masuhara, Prof. Ken Wakita and Prof. Osamu Watanabe. They refereed this thesis and gave me many useful comments.

Finally, I would like to thank Junko Matsuda, my wife, for her support during this work.

Contents

Chapter 1 Introduction and preliminaries

In this thesis, we mainly treat the following two objects:

- ⋆ **Lambda-calculus**: Lambda-abstraction is a basic operation which constructs a new higher-order function from some higher-order functions. Lambda-calculi are formal systems, introduced by Church, which formalize the lambda-abstraction. Because many functional programming languages use the lambda-abstraction to construct programs, lambda-calculi are studied as a basic theory of functional programming languages today.

- ⋆ **Proof theory**: Proof theory is a field of mathematical logic which studies "mathematical proofs" as mathematical objects formally by representing mathematical proofs as formal objects and analyzing them.

It is well-known that there is a closed connection called the *Curry-Howard correspondence* between them. The aim of this thesis is to give a study on these two topics and give an extension of the Curry-Howard correspondence. This chapter explains the background of our work briefly.

Section 1.1 gives an introduction to lambda-calculus. Section 1.2 gives an introduction to proof theory. Section 1.3 gives an introduction to the Curry-Howard correspondence.

1.1 Lambda-calculus

Lambda-abstraction is a basic operation which constructs a new higher-order function from some higher-order functions. For example, a function which receives a unary function on $\mathcal{N}$ (we write the set of natural numbers as $\mathcal{N}$ in this thesis) and a natural number and returns a value obtained by applying the unary function to the natural number twice is written as

$$(\lambda x : \mathcal{N} \to \mathcal{N}.(\lambda y : \mathcal{N}.(x(xy))))$$

with the lambda-abstraction notation. The intended work of this function is as follows

$$(\lambda x : \mathcal{N} \to \mathcal{N}.(\lambda y : \mathcal{N}.(x(xy))))(f)(3) = f(f(3)).^{1}$$

[1] A more detailed introduction to lambda-abstraction notation is written in (Hindley & Seldin, 2008; Takahashi, 1991) for example.

Lambda-calculi, introduced by Church (Church, 1944), are formal systems which formalize the work of the lambda-abstraction. Because many functional programming languages use the lambda-abstraction to construct programs [2], lambda-calculi are studied as a basic theory of functional programming languages today [3] .

1.1.1 Untyped lambda-beta-calculus

We first give an introduction to untyped lambda-beta-calculus. Suppose a countable set $\mathrm{V}_\lambda = \{\mathtt{x}_1, \mathtt{x}_2, \ldots\}$ of variables, called *lambda-variables*, is given. Then the set Tm_λ of *lambda-terms* is defined as follows:

1. Each lambda-variable is in Tm_λ.

2. If M, N are both in Tm_λ then (MN) is in Tm_λ. A term of this form is called a *lambda-application*.

3. If M is in Tm_λ and $x \in \mathrm{V}_\lambda$ then $(\lambda x.M)$ is in Tm_λ. A term of this form is called a *lambda-abstraction*.

We use metavariables $x, y, z, \ldots$ for lambda-variables, and use metavariables $M, N, P, Q, \ldots$ for lambda-terms. Parentheses will be omitted in such a way that $MNPQ$ denotes the term $(((MN)P)Q)$, $\lambda x.MN$ denotes $(\lambda x.(MN))$ and $\lambda x_1 \ldots x_n.M$ denotes $(\lambda x_1.(\lambda x_2.(\ldots(\lambda x_n.M)\ldots)))$.

Definition 1.1 (Free variable, subterm)**.** We define the sets $\mathrm{FV}_\lambda(M) \subset \mathrm{V}_\lambda$ and $\mathrm{Sub}(M) \subset \mathrm{Tm}_\lambda$, for each lambda-term M, as follows:

1. $\mathrm{FV}_\lambda(x) = \{x\}$ and $\mathrm{Sub}(x) = \{x\}$.

2. $\mathrm{FV}_\lambda(MN) = \mathrm{FV}_\lambda(M) \cup \mathrm{FV}_\lambda(N)$ and $\mathrm{Sub}(MN) = \mathrm{Sub}(M) \cup \mathrm{Sub}(N) \cup \{MN\}$.

3. $\mathrm{FV}_\lambda(\lambda x.M) = \mathrm{FV}_\lambda(M) \setminus \{x\}$ and $\mathrm{Sub}(\lambda x.M) = \mathrm{Sub}(M) \cup \{\lambda x.M\}$.

We say x is *free* in M if $x \in \mathrm{FV}_\lambda(M)$. We say N is *subterm* of M if $N \in \mathrm{Sub}(M)$.

Definition 1.2 (Substitution)**.** For each M, N, we define $[N/x]M$ as follows:

1. $[N/x]M$ is M if $\mathrm{FV}_\lambda(M) = \emptyset$.

2. $[N/x]x$ is N.

3. $[N/x](PQ)$ is $[N/x]P[N/x]Q$.

[2] See (Abelson, Sussman, & Sussman, 1996, subsection 1.3.2.) or (Pierce, 2002, section 5.) for example.
[3] See (Abelson et al., 1996; Gunter, 1992; Pierce, 2002) for example.

4. $[N/x](\lambda y.P)$ is $\lambda y.[N/x]P$ if y is not x and $y \notin \mathrm{FV}_\lambda(N)$.

5. $[N/x](\lambda y.P)$ is $\lambda z.[N/x][z/y]P$, where z is the first lambda-variable [4] in $\mathrm{V}_\lambda \setminus \mathrm{FV}_\lambda(P)$, if y is not x and $y \in \mathrm{FV}_\lambda(N)$.

Here we choose to apply the rule with smallest number if many rules can apply to M [5].

Intuitively speaking, $[N/x]M$ is the lambda-term obtained from M by replacing all free occurrences of x by N.

Definition 1.3 (alpha-equivalent)**.** We say M is *alpha-equivalent* to N if $M \sim_\alpha N$ can be derived by the following rules:

(ρ) $M \sim_\alpha M$.

(τ) If $M_1 \sim_\alpha M_2$ and $M_2 \sim_\alpha M_3$ then $M_1 \sim_\alpha M_3$.

(σ) If $M_1 \sim_\alpha M_2$ then $M_2 \sim_\alpha M_1$.

(α) If $[x/y]M \sim_\alpha [x/z]N$ then $\lambda y.M \sim_\alpha \lambda z.N$.

In the following argument, if M is alpha-equivalent to N, we identify those two lambda-terms and write $M \equiv N$.

Then, we introduce a binary relation $\rhd_{1\beta}$ on Tm_λ which captures the work of the lambda-abstraction.

Definition 1.4 (beta-contraction, beta-reduction, beta-equivalent)**.** If N is obtained from M by replacing a subterm of the form $(\lambda x.P)Q$ by the term $[Q/x]P$, we write $M \rhd_{1\beta} N$. Strictly speaking, we write $M \rhd_{1\beta} N$ if $M \hookrightarrow_1 N$ can be derived by the following rules:

(β) $(\lambda x.M)N \hookrightarrow_1 [N/x]M$.

(ξ) If $M \hookrightarrow_1 N$ then $\lambda x.M \hookrightarrow_1 \lambda x.N$.

(σ) If $M \hookrightarrow_1 N$ then $PM \hookrightarrow_1 PN$ and $MQ \hookrightarrow_1 NQ$.

A term of the form $(\lambda x.M)N$ is called a *beta-redex* and the corresponding term $[N/x]M$ is called its *contractum.* $\rhd_{1\beta}$ is called the *beta-contraction* relation. We also define the binary relation $\rhd_\beta$ (*beta-reduction*) as the reflexive transitive closure of $\rhd_{1\beta}$, and define the binary relation $=_\beta$ (beta-equivalence) as the smallest equivalent relation including $\rhd_{1\beta}$.

[4] Recall lambda-variables are enumerable and each variable is assigned a natural number as its index: $\mathtt{x}_1, \mathtt{x}_2 \ldots$. The first variable in $\mathrm{V}_\lambda \setminus \mathrm{FV}_\lambda(P)$ means the lambda-variable with the smallest index in $\mathrm{V}_\lambda \setminus \mathrm{FV}_\lambda(P)$.

[5] In this thesis, we will follow this promise when we define new notions inductively.

$M \in \mathrm{Tm}_\lambda$ is called a *beta-normal form* if $M \not\triangleright_{1\beta} N$ for every $N \in \mathrm{Tm}_\lambda$. We say N is lambda-normal form of M if $M \triangleright_\beta N$ and N is a lambda-normal form.

Example 1.5.

$$(\lambda xy.x(xy))FN \triangleright_{1\beta} (\lambda y.F(Fy))N \triangleright_{1\beta} F(FN).$$

The following properties can be easily checked.

Theorem 1.6.

1. If $M \triangleright_\beta N$ then, $PM \triangleright_\beta PN$, $MQ \triangleright_\beta NQ$ and $\lambda x.M \triangleright_\beta \lambda x.N$ for each P, Q, x.

2. If $M \triangleright_\beta N$ and $P \triangleright_\beta Q$ then $[P/x]M \triangleright_\beta [Q/x]N$ for each x.

3. If $M \triangleright_\beta N$ then $\mathrm{FV}_\lambda(M) \supseteq \mathrm{FV}_\lambda(N)$.

Proof. See (Hindley & Seldin, 2008). □

We call the system (structure) $\langle \mathrm{Tm}_\lambda, \triangleright_{1\beta}\rangle$ the lambda-beta-calculus. The lambda-beta-calculus has very simple structure, but has very strong expressiveness. Church and Kleene proposed a computation model based on the lambda-beta-calculus, and showed that every recursive functions can be defined in the computation model:

Definition 1.7 (Church-numeral, lambda-definable function)**.** For each $n \in \mathcal{N}$, we give the lambda-term C_n as follows:

$$C_0 \equiv y, \quad C_{n+1} \equiv xC_n.$$

Then, we define the *Church numeral* $\overline{n}$ of n as $\lambda xy.C_n$. Note that each Church-numeral is a beta-normal form.

We say a k-ary partial function f on $\mathcal{N}$ is *lambda-definable* if there exists a lambda-term F which satisfies the following conditions for each $n_1, \ldots, n_k \in \mathcal{N}$:

- $F\overline{n_1}\ldots\overline{n_k} \triangleright_\beta \overline{f(n_1, \ldots, n_k)}$ if $f(n_1, \ldots, n_k)$ is defined.

- $F\overline{n_1}\ldots\overline{n_k}$ has no beta-normal form if $f(n_1, \ldots, n_k)$ is undefined.

Theorem 1.8. Each recursive function is lambda-definable, and each lambda-definable function is recursive.

Proof. See (Hindley & Seldin, 2008; Kleene, 1936). □

Note that each calculation step in lambda-beta-calculus is not unique, that is, there may be plural lambda-terms obtained from M by one-step beta-contraction. For example, we can obtain both $(\lambda x.xx)((\lambda y.y)z) \triangleright_{1\beta} ((\lambda y.y)z)((\lambda y.y)z)$ and $(\lambda x.xx)((\lambda y.y)z) \triangleright_{1\beta} (\lambda x.xx)z$. Then, the following question arises:

Is there a situation such that $F\overline{n_1}\ldots\overline{n_k} =_\tau \overline{m_1}$ and $F\overline{n_1}\ldots\overline{n_k} =_\tau \overline{m_2}$ for some $m_1 \neq m_2$.

If such a situation exists, the notion of lambda-definable function should be regarded to be worthless. The following theorem guarantees that such a situation cannot happen.

Theorem 1.9 (Church-Rosser theorem)**.** If $M \triangleright_\beta N_1$ and $M \triangleright_\beta N_2$ then there exists R such that $N_1 \triangleright_\beta R$ and $N_2 \triangleright_\beta R$.

$$\begin{tikzcd}
 & N_1 \arrow[dr, phantom, "\triangleright_\beta"] & \\
M \arrow[ur, phantom, "\triangleright_\beta"] \arrow[dr, phantom, "\triangleright_\beta"] & & \exists R \\
 & N_2 \arrow[ur, phantom, "\triangleright_\beta"] &
\end{tikzcd}$$

A proof of the Church-Rosser theorem is given in chapter 2.

1.1.2 Typed lambda-beta-calculus

In the above system, we can construct an ill-behaved term. For example, $(\lambda y.z)((\lambda x.xx)(\lambda x.xx))$ has a beta-normal form, but it can causes an infinite $\triangleright_{1\beta}$-sequence

$$(\lambda y.z)((\lambda x.xx)(\lambda x.xx)) \triangleright_{1\beta} (\lambda y.z)((\lambda x.xx)(\lambda x.xx)) \triangleright_{1\beta} (\lambda y.z)((\lambda x.xx)(\lambda x.xx)) \triangleright_{1\beta} \ldots$$

by contracting the beta-redex $(\lambda x.xx)(\lambda x.xx)$ repeatedly. Here, we will explain how we can remove those ill-behaved terms by treating only terms called typed lambda-terms.

Suppose a countable set $\mathrm{AT} = \{\mathtt{t}_1, \mathtt{t}_2, \ldots\}$ of *atomic types* is given. Then the set $\mathrm{Tp}_\rightarrow$ of *simple types* is defined as follows:

$$\begin{aligned}
&t \in \mathrm{AT} \\
&\sigma, \tau \in \mathrm{Tp}_\rightarrow ::= t \mid (\sigma \rightarrow \tau)
\end{aligned}$$

We use metavariables $s, t, u, \ldots$ for atomic types, and use metavariables $\sigma, \tau, \theta, \ldots$ for types.

Suppose, for each $\sigma \in \mathrm{Tp}_\rightarrow$, a countable set $\mathrm{V}^\sigma_\lambda = \{\mathtt{x}^\sigma_1, \mathtt{x}^\sigma_2, \ldots\}$ of typed lambda-variables is given. Then the set TpTm_λ of *typed lambda-terms* and a mapping $\mathrm{Type} : \mathrm{TpTm}_\lambda \rightarrow \mathrm{Tp}_\rightarrow$ are defined as follows:

1. If $x \in \mathrm{V}^\sigma_\lambda$ then $x \in \mathrm{TpTm}_\lambda$ and $\mathrm{Type}(x) = \sigma$.

2. If $M, N \in \mathrm{TpTm}_\lambda$, $\mathrm{Type}(M) = \sigma \rightarrow \tau$ and $\mathrm{Type}(N) = \sigma$ then $MN \in \mathrm{TpTm}_\lambda$ and $\mathrm{Type}(MN) = \tau$.

3. If $M \in \mathrm{TpTm}_\lambda$, $\mathrm{Type}(M) = \sigma$ and $x \in \mathrm{V}_\lambda^\tau$ then $\lambda x.M \in \mathrm{TpTm}_\lambda$ and $\mathrm{Type}(\lambda x.M) = \tau \to \sigma$.

Theorem 1.10 (Subject reduction theorem)**.** Let $M \in \mathrm{TpTm}_\lambda$. If $M \rhd_{1\beta} N$ then $N \in \mathrm{TpTm}_\lambda$ and $\mathrm{Type}(N) = \mathrm{Type}(M)$.

Proof. See (Hindley & Seldin, 2008). □

The following theorem shows that the ill-behaved terms is actually removed.

Theorem 1.11 (Strong normalization theorem)**.** If $M \in \mathrm{TpTm}_\lambda$, then there are no infinite sequences of beta-contraction starting from M.

Proof. See (Hindley & Seldin, 2008). □

1.2 Proof theory

When we write a mathematical proof, we naturally construct a new complicated proposition from some basic propositions by use of some logical connectives such as “or”, “and ”, “not” and “if ... then”. *Symbolic propositional logic* and *proof theory* study the work of such logical connectives by representing mathematical propositions and mathematical proofs as formal objects and analyzing them.

The language $\mathcal{L}$, which we use in this thesis, consists of the following symbols:

- **Propositional variables**: We prepare countably many *propositional variables* $\mathtt{p}_1, \mathtt{p}_2, \ldots$. Each propositional variable represents an arbitrary proposition. We write a set of every propositional variables as PV. We use metavariables $p, q, r, \ldots$ to represent propositional variables.

- **Logical symbols**: In this thesis, we treat only $\supset$ and $\perp$ as logical symbols [6] . $\alpha \supset \beta$ represents the proposition “if α then β”, and $\perp$ represents a contradicted proposition.

The set $\mathrm{Fml}_{\supset\perp}$ of formulas and the set $\mathrm{Fml}_{\supset}$ of implicational formulas are defined as follows:

1. Each propositional formula is in $\mathrm{Fml}_{\supset\perp}$ and $\mathrm{Fml}_{\supset}$.

2. $\perp$ is in $\mathrm{Fml}_{\supset\perp}$.

3. If α, β are both in $\mathrm{Fml}_{\supset\perp}$ (resp. $\mathrm{Fml}_{\supset}$) then $(\alpha \supset \beta)$ is in $\mathrm{Fml}_{\supset\perp}$ (resp. $\mathrm{Fml}_{\supset}$).

[6] In this thesis, we do not treat other logical connectives such as “and” and “or”. Proof theoretic treatment of such connectives is written in (Buss, 1998) for example.

We sometimes use the notation Fml to represent either $\mathrm{Fml}_{\supset}$ or $\mathrm{Fml}_{\supset\perp}$. We use metavariables $\alpha, \beta, \gamma, \ldots$ to represent formulas. $(\alpha \supset (\beta \supset \gamma))$ is abbreviated as $\alpha \supset \beta \supset \gamma$ and $\alpha \supset \perp$ is abbreviated as $\neg\alpha$. We define the formula $[\beta_1/p_1, \ldots, \beta_n/p_n]\alpha$ as the formula obtained from α by replacing each p_i by β_i simultaneously. A formula of the form $[\beta_1/p_1, \ldots, \beta_n/p_n]\alpha$ is called a substitution instance of α.

1.2.1 Classical logic and intuitionistic logic

Classical logic is a logic which is admitted and used by almost all mathematicians. There are many proof systems which formalize classical logic. One of the most famous system was given by Hilbert:

Definition 1.12. Hilbert style proof system **HK** for classical logic has the axioms (S), (K), (A), (P) and the inference rule (E $\supset$) [7] .

$$
\begin{array}{ll}
\text{(S)} & (p \supset q \supset r) \supset (p \supset q) \supset p \supset r \\
\text{(K)} & p \supset q \supset p \\
\text{(A)} & \perp \supset p \\
\text{(P)} & ((p \supset q) \supset p) \supset p
\end{array}
$$

$$
\frac{\alpha \supset \beta \quad \alpha}{\beta}\ (\mathrm{E} \supset)
$$

With these rules, we define the notion of **HK**-proof. Each **HK**-proof has its assumption set $\Gamma \subseteq \mathcal{N} \times \mathrm{Fml}$ and its conclusion $\alpha \in \mathrm{Fml}$. We write a pair $\langle n, \alpha \rangle \in \mathcal{N} \times \mathrm{Fml}$ as $n : \alpha$. **HK**-proofs are constructed as follows:

1. For each $n \in \mathcal{N}$ and $\alpha \in \mathrm{Fml}$,

$$n : \alpha$$

   is an **HK**-proof of α with the assumption set $\{n : \alpha\}$.

2. Let $(*) \in \{(\mathrm{S}), (\mathrm{K}), (\mathrm{A}), (\mathrm{P})\}$. If α is a substitution instance of the $(*)$-axiom, then

$$\frac{}{\alpha}\ (*)$$

   is an **HK**-proof of α with no assumption.

[7] The rule (E $\supset$) is sometimes called the *modus ponens*.

3. Let Π_1 be an **HK**-proof of $\alpha \supset \beta$ with the assumption set Γ and Π_2 be an **HK**-proof of α with the assumption set Δ. Then

$$\frac{\Pi_1 \quad \Pi_2}{\beta}\ (\mathrm{E}\supset)$$

is an **HK**-proof of β with the assumption set $\Gamma \cup \Delta$.

We write $\Gamma \vdash_{\mathbf{HK}} \alpha$ for $\Gamma \subseteq \mathrm{Fml}$ and $\alpha \in \mathrm{Fml}$, if there exists $\Gamma^+ \subseteq \mathcal{N} \times \mathrm{Fml}$ such that $\Gamma = \{\alpha \mid n : \alpha \in \Gamma^+\}$ and there exists an **HK**-proof of α with the assumption set Γ^+. We say α is provable in **HK** and write $\vdash_{\mathbf{HK}} \alpha$ if $\emptyset \vdash_{\mathbf{HK}} \alpha$.

The proof system $\mathbf{HK}_{\supset}$ for implicational classical logic consists of the axiom scheme $(\mathrm{S}), (\mathrm{K}), (\mathrm{P})$ and the inference rule $(\mathrm{E}\supset)$.

Intuitionistic logic is a logic, introduced by Brouwer, which admits only constructive reasoning. Heyting, a disciple of Brouwer, gave a Hilbert style proof system **HJ** for intuitionistic logic:

Definition 1.13. Hilbert style proof system **HJ** for intuitionistic logic consists of the axiom schemes $(\mathrm{S}), (\mathrm{K}), (\mathrm{A})$ and the inference rule $(\mathrm{E}\supset)$. The notion of **HJ**-proof and $\vdash_{\mathbf{HJ}}$ are defined in the same way as **HK**-proof and $\vdash_{\mathbf{HK}}$ respectively.

The proof system $\mathbf{HJ}_{\supset}$ for implicational intuitionistic logic consists of the axiom schemes $(\mathrm{S}), (\mathrm{K})$ and the inference rule $(\mathrm{E}\supset)$.

Example 1.14.
(1) From the following proof, we obtain $\{\neg\alpha, \alpha\} \vdash_{\mathbf{HJ}} \beta$.

$$\frac{\dfrac{}{\bot \supset \beta}\ (\mathrm{A}) \qquad \dfrac{1 : \neg\alpha \quad 2 : \alpha}{\bot}\ (\mathrm{E}\supset)}{\beta}\ (\mathrm{E}\supset)$$

(2) From the following proof, we obtain $\vdash_{\mathbf{HJ}} \alpha \supset \alpha$.

$$\frac{\dfrac{\dfrac{}{(\alpha \supset (\alpha \supset \alpha) \supset \alpha) \supset (\alpha \supset \alpha \supset \alpha) \supset \alpha \supset \alpha}\ (\mathrm{S}) \quad \dfrac{}{\alpha \supset (\alpha \supset \alpha) \supset \alpha}\ (\mathrm{K})}{(\alpha \supset \alpha \supset \alpha) \supset \alpha \supset \alpha}\ (\mathrm{E}\supset) \qquad \dfrac{}{\alpha \supset \alpha \supset \alpha}\ (\mathrm{K})}{\alpha \supset \alpha}\ (\mathrm{E}\supset)$$

As you can see from the above examples, Hilbert style proof has very different form from real mathematical proof. Gentzen (Gentzen, 1935) analysed many mathematical proofs and introduced the proof systems **NK** for classical logic and **NJ** for intuitionistic logic.

Definition 1.15 (Natural deduction style proof systems **NK** and **NJ**)**.** Each **NK**-proof has its assumption set $\Gamma \subseteq \mathcal{N} \times \mathrm{Fml}$ and its conclusion $\alpha \in \mathrm{Fml}$.

NK-proofs are constructed as follows:

1. For each $n \in \mathcal{N}$ and $\alpha \in \mathrm{Fml}$,

$$n : \alpha$$

is an **NK**-proof of α with the assumption set $\{n : \alpha\}$.

2. Let Π_1 be an **NK**-proof of $\alpha \supset \beta$ with the assumption set Γ and Π_2 be an **NK**-proof of α with the assumption set Δ. Then

$$\frac{\Pi_1 \quad \Pi_2}{\beta}\,(\mathrm{E} \supset)$$

is an **NK**-proof of β with the assumption set $\Gamma \cup \Delta$.

3. Let Π be an **NK**-proof of β with the assumption set Γ. Then

$$\frac{\Pi}{\alpha \supset \beta}\,(\mathrm{I} \supset \;\; n : \alpha)$$

is an **NK**-proof of $\alpha \supset \beta$ with the assumption set $\Gamma \setminus \{n : \alpha\}$.

4. Let Π be an **NK**-proof of $\perp$ with the assumption set Γ, then

$$\frac{\Pi}{\alpha}\,(\mathrm{Absurd})$$

is an **NK**-proof of α with the assumption set Γ.

5. Let Π be an **NK**-proof of $\neg\neg\alpha$ with the assumption set Γ, then

$$\frac{\Pi}{\alpha}\,(\mathrm{DNE})$$

is an **NK**-proof of α with the assumption set Γ.

We write $\Gamma \vdash_{\mathbf{NK}} \alpha$ for $\Gamma \subseteq \mathrm{Fml}$ and $\alpha \in \mathrm{Fml}$, if there exists $\Gamma^+ \subseteq \mathcal{N} \times \mathrm{Fml}$ such that $\Gamma = \{\alpha \mid n : \alpha \in \Gamma^+\}$ and there exists an **NK**-proof of α with the assumption set Γ^+. We say α is provable in **NK** and write $\vdash_{\mathbf{NK}} \alpha$ if $\emptyset \vdash_{\mathbf{NK}} \alpha$.

The proof system **NJ** for intuitionistic logic consists of the inference rules $(\mathrm{E} \supset), (\mathrm{I} \supset), (\mathrm{Absurd})$, that is, **NJ** is obtained from **NK** by removing the inference rule (DNE). We define the notation $\vdash_{\mathbf{NJ}}$ in the same way as $\vdash_{\mathbf{NK}}$. The proof system $\mathbf{NJ}_{\supset}$ for implicational intuitionistic logic consists of the inference rules $(\mathrm{E} \supset), (\mathrm{I} \supset)$.

Example 1.16.

1. We have $\{\alpha, \neg\alpha\} \vdash_{\mathbf{NK}} \beta$ because we can construct the following **NJ**-proof.

$$\dfrac{\dfrac{1 : \neg\alpha \quad 2 : \alpha}{\perp}\,(\mathrm{E} \supset)}{\beta}\,(\mathrm{Absurd})$$

2. We have $\vdash_{\mathbf{NJ}} \alpha \supset \alpha$ because we can construct the following **NJ**-proof.
$$\frac{1 : \alpha}{\alpha \supset \alpha}\ (\mathrm{I}\supset\ \ 1 : \alpha)$$

Note 1.17. Consider the following two **NJ**-proofs:

$$\frac{\dfrac{\dfrac{1 : \alpha \supset \beta \quad 2 : \alpha}{\beta}\ (\mathrm{E}\supset)}{(\alpha \supset \beta) \supset \beta}\ (\mathrm{I}\supset\ \ 1 : \alpha \supset \beta)}{\alpha \supset (\alpha \supset \beta) \supset \beta}\ (\mathrm{I}\supset\ \ 2 : \alpha) \qquad \frac{\dfrac{\dfrac{3 : \alpha \supset \beta \quad 4 : \alpha}{\beta}\ (\mathrm{E}\supset)}{(\alpha \supset \beta) \supset \beta}\ (\mathrm{I}\supset\ \ 3 : \alpha \supset \beta)}{\alpha \supset (\alpha \supset \beta) \supset \beta}\ (\mathrm{I}\supset\ \ 4 : \alpha)$$

One may notice that these proofs are essentially the same. In the following argument, we identify such proofs. Strictly speaking, we identify a proof Π to a proof Σ if Σ is obtained from Π by replacing labels of some discharged assumptions.

1.2.2 Proof contraction for $\mathbf{NJ}_{\supset}$

As written above, in proof theory, we studies properties of a logic by observing a formal proof system of the logic. One of the most useful tool in proof theory is proof contraction (proof transformation). For example, Prawitz (Prawitz, 1965) studies some proof contractions for some natural deduction style proof systems, and showed some important properties of those systems. In this subsection, we introduce a proof contraction called $\supset$-*contraction* for $\mathbf{NJ}_{\supset}$.

Definition 1.18. For each $\mathbf{NJ}_{\supset}$-proof Σ of α, we define the proof $[\Sigma \ /\ n : \alpha]\Pi$ as the $\mathbf{NJ}_{\supset}$-proof obtained from Π by replacing each assumption $n : \alpha$ by Σ. Here we assume that if m is used as a label of an assumption of Σ then there are no discharged assumptions with label m in Π.

Let Π be an $\mathbf{NJ}_{\supset}$-proof. If there exists a subproof of Π of the form

$$\frac{\dfrac{\Sigma}{\alpha \supset \beta}\ (\mathrm{I}\supset\ \ n : \alpha) \qquad \Omega}{\beta}\ (\mathrm{E}\supset)\ ,$$

where Σ is an $\mathbf{NJ}_{\supset}$-proof of β and Ω is an $\mathbf{NJ}_{\supset}$-proof of α, then we call the subproof a *detour* in Π. If Π' be obtained from Π by replacing a detour of the above form by the proof $[\Omega \ /\ n : \alpha]\Sigma$, then we write $\Pi \triangleright_{1\rightarrow} \Pi'$. We also define the relation $\triangleright_{\supset}$ as the reflexive transitive closure of $\triangleright_{1p}$. An $\mathbf{NJ}_{\supset}$-proof is said to be *normal* if it includes no detours.

Example 1.19.

$$\frac{\dfrac{\dfrac{\dfrac{1 : (\alpha \supset \alpha) \supset (\alpha \supset \alpha) \supset \beta \quad 2 : \alpha \supset \alpha}{(\alpha \supset \alpha) \supset \beta}\ (\mathrm{E}\supset) \qquad 2 : \alpha \supset \alpha}{\beta}\ (\mathrm{E}\supset)}{(\alpha \supset \alpha) \supset \beta}\ (\mathrm{I}\supset\ \ 2 : \alpha) \qquad \dfrac{3 : \alpha}{\alpha \supset \alpha}\ (\mathrm{I}\supset\ \ 3 : \alpha)}{\beta}\ (\mathrm{E}\supset)$$

$$\nabla_{1\supset}$$

$$\dfrac{\dfrac{1:(\alpha\supset\alpha)\supset(\alpha\supset\alpha)\supset\beta \quad \dfrac{3:\alpha}{\alpha\supset\alpha}\,(\mathrm{I}\supset\ 3:\alpha)}{(\alpha\supset\alpha)\supset\beta}\,(\mathrm{E}\supset) \qquad \dfrac{3:\alpha}{\alpha\supset\alpha}\,(\mathrm{I}\supset\ 3:\alpha)}{\beta}\,(\mathrm{E}\supset)$$

Theorem 1.20. If Σ is an **NJ**-proof and $\Sigma \rhd_{1\to} \Pi$ then Π is an **NJ**-proof.

Proof. See (Komori & Ono, 2010). □

Theorem 1.21. For each $\mathbf{NJ}_{\supset}$-proof Π, there exists a normal $\mathbf{NJ}_{\supset}$-proof Σ such that $\Pi \rhd_{\supset} \Sigma$.

Proof. See (Komori & Ono, 2010). □

From the above theorems, we can obtain an important property of intuitionistic logic:

Corollary 1.22. Intuitionistic logic is consistent, that is, there exists a formula α such that $\nvdash_{\mathbf{NJ}} \alpha$.

Proof. We can show $\nvdash_{\mathbf{NJ}_{\supset}} p$ as follows: If $\vdash_{\mathbf{NJ}_{\supset}} p$, then there exists a normal $\mathbf{NJ}_{\supset}$-proof of p with no assumption. However we cannon construct such a proof. □

1.3 The Curry-Howard correspondence

We introduced the typed lambda-beta-calculus in subsection 1.1.2 and the proof system $\mathbf{NJ}_{\supset}$ in subsection 1.2.1. Each system has its own history and philosophy, but there are many similarities between these systems. The correspondence between these system was discovered by Howard (Howard, 1980) and, since then, have been studied in many fields such as mathematics, computer science and philosophy. The correspondence and the correspondence between $\mathbf{HJ}_{\supset}$ and the combinatory logic **SK** discovered by Curry (Curry, Feys, Craig, & Craig, 1958) are called the *Curry-Howard correspondence.* In subsection 1.3.1, we give an explanation on the correspondence between the lambda-beta-calculus and $\mathbf{NJ}_{\supset}$.

In the following argument, we use the following notation: We define Pr as the set of $\mathbf{NJ}_{\supset}$-proof. Furthermore, we write the assumption of $\Pi \in \mathrm{Pr}$ as $\mathrm{Ass}(\Pi) \subseteq (\mathcal{N} \times \mathrm{Fml})$, and write the conclusion of Π as $\mathrm{Con}(\Pi) \in \mathrm{Fml}$.

1.3.1 Typed lambda-beta-calculus and $\mathbf{NJ}_{\supset}$

Because both AT (the set of atomic types) and PV (the set of propositional variables) are enumerable, we can give a bijection How_0 from AT into PV as $\mathrm{How}_0(\mathtt{t}_i) = \mathtt{p}_i$. Based on this bijection, we will observe the correspondence between the lambda-beta-calculus and $\mathbf{NJ}_{\supset}$.

We first define a mapping How_1 from $\mathrm{Tp}_{\rightarrow}$ (the set of simple types) into $\mathrm{Fml}_{\supset}$ (the set of implicational formulas) as

$$\mathrm{How}_1(\mathtt{t}_i) = \mathrm{How}_0(\mathtt{t}_i)(= \mathtt{p}_i), \quad \mathrm{How}_1(\tau \rightarrow \sigma) = \mathrm{How}_1(\tau) \supset \mathrm{How}_1(\sigma).$$

Then, define a mapping $\mathrm{How}_2 : \bigcup_{\tau \in \mathrm{Tp}_{\rightarrow}} \mathrm{V}^{\tau}_{\lambda} \rightarrow (\mathcal{N} \times \mathrm{Fml})$ as

$$\mathrm{How}_2(\mathtt{x}^{\tau}_n) = n : \mathrm{How}_1(\tau).$$

Next, we define a mapping $\mathrm{How}_3 : \mathrm{TpTm}_{\lambda} \rightarrow \mathrm{Pr}$, which satisfies $\mathrm{Con}(\mathrm{How}_3(M)) = \mathrm{How}_1(\mathrm{Type}(M))$ and $\mathrm{Ass}(\mathrm{How}_3(M)) = \mathrm{How}_2(\mathrm{FV}_{\lambda}(M))$, as follows:

1. $\mathrm{How}_3(\mathtt{x}^{\tau}_n)$ is the following $\mathbf{NJ}_{\supset}$-proof.

$$n : \mathrm{How}_1(\tau)$$

2. Let $M \equiv PQ$. In this case, we have $\mathrm{Type}(P) = \tau \rightarrow \sigma$, $\mathrm{Type}(Q) = \tau$ and $\mathrm{Type}(M) = \sigma$ for some τ, σ. By induction hypothesis, we obtain an $\mathbf{NJ}_{\supset}$-proof $\mathrm{How}_3(P)$ of $\mathrm{How}_1(\tau) \supset \mathrm{How}_1(\sigma)$ and an $\mathbf{NJ}_{\supset}$-proof $\mathrm{How}_1(Q)$ of $\mathrm{How}_1(\tau)$. Then we define $\mathrm{How}_3(M)$ as the following proof.

$$\frac{\mathrm{How}_3(P) \quad \mathrm{How}_3(Q)}{\mathrm{How}_1(\sigma)} \ (\mathrm{E} \supset)$$

3. Let $M \equiv \lambda \mathtt{x}^{\tau}_n.N$. In this case, we have $\mathrm{Type}(N) = \sigma$ and $\mathrm{Type}(M) = \tau \rightarrow \sigma$ for some σ. By induction hypothesis, we obtain an $\mathbf{NJ}_{\supset}$-proof $\mathrm{How}_3(N)$ of $\mathrm{How}_1(\sigma)$. Then we define $\mathrm{How}_3(M)$ as the following proof.

$$\frac{\mathrm{How}_3(N)}{\mathrm{How}_1(\tau) \supset \mathrm{How}_1(\sigma)} \ (\mathrm{I} \supset \ n : \mathrm{How}_1(\tau))$$

Finally let $\mathrm{How} = \mathrm{How}_1 \oplus \mathrm{How}_2 \oplus \mathrm{How}_3$. Then How tells us that there is a close connection between the lambda-beta-calculus and $\mathbf{NJ}_{\supset}$ in the following sense:

Theorem 1.23. How is an isomorphism from the structure $\langle \mathrm{Tp}_{\supset} \oplus \bigcup_{\tau \in \mathrm{Tp}_{\rightarrow}} \mathrm{V}^{\tau}_{\lambda} \oplus \mathrm{TpTm}_{\lambda} : \mathrm{Type}, \mathrm{FV}_{\lambda} : \rhd_{1\beta} \rangle$ into the structure $\langle \mathrm{Fml}_{\supset} \oplus (\mathcal{N} \times \mathrm{Fml}_{\supset}) \oplus \mathrm{Pr} : \mathrm{Con}, \mathrm{Ass} : \rhd_{1p} \rangle$ [8] , i.e. the following properties hold for each $M, N \in \mathrm{TpTm}_{\lambda}$:

[8] In this thesis, $\langle S : f_1, \ldots, f_n : r_1, \ldots, r_m \rangle$ means a many sorted structure where S is the base set of this structure and each f_i is a function and each r_j is a relation.

1. How_1 is a bijection from $\mathrm{Tp}_{\to}$ into $\mathrm{Fml}_{\supset}$.

$$(\mathtt{t}_1 \to \mathtt{t}_1) \to (\mathtt{t}_1 \to \mathtt{t}_1) \to \mathtt{t}_2$$

$$\Big\updownarrow \ \text{How}$$

$$(\mathrm{p}_1 \supset \mathrm{p}_1) \supset (\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2$$

2. How_2 is a bijection from $\bigcup_{\tau \in \mathrm{Tp}_{\to}} \mathrm{V}^{\tau}_{\lambda}$ into $(\mathcal{N} \times \mathrm{Fml}_{\supset})$.

$$\mathrm{x}_1^{(\mathtt{t}_1 \to \mathtt{t}_1) \to (\mathtt{t}_1 \to \mathtt{t}_1) \to \mathtt{t}_2}$$

$$\Big\updownarrow \ \text{How}$$

$$1 : (\mathrm{p}_1 \supset \mathrm{p}_1) \supset (\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2$$

3. How_3 is a bijection from TpTm_{λ} into Pr.

$$(\lambda \mathrm{x}_2^{\mathtt{t}_1 \to \mathtt{t}_1}.\ \mathrm{x}_1^{(\mathtt{t}_1 \to \mathtt{t}_1) \to (\mathtt{t}_1 \to \mathtt{t}_1) \to \mathtt{t}_2} \mathrm{x}_2^{\mathtt{t}_1 \to \mathtt{t}_1} \mathrm{x}_2^{\mathtt{t}_1 \to \mathtt{t}_1})(\lambda \mathrm{x}_3^{\mathtt{t}_1}.\ \mathrm{x}_3^{\mathtt{t}_1})$$

$$\Big\updownarrow \ \text{How}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{1 : (\mathrm{p}_1 \supset \mathrm{p}_1) \supset (\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2 \qquad 2 : \mathrm{p}_1 \supset \mathrm{p}_1}{(\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2}\,(\mathrm{E} \supset) \qquad 2 : \mathrm{p}_1 \supset \mathrm{p}_1}{\mathrm{p}_2}\,(\mathrm{E} \supset)}{(\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2}\,(\mathrm{I} \supset\ 2 : \mathrm{p}_1) \qquad \dfrac{3 : \mathrm{p}_1}{\mathrm{p}_1 \supset \mathrm{p}_1}\,(\mathrm{I} \supset\ 3 : \mathrm{p}_1)}{\mathrm{p}_2}\,(\mathrm{E} \supset)$$

4. $\mathrm{How}_1(\mathrm{Type}(M)) = \mathrm{Con}(\mathrm{How}_3(M))$.

$$\mathrm{Type}\ \Big((\lambda \mathrm{x}_2^{\mathtt{t}_1 \to \mathtt{t}_1}.\ \mathrm{x}_1^{(\mathtt{t}_1 \to \mathtt{t}_1) \to (\mathtt{t}_1 \to \mathtt{t}_1) \to \mathtt{t}_2} \mathrm{x}_2^{\mathtt{t}_1 \to \mathtt{t}_1} \mathrm{x}_2^{\mathtt{t}_1 \to \mathtt{t}_1})(\lambda \mathrm{x}_3^{\mathtt{t}_1}.\ \mathrm{x}_3^{\mathtt{t}_1}) \Big) \ = \ \mathtt{t}_2$$

$$\Big\updownarrow \ \text{How}$$

$$\mathrm{Con}\ \left(\dfrac{\dfrac{\dfrac{\dfrac{1 : (\mathrm{p}_1 \supset \mathrm{p}_1) \supset (\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2 \qquad 2 : \mathrm{p}_1 \supset \mathrm{p}_1}{(\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2}\,(\mathrm{E} \supset) \qquad 2 : \mathrm{p}_1 \supset \mathrm{p}_1}{\mathrm{p}_2}\,(\mathrm{E} \supset)}{(\mathrm{p}_1 \supset \mathrm{p}_1) \supset \mathrm{p}_2}\,(\mathrm{I} \supset\ 2 : \mathrm{p}_1) \qquad \dfrac{3 : \mathrm{p}_1}{\mathrm{p}_1 \supset \mathrm{p}_1}\,(\mathrm{I} \supset\ 3 : \mathrm{p}_1)}{\mathrm{p}_2}\,(\mathrm{E} \supset) \right)$$

$$= \ \mathrm{p}_2$$

5. $\mathrm{How}_2(\mathrm{FV}_\lambda(M)) = \mathrm{Ass}(\mathrm{How}_3(M))$.

$$\mathrm{FV}_\lambda\left((\lambda \mathrm{x}_2^{\mathrm{t}_1\to \mathrm{t}_1}.\, \mathrm{x}_1^{(\mathrm{t}_1\to \mathrm{t}_1)\to(\mathrm{t}_1\to \mathrm{t}_1)\to \mathrm{t}_2} \mathrm{x}_2^{\mathrm{t}_1\to \mathrm{t}_1} \mathrm{x}_2^{\mathrm{t}_1\to \mathrm{t}_1})(\lambda \mathrm{x}_3^{\mathrm{t}_1}.\, \mathrm{x}_3^{\mathrm{t}_1})\right) = \left\{\mathrm{x}_1^{(\mathrm{t}_1\to \mathrm{t}_1)\to(\mathrm{t}_1\to \mathrm{t}_1)\to \mathrm{t}_2}\right\}$$

$$\Updownarrow\ \text{How}$$

$$\mathrm{Ass}\left(\dfrac{\dfrac{\dfrac{\dfrac{1:(\mathrm{p}_1\supset \mathrm{p}_1)\supset(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2 \quad 2:\mathrm{p}_1\supset \mathrm{p}_1}{(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2}\,(\mathrm{E}\supset) \quad 2:\mathrm{p}_1\supset \mathrm{p}_1}{\mathrm{p}_2}\,(\mathrm{E}\supset)}{(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2}\,(\mathrm{I}\supset\ 2:\mathrm{p}_1) \quad \dfrac{3:\mathrm{p}_1}{\mathrm{p}_1\supset \mathrm{p}_1}\,(\mathrm{I}\supset\ 3:\mathrm{p}_1)}{\mathrm{p}_2}\,(\mathrm{E}\supset)\right)$$

$$= \left\{\ 1:(\mathrm{p}_1\supset \mathrm{p}_1)\supset(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2\ \right\}$$

6. If $M \rhd_{1\beta} N$ then $\mathrm{How}_3(M) \rhd_{1p} \mathrm{How}_3(N)$.

$$(\lambda \mathrm{x}_2^{\mathrm{t}_1\to \mathrm{t}_1}.\, \mathrm{x}_1^{(\mathrm{t}_1\to \mathrm{t}_1)\to(\mathrm{t}_1\to \mathrm{t}_1)\to \mathrm{t}_2} \mathrm{x}_2^{\mathrm{t}_1\to \mathrm{t}_1} \mathrm{x}_2^{\mathrm{t}_1\to \mathrm{t}_1})(\lambda \mathrm{x}_3^{\mathrm{t}_1}.\, \mathrm{x}_3^{\mathrm{t}_1})$$

$$\rhd_{1\beta} \qquad \mathrm{x}_1^{(\mathrm{t}_1\to \mathrm{t}_1)\to(\mathrm{t}_1\to \mathrm{t}_1)\to \mathrm{t}_2}(\lambda \mathrm{x}_3^{\mathrm{t}_1}.\, \mathrm{x}_3^{\mathrm{t}_1})(\lambda \mathrm{x}_3^{\mathrm{t}_1}.\, \mathrm{x}_3^{\mathrm{t}_1})$$

$$\Updownarrow\ \text{How}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{1:(\mathrm{p}_1\supset \mathrm{p}_1)\supset(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2 \quad 2:\mathrm{p}_1\supset \mathrm{p}_1}{(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2}\,(\mathrm{E}\supset) \quad 2:\mathrm{p}_1\supset \mathrm{p}_1}{\mathrm{p}_2}\,(\mathrm{E}\supset)}{(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2}\,(\mathrm{I}\supset\ 2:\mathrm{p}_1) \quad \dfrac{3:\mathrm{p}_1}{\mathrm{p}_1\supset \mathrm{p}_1}\,(\mathrm{I}\supset\ 3:\mathrm{p}_1)}{\mathrm{p}_2}\,(\mathrm{E}\supset)$$

$$\rhd_{1\supset} \qquad \dfrac{\dfrac{1:(\mathrm{p}_1\supset \mathrm{p}_1)\supset(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2 \quad \dfrac{3:\mathrm{p}_1}{\mathrm{p}_1\supset \mathrm{p}_1}\,(\mathrm{I}\supset\ 3:\mathrm{p}_1)}{(\mathrm{p}_1\supset \mathrm{p}_1)\supset \mathrm{p}_2}\,(\mathrm{E}\supset) \quad \dfrac{3:\mathrm{p}_1}{\mathrm{p}_1\supset \mathrm{p}_1}\,(\mathrm{I}\supset\ 3:\mathrm{p}_1)}{\mathrm{p}_2}\,(\mathrm{E}\supset)$$

Furthermore, through the isomorphism How, we can easily discover a close connection of each pair of concepts written below:

lambda-beta-calculus	$\mathbf{NJ}_{\supset}$
atomic type	propositional variable
type	formula
lambda-term	proof
- lambda-variable	- assumption
- lambda-application	- (E $\supset$)
- lambda-abstraction	- (I $\supset$)
free variable in a lambda-term	assumption set of a proof
type of a lambda-term	conclusion of a proof
beta-contraction $\triangleright_{1\beta}$	proof-contraction $\triangleright_{1p}$
beta-normal form	normal proof

In the following argument, we identify each pair of concepts written above.

Chapter 2 A simplified proof of the Church-Rosser theorem

As written in chapter 1, the Church-Rosser property of the lambda-beta-calculus is an important property which guarantees that the lambda-beta-calculus is well-behaved as a computation model. In this chapter, we give a new proof by improving the proof given in (Takahashi, 1989). Furthermore, we give a proof method which can be applied to abstract term rewriting systems. The result in this chapter was given by Komori, Yamakawa and the author in (Komori et al., 2014).

Section 2.1 explains how Takahashi (Takahashi, 1989) proved the Church-Rosser theorem. Our proof is given in section 2.2. In section 2.3, we explain some advantages of our proof method. Section 2.4 gives the conclusion of this chapter and give some future works.

In the following argument, we write $M \triangleright_{n\beta} N$ if N is obtained from M by n-step beta contraction, that is, there are lambda-terms $M_0, M_1, \ldots, M_n$ such that

$$M \equiv M_0 \triangleright_{1\beta} M_1 \triangleright_{1\beta} \ldots \triangleright_{1\beta} M_n \equiv N.$$

2.1 Takahashi's proof

The original proof of the theorem was given by Church and Rosser in (Church & Rosser, 1936). However, their proof method was not particularly simple, and many other proof methods have been given by many other researchers (see (Barendregt, 1984) and (Hindley & Seldin, 2008)). One of the simplest proof was given by Takahashi (Takahashi, 1989, 1995). The following notions are key notions of her proof:

Definition 2.1 (Parallel-beta-contraction)**.** We define a binary relation $\triangleright_{1p\beta}$, called *parallel-beta-contraction*, on lambda-terms as follows:

1. $x \triangleright_{1p\beta} x$.

2. If $M \triangleright_{1p\beta} N$ then $\lambda x.M \triangleright_{1p\beta} \lambda x.N$.

3. If $M_1 \triangleright_{1p\beta} N_1$ and $M_2 \triangleright_{1p\beta} N_2$ then $M_1 M_2 \triangleright_{1p\beta} N_1 N_2$.

4. If $M_1 \triangleright_{1p\beta} N_1$ and $M_2 \triangleright_{1p\beta} N_2$ then $(\lambda x.M_1)M_2 \triangleright_{1p\beta} [N_2/x]N_1$.

Furthermore, we define the relation $\triangleright_{p\beta}$ as the reflexive transitive closure of the above relation.

Definition 2.2 (Takahashi-translation)**.** For each lambda-term M, we define the lambda-term M^T as follows:

1. $x^T \equiv x$.
2. $((\lambda x.M_1)M_2)^T \equiv [M_2^T/x]M_1^T$.
3. $(M_1 M_2)^T \equiv M_1^T M_2^T$.
4. $(\lambda x.M)^T \equiv \lambda x.M^T$.

We call this translation Takahashi-translation.

In addition, we inductively define the notation M^{nT} as follows.

1. $M^{0T} \equiv M$.
2. $M^{(n+1)T} \equiv (M^{nT})^T$.

Example 2.3. Let $M \equiv (\lambda x.xx)((\lambda y.y)z)$. Then we obtain all of the following relations.

$$M \rhd_{1p\beta} (\lambda x.xx)((\lambda y.y)z), \quad M \rhd_{1p\beta} ((\lambda y.y)z)((\lambda y.y)z),$$

$$M \rhd_{1p\beta} (\lambda x.xx)z, \quad M \rhd_{1p\beta} zz.$$

Furthermore, we have $M^T \equiv zz$.

Intuitively speaking, parallel-beta-reduction reduces a number of redexes in a lambda-term simultaneously, and Takahashi-translation reduces all of the redexes in a lambda-term simultaneously. From this intuition, we can easily check the following theorem.

Theorem 2.4.

($p1$) $M \rhd_{1\beta} N \implies M \rhd_{1p\beta} N$.

($p2$) $M \rhd_{1p\beta} N \implies M \rhd_{\beta} N$.

($p3$) $M \rhd_{1p\beta} N \implies N \rhd_{1p\beta} M^T$.

Proof. See (Takahashi, 1991). □

With the above properties, Takahashi proved the Church-Rosser property of the lambda-beta-calculus as follows: Suppose $M \rhd_{3\beta} P_3$ and $M \rhd_{2\beta} Q_2$, for example.

$$\begin{array}{ccccccc}
M & \rhd_{1\beta} & P_1 & \rhd_{1\beta} & P_2 & \rhd_{1\beta} & P_3 \\
\triangledown_{1\beta} & & & & & & \\
Q_1 & & & & & & \\
\triangledown_{1\beta} & & & & & & \\
Q_2 & & & & & &
\end{array}$$

Then, from $(p1)$, we have $M \triangleright_{3p\beta} P_3$ and $M \triangleright_{2p\beta} Q_2$.

$$
\begin{array}{ccccccc}
M & \triangleright_{1p\beta} & P_1 & \triangleright_{1p\beta} & P_2 & \triangleright_{1p\beta} & P_3 \\
\triangledown_{1p\beta} & & & & & & \\
Q_1 & & & & & & \\
\triangledown_{1p\beta} & & & & & & \\
Q_2 & & & & & &
\end{array}
$$

Here, from $(p3)$, we can check $P_3 \triangleright_{p\beta} P_1^{2T}$ and $Q_2 \triangleright_{p\beta} P_2^{2T}$ by the following figure.

$$
\begin{array}{ccccccc}
M & \triangleright_{1p\beta} & P_1 & \triangleright_{1p\beta} & P_2 & \triangleright_{1p\beta} & P_3 \\
\triangledown_{1p\beta} & & \triangledown_{1p\beta} & & \triangledown_{1p\beta} & & \triangledown_{1p\beta} \\
Q_1 & \triangleright_{1p\beta} & M^T & \triangleright_{1p\beta} & P_1^T & \triangleright_{1p\beta} & P_2^T \\
\triangledown_{1p\beta} & & \triangledown_{1p\beta} & & \triangledown_{1p\beta} & & \triangledown_{1p\beta} \\
Q_2 & \triangleright_{1p\beta} & Q_1^T & \triangleright_{1p\beta} & M^{2T} & \triangleright_{1p\beta} & P_1^{2T}
\end{array}
$$

Hence, from $(p2)$, we obtain both $P_2 \triangleright_{\beta} P_1^{2T}$ and $Q_3 \triangleright_{\beta} P_1^{2T}$.

2.2 A simplified proof

Parallel reductions have many interesting properties. For example, Takahashi showed the leftmost reduction theorem of the lambda-beta-calculus with $\triangleright_{1p\beta}$ in (Takahashi, 1989). On the other hand, we can say that the notion of parallel reduction is not necessarily essential, because the Church-Rosser theorem is described without this notion. This chapter gives a proof of the theorem without the notion of parallel reduction.

2.2.1 Outline

Recall that M^T is obtained from M by reducing all of the redexes existing in M simultaneously and M^{nT} is obtained from M by applying Takahashi translation n-times. By this intuition, we can expect that Takahashi translation satisfies following properties.

$(t1)$ $M \triangleright_{\beta} M^T$.

$(t2)$ If $M \triangleright_{n\beta} N$ then $N \triangleright_{\beta} M^{nT}$.

Using $(t1)$ and $(t2)$, we shall prove the following fact, which is a stronger statement than the Church-Rosser property.

$(t3)$ If $M \triangleright_{n\beta} M_1$, $M \triangleright_{m\beta} M_2$ and $k = \max\{n, m\}$ then $M_1 \triangleright_{\beta} M^{kT}$ and $M_2 \triangleright_{\beta} M^{kT}$.

This is the outline of our proof.

2.2.2 Proof

First, we can easily verify the following three lemmas by induction on the structure of M.

Lemma 2.5. $\mathrm{FV}_\lambda(M^T) \subseteq \mathrm{FV}_\lambda(M)$.

Lemma 2.6. $M \rhd_\beta M^T$.

Lemma 2.7. If $M \rhd_{1\beta} N$ then $N \rhd_\beta M^T$.

Furthermore, we can obtain the following lemma.

Lemma 2.8. $((\lambda x.M)N)^T \rhd_\beta ([N/x]M)^T$.

Proof. We have $((\lambda x.M)N)^T \equiv [N^T/x]M^T$ and we can verify the following properties, simultaneously, by easy induction on the structure of M.

1. If N is not a lambda-abstraction (i.e. N does not have the form $\lambda y.N'$), then $[N^T/x]M^T \equiv ([N/x]M)^T$.

2. $[\lambda y.N_1^T/x]M^T \rhd_\beta ([\lambda y.N_1/x]M)^T$.

□

These lead to the following lemmas and theorems.

Lemma 2.9. If $M \rhd_{1\beta} N$ then $M^T \rhd_\beta N^T$.

Proof. By induction on the structure of M. We give the proof only of the nontrivial cases below.

1. Let $M \equiv (\lambda x.P_1)P_2$.

   (a) Let $N \equiv [P_2/x]P_1$. In this case, we obtain $M^T \rhd_\beta N^T$ by lemma 2.8.

   (b) Let $N \equiv (\lambda x.Q_1)P_2$ for some Q_1 such that $P_1 \rhd_{1\beta} Q_1$. In this case, we first obtain $N^T \equiv [P_2^T/x]Q_1^T$. Furthermore, by induction hypothesis, $P_1^T \rhd_\beta Q_1^T$. Hence $M^T \equiv [P_2^T/x]P_1^T \rhd_\beta [P_2^T/x]Q_1^T \equiv N^T$.

   (c) Let $N \equiv (\lambda x.P_1)Q_2$ for some Q_2 such that $P_2 \rhd_{1\beta} Q_2$. In this case, we first obtain $N^T \equiv [Q_2^T/x]P_1^T$. Furthermore, by induction hypothesis, $P_2^T \rhd_\beta Q_2^T$. Hence $M^T \equiv [P_2^T/x]P_1^T \rhd_\beta [Q_2^T/x]P_1^T \equiv N^T$.

2. Suppose that $M \equiv P_1 P_2$ where P_1 is not a lambda-abstraction and $N \equiv Q_1 P_2$ for some Q_1 such that $P_1 \rhd_{1\beta} Q_1$. We can easily verify the result when Q_1 is not an abstraction, and so we consider the case when Q_1 is $\lambda y.R$. From $P_1 \rhd_{1\beta} \lambda y.R$ and the induction hypothesis, we obtain $P_1^T \rhd_\beta (\lambda y.R)^T \equiv \lambda y.R^T$. Therefore,

$$M^T \equiv P_1^T P_2^T \rhd_\beta (\lambda y.R^T) P_2^T \rhd_{1\beta} [P_2^T/y] R^T \equiv N^T.$$

□

Lemma 2.10. If $M \rhd_\beta N$ then $M^T \rhd_\beta N^T$.

Proof. If $M \rhd_\beta N$, then we have $M \rhd_{n\beta} N$ for some n. The result is verified by induction on this n. If $n = 0$,then $N \equiv M$. Therefore $N^T \equiv M^T$. If $n > 0$, then there exists R such that $M \rhd_{1\beta} R \rhd_{(n-1)\beta} N$. Here we have $R^T \rhd_\beta N^T$ by induction hypothesis, and we also have $M^T \rhd_\beta R^T$ by lemma 2.9. Therefore $M^T \rhd_\beta N^T$. □

Lemma 2.11. If $M \rhd_\beta N$ then $M^{nT} \rhd_\beta N^{nT}$.

Proof. By applying lemma 2.10 repeatedly. □

Lemma 2.12. If $M \rhd_{n\beta} N$ then $N \rhd_\beta M^{nT}$.

Proof. By induction on n. If $n = 0$ then we clearly say $M \rhd_\beta M \equiv M^{0T}$. Let $n > 0$, there exists R such that $M \rhd_{1\beta} R \rhd_{(n-1)\beta} N$. Here $N \rhd_\beta R^{(n-1)T}$ by induction hypothesis. On the other hand, by $M \rhd_{1\beta} R$, we have $R \rhd_\beta M^T$ using lemma 2.7. So we can get $R^{(n-1)T} \rhd_\beta M^{nT}$ by lemma 2.11. Therefore $N \rhd_\beta M^{nT}$. □

Then we prove the Church-Rosser theorem.

Proof of theorem 1.9. Suppose $M \rhd_\beta N_1$ and $M \rhd_\beta N_2$, then $M \rhd_{n\beta} N_1$ $M \rhd_{m\beta} N_2$ for some n, m. We can assume $n \leq m$. Because we obtain $N_2 \rhd_\beta M^{mT}$ from lemma 2.11, it suffices to show $N_1 \rhd_\beta M^{mT}$. We first obtain $N_1 \rhd_\beta M^{nT}$ from lemma 2.11. Furthermore, we obtain $M^{nT} \rhd_\beta M^{mT}$ from lemma 2.6. Hence $N_1 \rhd_\beta M^{mT}$.

$$\begin{array}{ccccccc}
M & \rhd_{1\beta} & P_1 & \rhd_{1\beta} & P_2 & \rhd_{1\beta} & P_3 \\
\triangledown_{1\beta} & & & & & & \triangledown_{1\beta} \\
Q_1 & & & & & & \\
\triangledown_{1\beta} & & & & & & \vdots \\
Q_2 & \rhd_{1\beta} & \dots & \rhd_{1\beta} & M^{2T} & \rhd_{\beta} & \triangledown_{1\beta} \\
 & & & & & & M^{3T}
\end{array}$$

□

2.3 Some advantages of our proof

Our proof has some advantages in comparison with the existing proof methods.

2.3.1 Brevity

It can be said, in some sense, that our idea improves Takahashi's idea.

The notion of parallel reduction was given by Tait and Martin-Löf in the early 1970s (see (Barendregt, 1984) or (Hindley & Seldin, 2008)). They discovered the diamond property of parallel reduction stated below.

$$\begin{tikzcd}
 & N_1 \arrow[dr, phantom, sloped, "\triangleright_{1p\beta}"] & \\
M \arrow[ur, phantom, sloped, "\triangleright_{1p\beta}"] \arrow[dr, phantom, sloped, "\triangleright_{1p\beta}"] & & \exists R \\
 & N_2 \arrow[ur, phantom, sloped, "\triangleright_{1p\beta}"] &
\end{tikzcd} \qquad \text{(diamond property of } \triangleright_{1p\beta}\text{)}$$

They verified this property by giving R which depends on the form of N_1 and N_2.

Takahashi improved this proof and gave the following stronger property in (Takahashi, 1989).

$$\begin{tikzcd}
 & N_1 \arrow[dr, phantom, sloped, "\triangleright_{1p\beta}"] & \\
M \arrow[ur, phantom, sloped, "\triangleright_{1p\beta}"] \arrow[dr, phantom, sloped, "\triangleright_{1p\beta}"] & & M^T \\
 & N_2 \arrow[ur, phantom, sloped, "\triangleright_{1p\beta}"] &
\end{tikzcd}$$

In her idea, the meeting term M^T can be found without the information of the forms of M_1 and M_2.

Our proof promotes this improvement:

$$\begin{tikzcd}
 & N_1 \arrow[dr, phantom, sloped, "\triangleright_{\beta}"] & \\
M \arrow[ur, phantom, sloped, "\triangleright_{n\beta}"] \arrow[dr, phantom, sloped, "\triangleright_{m\beta}"] & & M^{\max\{n,m\}T} \\
 & N_2 \arrow[ur, phantom, sloped, "\triangleright_{\beta}"] &
\end{tikzcd}$$

Note that, the meeting term $M^{\max\{n,m\}T}$ can be found without the information of the forms of M_1, M_2 and the terms occurring in the processes of the reductions $M \triangleright_{n\beta} N_1$ and $M \triangleright_{m\beta} N_2$.

2.3.2 applicability to other systems

One may notice, by observing our proof, that the properties of Takahashi-translation which essentially work well in our proof are the following three properties.

$(T1)$ $M \rhd_\beta M^T$.

$(T2)$ If $M \rhd_{1\beta} N$ then $M^T \rhd_\beta N^T$.

$(T3)$ If $M \rhd_{1\beta} N$ then $N \rhd_\beta M^T$.

Then, one may also notice that any translation $\circ$ which satisfies the following three properties enables us to prove the Church-Rosser property in the same way.

$(\circ 1)$ $M \rhd_\beta M^\circ$.

$(\circ 2)$ If $M \rhd_{1\beta} N$ then $M^\circ \rhd_\beta N^\circ$.

$(\circ 3)$ If $M \rhd_{1\beta} N$ then $N \rhd_\beta M^\circ$.

This proof method can be applied to a more general case:

An *(abstract) term rewriting system* is a structure $\mathcal{A} = \langle A, \hookrightarrow_1 \rangle$ where $\hookrightarrow_1$ is a binary relation, called a contraction relation, on A. For given contraction relation $\hookrightarrow_1$, we define $\hookrightarrow$ as the reflexive transitive closure. We say $\mathcal{A}$ has the Church-Rosser property if, for each $M, N_1, N_2 \in A$ such that $M \hookrightarrow N_1$ and $M \hookrightarrow N_2$, there exists $R \in A$ such that $N_1 \hookrightarrow R$ and $N_2 \hookrightarrow R$. The Church-Rosser property is one of the most important notion in the study of term rewriting systems.

Theorem 2.13. [1] A term rewriting system $\mathcal{A} = \langle A, \hookrightarrow_1 \rangle$ has the Church-Rosser property, if there exists a translation $\circ$ on A which satisfies the following properties for each $M, N \in A$.

$(\circ 1)$ $M \hookrightarrow M^\circ$.

$(\circ 2)$ If $M \hookrightarrow_1 N$ then $M^\circ \hookrightarrow N^\circ$.

$(\circ 3)$ If $M \hookrightarrow N$ then $N \hookrightarrow M^\circ$.

We say a term rewriting system has the *Z-property* if there exists a translation $\circ$ satisfying the above properties.

This proof method enables us to prove the Church-Rosser property of some other term rewriting systems, in fact. In the following, as an application example, we prove the Church-Rosser property of the lambda-beta-eta-calculus [2] .

[1] This fact was proved in (Komori et al., 2014), but was also proved in (Dehornoy & van Oostrom, 2008) independently.

[2] Other application examples can be found in (Nakazawa & Nagai, 2014), (Nakazawa & Naya, 2015) and (Yamakawa & Komori, 2015), for example.

Definition 2.14 (The lambda-beta-eta-calculus)**.** For each lambda-terms M, N, we write $M \rhd_{1\beta\eta} N$ if $M \hookrightarrow_1 N$ is derivable by the rules $(\beta), (\xi), (\sigma)$ written in definition 1.4 and the following rule.

(η) If $x \notin \mathrm{FV}_\lambda(M)$ then $\lambda x.Mx \hookrightarrow_1 M$.

An eta-redex is a lambda-term of the form $\lambda x.Mx$ where $x \notin \mathrm{FV}_\lambda(M)$. We call the term rewriting system $\langle \Lambda, \rhd_{1\beta\eta}\rangle$ the *lambda-beta-eta-calculus*.

Example 2.15. $\lambda x.yx \rhd_{1\beta\eta} y$ but $\lambda x.xx \not\rhd_{1\beta\eta} x$.

Theorem 2.16 (The Church-Rosser theorem of the lambda-beta-eta-calculus)**.** If $M \rhd_{\beta\eta} N_1$ and $M \rhd_{\beta\eta} N_2$ then there exists R such that $N_1 \rhd_{\beta\eta} R$ and $N_2 \rhd_{\beta\eta} R$.

Proof. By theorem 2.13, it suffices to show that there exists a translation $\circ$ on Λ which satisfies ($\circ$1)-($\circ$3). Such a translation can be given as follows.

1. $x^\circ \equiv x$.

2. $((\lambda x.M_1)M_2)^\circ \equiv [M_2^\circ/x]M_1^\circ$.

3. $(M_1M_2)^\circ \equiv M_1^\circ M_2^\circ$.

4. $(\lambda x.Mx)^\circ \equiv M^\circ$, if $x \notin \mathrm{FV}_\lambda(M)$ and Mx is not a β-redex.

5. $(\lambda x.M)^\circ \equiv \lambda x.M^\circ$.

Actually, we can show this $\circ$ satisfies ($\circ$1)-($\circ$3) as follows:

We can easily verify ($\circ$1) and ($\circ$3), and therefore we prove only ($\circ$2). Because the cases when $M \rhd_{N\beta\eta}$ is derived without the rule (η) can be checked in the same way as the case of the lambda-beta-calculus, we only treat the case when $M \rhd_{N\beta\eta}$ is derived by the rule (η) Let $M \equiv \lambda x.M_1x$ ($x \notin \mathrm{FV}_\lambda(M_1)$) and let M_1 be not an abstract, then N is either M_1 or $\lambda x.N_1x$ ($M_1 \rhd_{1\beta\eta} N_1$). If $N \equiv M_1$, we have $M^\circ \equiv M_1^\circ \equiv N^\circ$. Let $N \equiv \lambda x.N_1x$. First, we have $M_1^\circ \rhd_{\beta\eta} N_1^\circ$ by induction hypothesis. And we also get $x \notin \mathrm{FV}_\lambda(N_1)$ by $M_1 \rhd_{1\beta\eta} N_1$ and $x \notin \mathrm{FV}_\lambda(M_1)$. Therefore we obtain $M^\circ \equiv M_1^\circ \rhd_{\beta\eta} N_1^\circ \equiv N^\circ$. $\square$

Note 2.17. Although $\lambda x.(\lambda x.M)x$ is an eta-redex, we defined $(\lambda x.(\lambda x.M)x)^\circ$ by using beta-contraction. This definition can seem unnatural, and the following definition may be natural.

$$
\begin{aligned}
&(1)\ x^\circ \equiv x\\
&(2)\ ((\lambda x.M_1)M_2)^\circ \equiv [M_2^\circ/x]M_1^\circ\\
&(3)\ (M_1M_2)^\circ \equiv M_1^\circ M_2^\circ\\
&(4)\ (\lambda x.Mx)^\circ \equiv M^\circ,\ \text{if } x \notin FV(M)\\
&(5)\ (\lambda x.M)^\circ \equiv \lambda x.M^\circ
\end{aligned}
$$

Takahashi proved the Church-Rosser theorem of the lambda-beta-eta-calculus with this in (Takahashi, 1989). We can also use this for our method, but this definition makes the proof a little more difficult. For example, if we adopt the above definition, we have to discuss the case when $M \equiv \lambda x.M_1x$ ($x \notin FV(M_1)$), $M_1 \equiv \lambda y.M_2$ and $N \equiv \lambda x.[x/y]M_2$, in addition to the cases we discussed above.

2.4 Conclusion and future work

In this chapter, we give a new proof to the Church-Rosser theorem of the lambda-beta-calculus with the Z-property. In addition, we extract, from our proof, a proof method which can generally apply to other rewriting system.

We think the Z-property has many other interesting properties. For example, Fujita (Fujita, 2015) and Nakazawa (Nakazawa & Fujita, 2015) showed certain properties of some rewriting systems by use of the condition. We expect the study on the Z-property will develop.

Chapter 3 An extension of the Curry-Howard correspondence

In section 1.3, we introduce the Curry-Howard correspondence. It tells us, from proof theoretic view, that the work of the lambda-abstraction can be simulated with Gentzen's proof system **NJ** and the proof contraction, written $\triangleright_{1\supset}$, which gets rid of a detour in a proof. The computational meaning of proofs is now investigated in a wide range of fields, including not only intuitionistic logic but also classical logic and modal logic (Kobayashi, 1997; Miyamoto & Igarashi, 2004).

One of the most famous extension of the Curyy-Howard correspondence was given by Parigot (Parigot, 1992). He gave a typed lambda-calculus, called the lambda-mu-calculus, which corresponds to the $[\supset, \perp]$-fragment of classical logic and has more expressive power than the lambda-beta-calculus. In (Sørensen & Urzyczyn, 2006), it was stated that the lambda-mu-calculus can capture the work of the operators **catch** and **throw** of functional programming language which cannot be captured with the lambda-beta-calculus. In this chapter, we give a typed lambda-calculus, called the intuitionistic lambda-rho-calculus, which corresponds to the implicational fragment of intuitionistic logic and can capture the work of the operators **catch** and **throw**. Because our system is weaker than the lambda-mu-calculus as proof system, it can be said that our result gives a stronger result than the work of (Sørensen & Urzyczyn, 2006). The result of this chapter was given by Fujita, Kashima, Komori and the author in (Fujita et al., 2015; Matsuda, 2015c).

In section 3.1, we give a brief introduction to the lambda-mu-calculus and explain the motivation of our work. Our result is given in section 3.2. Conclusion and future works are written in section 3.3.

3.1 Introduction and preliminary

3.1.1 Preliminary: The lambda-mu-calculus

Parigot (Parigot, 1992, 1993, 2000) refined Griffin's idea (Griffin, 1989) and gave an elegant typed lambda-calculus called the *lambda-mu-calculus* which corresponds to classical logic.

Definition 3.1 (Typed lambda-mu-term)**.**

1. Extended type:

We first extend the set $\mathrm{Tp}_{\rightarrow}$ of types to the set $\mathrm{Tp}_{\rightarrow\perp}$ as follows.

(a) Each atomic type is in $\mathrm{Tp}_{\rightarrow\perp}$.

(b) $\perp$ is in $\mathrm{Tp}_{\rightarrow\perp}$.

(c) If τ, σ are both types then $\tau \rightarrow \sigma$ is in $\mathrm{Tp}_{\rightarrow\perp}$.

We write $\sigma \rightarrow \perp$ as $\neg\sigma$.

2. Lambda-mu-term:

Suppose, for each $\sigma \in \mathrm{Tp}_{\rightarrow\perp}$, a countable set $\mathrm{V}_{\lambda}^{\sigma} = \{\mathtt{x}_1^{\sigma}, \mathtt{x}_2^{\sigma}, \ldots\}$ of typed lambda-variables and a countable set $\mathrm{V}_{\mu}^{\neg\sigma} = \{\mathtt{a}_1^{\neg\sigma}, \mathtt{a}_2^{\neg\sigma}, \ldots\}$ of typed *mu-variables* are given. Then the set $\mathrm{TpTm}'_{\lambda\mu}$ of *(typed) pseudo-lambda-mu-terms* and a mapping Type from $\mathrm{TpTm}'_{\lambda\mu}$ into $\mathrm{Tp}_{\rightarrow\perp}$ are defined as follows:

(a) If $x \in \mathrm{V}_{\lambda}^{\sigma}$ then $x \in \mathrm{TpTm}'_{\lambda\mu}$ and $\mathrm{Type}(x) = \sigma$.

(b) If $M, N \in \mathrm{TpTm}'_{\lambda\mu}$, $\mathrm{Type}(M) = \sigma \rightarrow \tau$ and $\mathrm{Type}(N) = \sigma$ then $MN \in \mathrm{TpTm}'_{\lambda\mu}$ and $\mathrm{Type}(MN) = \tau$.

(c) If $M \in \mathrm{TpTm}'_{\lambda\mu}$, $\mathrm{Type}(M) = \sigma$ and $x \in \mathrm{V}_{\lambda}^{\tau}$ then $\lambda x.M \in \mathrm{TpTm}'_{\lambda\mu}$ and $\mathrm{Type}(\lambda x.M) = \tau \rightarrow \sigma$.

(d) If $M \in \mathrm{TpTm}'_{\lambda\mu}$, $\mathrm{Type}(M) = \sigma$ and $a \in \mathrm{V}_{\mu}^{\neg\sigma}$ then $aM \in \mathrm{TpTm}'_{\lambda\mu}$ and $\mathrm{Type}(aM) = \perp$. A pseudo-lambda-mu-term of this form is called a *mu-application*.

(e) If $M \in \mathrm{TpTm}'_{\lambda\mu}$, $\mathrm{Type}(M) = \perp$, $a \in \mathrm{V}_{\mu}^{\neg\sigma}$ then $\mu a.M \in \mathrm{TpTm}'_{\lambda\mu}$ and $\mathrm{Type}(\mu a.M) = \sigma$. A pseudo-lambda-mu-term of this form is called a *mu-abstraction*.

Then the set $\mathrm{TpTm}_{\lambda\mu} \subseteq \mathrm{TpTm}'_{\lambda\mu}$ of *lambda-mu-terms* is defined as follows:

(a) Each lambda-variable is in $\mathrm{TpTm}_{\lambda\mu}$.

(b) If $M, N \in \mathrm{TpTm}_{\lambda\mu}$, $\mathrm{Type}(M)\sigma \rightarrow \tau$ and $\mathrm{Type}(N) = \sigma$ then $MN \in \mathrm{TpTm}_{\lambda\mu}$.

(c) If $M \in \mathrm{TpTm}_{\lambda\mu}$ and x is a lambda-variable then $\lambda x.M \in \mathrm{TpTm}_{\lambda\mu}$.

(d) If $M \in \mathrm{TpTm}_{\lambda\mu}$, $\mathrm{Type}(M) = \sigma$, $a \in \mathrm{V}_{\mu}^{\neg\sigma}$ and b is a mu-variable then $\mu b.aM \in \mathrm{TpTm}_{\lambda\mu}$.

Although a mu-variable is not a (pseudo-)lambda-mu-term, we sometimes write $\mathrm{Type}(a) = \sigma$ if $a \in \mathrm{V}_{\mu}^{\sigma}$. We use metavariables $M, N, P, Q, \ldots$ for (pseudo-)lambda-mu-terms, $x, y, z, \ldots$ for lambda-variables, $a, b, c, \ldots$ for mu-variables.

3. Free variable:

   For given $M \in \mathrm{TpTm}'_{\lambda\mu}$, the set $\mathrm{FV}_\lambda(M)$ of *free lambda-variables* of M and the set $\mathrm{FV}_\mu(M)$ of *free mu-variables* of M are defined as follows.

   (a) $\mathrm{FV}_\lambda(x) = x$ and $\mathrm{FV}_\mu(x) = \emptyset$.

   (b) $\mathrm{FV}_\lambda(MN) = \mathrm{FV}_\lambda(M) \cup \mathrm{FV}_\lambda(N)$ and $\mathrm{FV}_\mu(MN) = \mathrm{FV}_\mu(M) \cup \mathrm{FV}_\mu(N)$.

   (c) $\mathrm{FV}_\lambda(\lambda x.M) = \mathrm{FV}_\lambda(M) \setminus \{x\}$ and $\mathrm{FV}_\mu(\lambda x.M) = \mathrm{FV}_\mu(M)$.

   (d) $\mathrm{FV}_\lambda(aM) = \mathrm{FV}_\lambda(M)$ and $\mathrm{FV}_\mu(aM) = \mathrm{FV}_\mu(M) \cup \{a\}$.

   (e) $\mathrm{FV}_\lambda(\mu a.M) = \mathrm{FV}_\lambda(M)$ and $\mathrm{FV}_\mu(\mu a.M) = \mathrm{FV}_\mu(M) \setminus \{a\}$.

   We say M is *closed* if $\mathrm{FV}_\lambda(M) = \mathrm{FV}_\mu(M) = \emptyset$.

$\mathrm{TpTm}_{\lambda\mu}$ corresponds to classical logic in the following sense.

Theorem 3.2. For each type (formula) $\sigma \in \mathrm{Tp}_{\to\perp}$, $\vdash_{\mathbf{HK}} \sigma$ if and only if there exists a closed lambda-mu-term M such that $\mathrm{Type}(M) = \sigma$.

Example 3.3. There exists a closed lambda-mu-term whose type is $((\sigma \to \tau) \to \sigma) \to \sigma$. [1] In fact, for $x \in \mathrm{V}_\lambda^\sigma$, $y \in \mathrm{V}_\lambda^{(\sigma\to\tau)\to\sigma}$, $a \in \mathrm{V}_\lambda^{\neg\sigma}$, $b \in \mathrm{V}_\mu^{\neg\tau}$,

$$\mathrm{Type}(\lambda y.\mu a.a(y(\lambda x.\mu b.ax))) = ((\sigma \to \tau) \to \sigma) \to \sigma.$$

Definition 3.4 (Substitution)**.** We introduce two kinds of substitution operations. We first define, for each $M \in \mathrm{TpTm}'_{\lambda\mu}$ and $a, b \in \mathrm{V}_\mu$ such that $\mathrm{Type}(a) = \mathrm{Type}(b)$, the pseudo-lambda-mu-term $[b/a]M$ as follows:

1. $[b/a]M$ is M if $a \notin \mathrm{FV}_\mu(M)$.

2. $[b/a](PQ)$ is $[b/a]P[b/a]Q$.

3. $[b/a](\lambda x.P)$ is $\lambda x.[b/a]P$.

4. $[b/a](aM)$ is $b[b/a]M$.

5. $[b/a](cM)$ is $c[b/a]M$.

6. $[b/a](\mu b.M)$ is $\mu c.[b/a][c/b]M$ where c is the first variable in $\mathrm{V}_\mu^{\mathrm{Type}(b)} \setminus \{b\}$.

7. $[b/a](\mu c.M)$ is $\mu c.[b/a]M$.

[1] This is called *Peirce's formula.* It is known that Peirce's formula is provable in classical logic but is not provable in intuitionistic logic.

Then we define, for each $M, N \in \mathrm{TpTm}'_{\lambda\mu}$ and $x \in \mathrm{V}_\lambda^{\mathrm{Type}(N)}$, the pseudo-lambda-mu-term $[N/x]M$ as follows:

1. $[N/x]M$ is M if $x \notin \mathrm{FV}_\lambda(M)$.
2. $[N/x]x$ is N.
3. $[N/x](PQ)$ is $[N/x]P[N/x]Q$.
4. $[N/x](\lambda y.P)$ is $\lambda y.[N/x]P$ if $y \notin \mathrm{FV}_\lambda(N)$.
5. $[N/x](\lambda y.P)$ is $\lambda z.[N/x][z/y]P$, where z is the first variable in $\mathrm{V}_\lambda^{\mathrm{Type}(y)} \setminus \mathrm{FV}_\lambda(N)$, if $y \in \mathrm{FV}_\lambda(N)$.
6. $[N/x](aP)$ is $a[N/x]P$.
7. $[N/x](\mu a.P)$ is $\mu a.[N/x]P$ if $a \notin \mathrm{FV}_\mu(N)$.
8. $[N/x](\mu a.P)$ is $\mu b.[N/x][b/a]P$, where b is the first variable in $\mathrm{V}_\mu^{\mathrm{Type}(a)} \setminus \mathrm{FV}_\mu(N)$.

Definition 3.5 (alpha-equivalence)**.** Let $M, N \in \mathrm{TpTm}'_{\lambda\mu}$. We say M is alpha-equivalent to N if $M \sim_\alpha$ is derivable by the following rules.

(ρ) $M \sim_\alpha M$.

(τ) If $M_1 \sim_\alpha M_2$ and $M_2 \sim_\alpha M_3$ then $M_1 \sim_\alpha M_3$.

(σ) If $M_1 \sim_\alpha M_2$ then $M_2 \sim_\alpha M_1$.

$(\alpha)_\lambda$ If $[x/y]M \sim_\alpha [x/z]N$ then $\lambda y.M \sim_\alpha \lambda z.N$.

$(\alpha)_\mu$ If $[a/b]M \sim_\alpha [a/c]N$ then $\mu b.M \sim_\alpha \mu c.N$.

In the following, we identify M to N and write $M \equiv N$ if M is alpha-equivalent to N. Then, we introduce more two substitution operations.

Definition 3.6. Let N be a pseudo-lambda-mu-term, and let a, b be mu-variables such that $\mathrm{Type}(N) = \sigma$, $\mathrm{Type}(a) = \neg(\sigma \to \tau)$ and $\mathrm{Type}(b) = \neg\tau$ for some σ, τ. Then, for each $M \in \mathrm{TpTm}'_{\lambda\mu}$, we define the pseudo-lambda-mu-term $[a \Leftarrow b, N]M$ as follows:

1. $[a \Leftarrow b, N]M \equiv M$ if $a \notin \mathrm{FV}_\mu(M)$.
2. $[a \Leftarrow b, N](PQ) \equiv [a \Leftarrow b, N]P[a \Leftarrow b, N]Q$.
3. $[a \Leftarrow b, N](\lambda x.P) \equiv \lambda x.[a \Leftarrow b, N]P$ if $x \notin \mathrm{FV}_\lambda(N)$.

4. $[a \Leftarrow b, N](\lambda x.P) \equiv \lambda y.[a \Leftarrow b, N][y/x]P$, where y is the first variable in $\mathrm{V}_\lambda^{\mathrm{Type}(x)} \setminus \mathrm{FV}_\lambda(N)$, if $x \in \mathrm{FV}_\lambda(N)$.

5. $[a \Leftarrow b, N](aP) \equiv b([a \Leftarrow b, N]PN)$.

6. $[a \Leftarrow b, N](cP) \equiv c[a \Leftarrow b, N]P$.

7. $[a \Leftarrow b, N](\mu c.P) \equiv \mu c.[a \Leftarrow b, N]P$ if $c \notin (\mathrm{FV}_\mu(N) \cup \{b\})$.

8. $[a \Leftarrow b, N](\mu c.P) \equiv \mu d.[a \Leftarrow b, N][d/c]P$, where d is the first variable in $\mathrm{V}_\mu^{\mathrm{Type}(c)} \setminus (\mathrm{FV}_\mu(N) \cup \{b\})$, if $c \in (\mathrm{FV}_\mu(N) \cup \{b\})$.

Let N be a pseudo-lambda-mu-term and let a, b be mu-variables such that $\mathrm{Type}(N) = \sigma \to \tau$, $\mathrm{Type}(a) = \neg\sigma$ and $\mathrm{Type}(b) = \neg\tau$ for some σ, τ. Then, for each $M \in \mathrm{TpTm}'_{\lambda\mu}$, the pseudo-lambda-mu-term $[b, N \Rightarrow a]M$ is defined as follows:

1. $[b, N \Rightarrow a]M \equiv M$ if $a \notin \mathrm{FV}_\mu(M)$.

2. $[b, N \Rightarrow a](PQ) \equiv [b, N \Rightarrow a]P[b, N \Rightarrow a]Q$.

3. $[b, N \Rightarrow a](\lambda x.P) \equiv \lambda x.[b, N \Rightarrow a]P$ if $x \notin \mathrm{FV}_\lambda(N)$.

4. $[b, N \Rightarrow a](\lambda x.P) \equiv \lambda y.[b, N \Rightarrow a][y/x]P$, where y is the first variable in $\mathrm{V}_\lambda^{\mathrm{Type}(x)} \setminus \mathrm{FV}_\lambda(N)$, if $x \in \mathrm{FV}_\lambda(N)$.

5. $[b, N \Rightarrow a](aP) \equiv b(N[b, N \Rightarrow a]P)$.

6. $[b, N \Rightarrow a](cP) \equiv c[b, N \Rightarrow a]P$.

7. $[b, N \Rightarrow a](\mu c.P) \equiv \mu c.[b, N \Rightarrow a]P$ if $c \notin (\mathrm{FV}_\mu(N) \cup \{b\})$.

8. $[b, N \Rightarrow a](\mu c.P) \equiv \mu d.[b, N \Rightarrow a][d/c]P$, where d is the first variable in $\mathrm{V}_\mu^{\mathrm{Type}(c)} \setminus (\mathrm{FV}_\mu(N) \cup \{b\})$, if $c \in (\mathrm{FV}_\mu(N) \cup \{b\})$.

Example 3.7. Consider the term $\mu b.a(\mu c.ax)$ where $\mathrm{Type}(a) = \mathrm{Type}(c) = \neg(\sigma \to \tau)$, $\mathrm{Type}(b) = \theta$ and $\mathrm{Type}(x) = \sigma \to \tau$. Let $\mathrm{Type}(M) = \sigma$, $\mathrm{Type}(N) = (\sigma \to \tau) \to \pi$, $\mathrm{Type}(d) = \neg\tau$ and $\mathrm{Type}(e) = \neg\pi$. Then

$$[a \Leftarrow d, M](\mu b.a(\mu c.ax)) \equiv \mu b.d((\mu c.d(xM))M),$$
$$[e, N \Rightarrow a](\mu b.a(\mu c.ax)) \equiv \mu b.e(N(\mu c.e(Nx))).$$

Definition 3.8. Let $M, N \in \mathrm{TpTm}'_{\lambda\mu}$. We write $M \rhd_{1p} N$ if $M \hookrightarrow_1 N$ can be derived by the following rules:

(τ) $(\lambda x.M)N \hookrightarrow_1 [N/x]M$.

(μr) $(\mu a.M)N \hookrightarrow_1 \mu b.[a \Leftarrow b, N]M$ where $\mathrm{Type}(a) = (\sigma \to \tau)$, $\mathrm{Type}(N) = \sigma$ and $\mathrm{Type}(b) = \neg\tau$.

(μl) $N(\mu a.M) \hookrightarrow_1 \mu b.[b, N \Rightarrow a]M$ where $\mathrm{Type}(a) = \neg\sigma$, $\mathrm{Type}(N) = \sigma \to \tau$ and $\mathrm{Type}(b) = \neg\tau$.

(ζ) $a(\mu b.M) \hookrightarrow_1 [a/b]M$.

(η_μ) $\mu a.aM \hookrightarrow_1 M$ if $a \notin \mathrm{FV}_\mu(M)$.

(ξ_λ) If $M \hookrightarrow_1 N$ then $\lambda x.M \hookrightarrow_1 \lambda x.N$.

(ξ_μ) If $M \hookrightarrow_1 N$ then $\mu a.M \hookrightarrow_1 \mu a.N$.

(σ_λ) If $M \hookrightarrow_1 N$ then $PM \hookrightarrow_1 PN$ and $MQ \hookrightarrow_1 NQ$.

(σ_μ) If $M \hookrightarrow_1 N$ then $aM \hookrightarrow_1 aN$.

In addition, we define the relation $\rhd_p$ as the reflexive transitive closure of $\rhd_{1p}$.

Example 3.9.

1. Let $M \equiv (\mu a.a(\mu c.ax))y$, where $\mathrm{Type}(a) = \mathrm{Type}(c) = \neg(\sigma \to \tau)$, $\mathrm{Type}(x) = \sigma \to \tau$ and $\mathrm{Type}(y) = \sigma$. Then we obtain $M \rhd_{1p} \mu d.d((\mu c.d(xy))y)$, where d is a mu-variable such that $\mathrm{Type}(d) = \neg\tau$.

2. Let $\mathrm{Type}(a) = \neg\sigma$, $\mathrm{Type}(b) = \neg\tau$, $\mathrm{Type}(N) = \sigma$, $\mathrm{Type}(P) = \tau \to \theta$, $\mathrm{Type}(Q) = \theta \to \sigma$ and $\mathrm{FV}_\mu(N) = \emptyset$. Then we obtain

$$\begin{array}{llll} \mu a.a((P(\mu b.aN))Q) & \rhd_{1p} & \mu a.a((\mu c.aN)Q) & ((\mu l), (\xi_\mu), (\sigma_\lambda) \text{ and } (\sigma_\mu)) \\ & \rhd_{1p} & \mu a.a(\mu d.aN) & ((\mu r), (\xi_\mu), (\sigma_\lambda) \text{ and } (\sigma_\mu)) \\ & \rhd_{1p} & \mu a.aN & ((\zeta) \text{ and } (\xi_\mu)) \\ & \rhd_{1p} & N & ((\eta_\mu)) \end{array}$$

where $c \in \mathrm{V}_\mu^{\neg\theta}$ and $d \in \mathrm{V}_\mu^{\neg\sigma}$.

In general, if $\mathrm{FV}_\mu(N) = \emptyset$ and $a \not\equiv b$, then we obtain

$$(\mu b.aN)P \rhd_{1p} \mu c.aN, \quad Q(\mu b.aN) \rhd_{1p} \mu d.aN, \quad \mu a.a(\mu b.aN) \rhd_p N$$

for some c, d.

Theorem 3.10. If $M \in \mathrm{TpTm}_{\lambda\mu}$ and $M \rhd_{1p} N$ then $N \in \mathrm{TpTm}_{\lambda\mu}$ and $\mathrm{Type}(N) = \mathrm{Type}(M)$.

We call the system $\langle \mathrm{TpTm}_{\lambda\mu}, \rhd_{1p} \rangle$ the *(typed) lambda-mu-calculus*. The lambda-mu-calculus has higher expressive power than the lambda-beta-calculus. The following application examples [2] tells us the expressive power of the lambda-mu-calculus.

Example 3.11.

1. The lambda-mu-calculus can treat streams, which are sequences of data elements made available over time. For this topic, see (Nakazawa & Katsumata, 2012; Saurin, 2005) for example.

2. Some functional programming languages have operators **catch** and **throw** [3] . In (Sørensen & Urzyczyn, 2006, chapter 6), it is stated that the lambda-mu-calculus can capture the work of those operators, which cannot be captured with the lambda-beta-calculus:

   We first define the set $\mathcal{C}_{\lambda\mu}$ of *lambda-mu-catch-contexts* and a mapping $\mathrm{Type}^c : \mathcal{C}_{\lambda\mu} \to \mathrm{Tp}_{\to\perp}$ as follows (we write $C[\ ]_\varphi : \sigma$ if both $C[\ ]_\varphi \in \mathcal{C}_{\lambda\mu}$ and $\mathrm{Type}^c(C[\ ]_\varphi) = \sigma$ hold).

   $(c0)$ $[\![\ ]\!]_\varphi : \varphi$.

   $(c1)$ $C[\ ]_\varphi : \sigma,\ M \in \mathrm{TpTm}'_{\lambda\mu},\ \mathrm{Type}(M) = \sigma \to \tau \implies MC[\ ]_\varphi : \tau$.

   $(c2)$ $C[\ ]_\varphi : \sigma \to \tau,\ M \in \mathrm{TpTm}'_{\lambda\mu},\ \mathrm{Type}(M) = \sigma \implies C[\ ]_\varphi M : \tau$.

   We use metavariables $C, D, \ldots$ to stand for arbitrary contexts. Parentheses are omitted under the convention of association to the left. We then define the pseudo-lambda-mu-term $C[M]_\varphi$, for each $M \in \mathrm{TpTm}_{\lambda\mu}$ such that $\mathrm{Type}(M) = \varphi$ and each $C[\ ]_\varphi \in \mathcal{C}_{\lambda\mu}$, as follows.

   $(c0)'$ $C[\ ]_\varphi \equiv [\![\ ]\!]_\varphi \implies C[M]_\varphi \equiv M$.

   $(c1)'$ $C[\ ]_\varphi \equiv ND[\ ]_\varphi \implies C[M]_\varphi \equiv ND[M]_\varphi$.

   $(c2)'$ $C[\ ]_\varphi \equiv D[\ ]_\varphi N \implies C[M]_\varphi \equiv D[M]_\varphi N$.

[2] Other examples can be found in (Bierman, 1998), for example.

[3] In (Sørensen & Urzyczyn, 2006), the following intuitive explanation for those operators is given: The program $P =$ **catch** a **in** M normally returns the result of the program M. however, if we encounter the program **throw** N **to** a during evaluating M, then the evaluation of M is aborted and P returns the result of N. For example, $1 + ($**catch** a **in** $(2 + ($**throw** 3 **to** $a)))$ returns 4. Cite also (Graham, 1996).

We give the terms which work as the **catch** operator and the **throw** operator as follows.

$$\textbf{catch } a \textbf{ in } M \equiv \mu a.aM, \quad \textbf{throw } N \textbf{ to } a \equiv \mu b.aN$$

where b is an appropriate mu-variable which depends on the context. We can easily show that if $\mathrm{Type}(a) = \neg\sigma$ and $\mathrm{FV}_\mu(N) = \emptyset$ then, for any lambda-mu-catch-context $C[\ ]_\tau : \sigma$,

$$\textbf{catch } a \textbf{ in } C[\textbf{throw } N \textbf{ to } a]_\tau \rhd_p N.$$

See also example 3.9-2.

3.1.2 Motivation and aim of chapter 3

Subsection 3.1.1 introduced a typed lambda-calculus, which is called the lambda-mu-calculus and corresponds to the $[\to, \bot]$-fragment of classical logic, and showed the work of the operators **catch** and **throw** can partly be simulated with the system. It is clear that the work of the operators cannot be simulated with the lambda-beta-calculus, and hence it can be said that the lambda-mu-calculus has higher expressive power than the lambda-beta-calculus. However let me raise the following questions here:

Q1. Recall that the typed lambda-beta-calculus corresponds to intuitionistic logic but the lambda-mu-calculus corresponds to classical logic. Here, do we essentially need the extension? In other words, is there a typed lambda-calculus which corresponds to intuitionistic logic and can simulate **catch** and **throw**?

Q2. Recall that the typed lambda-beta-calculus treats only simple types but the typed lambda-mu-calculus treats extends the notion of type. Here, do we essentially need the extension? In other words, is there a simple typed lambda-calculus which can simulate **catch** and **throw**?

Q3. Parigot's symmetric contraction $\rhd_{1p}$ is very complex and is difficult to treat. Is there a typed lambda-calculus which can simulate **catch** and **throw** but whose contraction rules are easier to treat the lambda-mu-calculus?

The aim of this chapter is to give an affirmative answer to the above questions, in other words, to give a typed lambda-calculus which corresponds to implicational fragment of intuitionistic logic and can simulate the work of the **catch** operator and the **throw** operator.

In section 3.2, we will introduce the typed lambda-calculus stated above and will investigate the system. Our system is based on the lambda-rho-calculus, given by Komori

(Komori, 2013). Before going to section 3.2, we introduce Komori's system in the following subsection.

3.1.3 Preliminary: The lambda-rho-calculus

Komori got inspiration from the proof system for implicational classical logic given in (Baba, Hirokawa, Kashima, Komori, & Takeuti, 2000), and introduced a typed lambda-calculus called the *lambda-rho-calculus* corresponding to implicational fragment of classical logic in (Komori, 2013).

Definition 3.12 (The lambda-rho-calculus)**.** Suppose, for each $\sigma \in \mathrm{Tp}_{\to}$, a countable set $\mathrm{V}_{\lambda}^{\sigma} = \{\mathtt{x}_1^{\sigma}, \mathtt{x}_2^{\sigma}, \ldots\}$ of typed lambda-variables and a countable set $\mathrm{V}_{\rho}^{\sigma} = \{\mathtt{a}_1^{\sigma}, \mathtt{a}_2^{\sigma}, \ldots\}$ of typed *rho-variables* (we use metavariables $a, b, c, \ldots$ for rho-variables) are given. Then the set $\mathrm{TpTm}_{\lambda\rho}$ of *(typed) lambda-rho-terms* and a mapping Type from $\mathrm{TpTm}'_{\lambda\rho}$ into $\mathrm{Tp}_{\to}$ are defined as follows:

1. If $x \in \mathrm{V}_{\lambda}^{\sigma}$, then $x \in \mathrm{TpTm}_{\lambda\rho}$ and $\mathrm{Type}(x) = \sigma$.

2. If $M, N \in \mathrm{TpTm}_{\lambda\rho}$ such that $\mathrm{Type}(M) = \sigma \to \tau$ and $\mathrm{Type}(N) = \sigma$, then $MN \in \mathrm{TpTm}_{\lambda\rho}$ and $\mathrm{Type}(MN) = \tau$.

3. If $M \in \mathrm{TpTm}_{\lambda\rho}$ such that $\mathrm{Type}(M) = \sigma$ and $x \in \mathrm{V}_{\lambda}^{\tau}$, then $\lambda x.M \in \mathrm{TpTm}_{\lambda\rho}$ and $\mathrm{Type}(\lambda x.M) = \tau \to \sigma$.

4. If $M \in \mathrm{TpTm}_{\lambda\rho}$ such that $\mathrm{Type}(M) = \sigma$ and $a \in \mathrm{V}_{\rho}^{\sigma}$, then $(aM)^{\tau} \in \mathrm{TpTm}_{\lambda\rho}$ and $\mathrm{Type}((aM)^{\tau}) = \tau$. We call a term of this form a *rho-application*.

5. If $M \in \mathrm{TpTm}_{\lambda\rho}$ such that $\mathrm{Type}(M) = \sigma$ and $a \in \mathrm{V}_{\rho}^{\sigma}$, then $\rho a.M \in \mathrm{TpTm}_{\lambda\rho}$ and $\mathrm{Type}(\rho a.M) = \sigma$. We call a term of this form a *rho-abstraction*.

We sometimes write $\bigcup_{\sigma \in \mathrm{Tp}_{\to}} \mathrm{V}_{\lambda}^{\sigma}$ as V_{λ} and write $\bigcup_{\sigma \in \mathrm{Tp}_{\to}} \mathrm{V}_{\rho}^{\sigma}$ as V_{ρ}.

Next, we define, for each $M \in \mathrm{TpTm}_{\lambda\rho}$, the set $\mathrm{FV}_{\lambda}(M)$ of free lambda-variables in M, the set of $\mathrm{FV}_{\rho}(M)$ of free rho-variables in M, the set $\mathrm{BV}_{\lambda}(M)$ of bound lambda-variables in M and the set $\mathrm{BV}_{\rho}(M)$ of bound rho-variables in M as follows:

1. $\mathrm{FV}_{\lambda}(x) = \{x\}$, $\mathrm{FV}_{\rho}(x) = \mathrm{BV}_{\lambda}(x) = \mathrm{BV}_{\rho}(x) = \emptyset$.

2. $\mathrm{FV}_{\lambda}(MN) = \mathrm{FV}_{\lambda}(M) \cup \mathrm{FV}_{\lambda}(N)$, $\mathrm{FV}_{\rho}(MN) = \mathrm{FV}_{\rho}(M) \cup \mathrm{FV}_{\rho}(N)$, $\mathrm{BV}_{\lambda}(MN) = \mathrm{BV}_{\lambda}(M) \cup \mathrm{BV}_{\lambda}(N)$ and $\mathrm{BV}_{\rho}(MN) = \mathrm{BV}_{\rho}(M) \cup \mathrm{BV}_{\rho}(N)$.

3. $\mathrm{FV}_{\lambda}(\lambda x.M) = \mathrm{FV}_{\lambda}(M) \setminus \{x\}$, $\mathrm{FV}_{\rho}(\lambda x.M) = \mathrm{FV}_{\rho}(M)$, $\mathrm{BV}_{\lambda}(\lambda x.M) = \mathrm{BV}_{\lambda}(M) \cup \{x\}$ and $\mathrm{BV}_{\rho}(\lambda x.M) = \mathrm{BV}_{\rho}(M)$.

4. $\mathrm{FV}_\lambda((aM)^\sigma) = \mathrm{FV}_\lambda(M)$, $\mathrm{FV}_\rho((aM)^\sigma) = \mathrm{FV}_\rho(M) \cup \{a\}$, $\mathrm{BV}_\lambda((aM)^\sigma) = \mathrm{BV}_\lambda(M)$ and $\mathrm{BV}_\rho((aM)^\sigma) = \mathrm{BV}_\rho((aM)^\sigma$.

5. $\mathrm{FV}_\lambda(\rho a.M) = \mathrm{FV}_\lambda(M)$, $\mathrm{FV}_\rho(\rho a.M) = \mathrm{FV}_\rho(M) \setminus \{a\}$, $\mathrm{BV}_\lambda(\rho a.M) = \mathrm{BV}_\lambda(M)$ and $\mathrm{BV}_\rho(\rho a.M) = \mathrm{BV}_\rho(M) \cup \{a\}$.

We say M is closed if $\mathrm{FV}_\lambda(M) \cup \mathrm{FV}_\rho(M) = \emptyset$.

Theorem 3.13. For each type $\sigma \in \mathrm{Tp}_\rightarrow$, $\vdash_{\mathbf{HK}_\supset} \sigma$ if and only if there exists a closed lambda-rho-term M such that $\mathrm{Type}(M) = \sigma$.

Example 3.14. We have

$$\mathrm{Type}(\lambda y.\rho a.(y(\lambda x.(ax)^\tau))) = ((\sigma \to \tau) \to \sigma) \to \sigma,$$

where $x \in \mathrm{V}_\lambda^\sigma$, $y \in \mathrm{V}_\lambda^{(\sigma\to\tau)\to\sigma}$ and $a \in \mathrm{V}_\rho^\sigma$.

We define the substitutions $[N/x]M$ and $[b/a]M$ and the notion of alpha-equivalent ($M \equiv N$) in the same way as the lambda-mu-calculus.

3.2 Intuitionistic lambda-rho-calculus

In this section, we give a typed lambda-calculus which satisfies the claims given in the questions Q1 and Q2 in subsection 3.1.2. In other words, we give a typed lambda-calculus which corresponds to the implicational fragment of intuitionistic logic and can capture the work of the operators **catch** and **throw**. Furthermore, we can say our system satisfies the claim in the question Q3, that is, the contraction of our system is easier, in some sense, to treat than Parigot's contraction.

We give the subsystem in subsection 3.2.1, and show some basic properties in subsection 3.2.2. In subsection 3.2.3, we show the system corresponds to intuitionistic logic. Then we explain how we can simulate the **catch** operator and the **throw** operator in 3.2.4. Last, as an example which shows the ease of use of our system, we show the strong normalization property of our system in subsection 3.2.5.

3.2.1 Definition

Our system is a subsystem of the lambda-rho-calculus given as follows:

Definition 3.15 (Intuitionistic lambda-rho-term)**.** We first define, for each $M \in \mathrm{TpTm}_{\lambda\rho}$ and $a \in \mathrm{V}_\rho$, the set $\mathrm{FV}_\lambda^a(M) \subseteq \mathrm{FV}_\lambda(M)$ as follows.

1. $\mathrm{FV}_\lambda^a(x) = \emptyset$.

2. $\mathrm{FV}^a_\lambda(MN) = \mathrm{FV}^a_\lambda(M) \cup \mathrm{FV}^a_\lambda(N)$.

3. $\mathrm{FV}^a_\lambda(\lambda x.M) = \mathrm{FV}^a_\lambda(M) \setminus \{x\}$.

4. $\mathrm{FV}^a_\lambda((aM)^\sigma) = \mathrm{FV}_\lambda(M)$.

5. $\mathrm{FV}^a_\lambda((bM)^\sigma) = \mathrm{FV}^a_\lambda(M)$.

6. $\mathrm{FV}^a_\lambda(\rho a.M) = \emptyset$.

7. $\mathrm{FV}^a_\lambda(\rho b.M) = \mathrm{FV}^a_\lambda(M)$.

Then, we define the set $\mathrm{TpTm}^I_{\lambda\rho}$ of *intuitionistic lambda-rho-terms* as follows.

1. Each $x \in \mathrm{V}_\lambda$ is in $\mathrm{TpTm}^I_{\lambda\rho}$.

2. If $M, N \in \mathrm{TpTm}^I_{\lambda\rho}$, $\mathrm{Type}(M) = \sigma \to \tau$ and $\mathrm{Type}(N) = \sigma$, then $MN \in \mathrm{TpTm}^I_{\lambda\rho}$.

3. If $M \in \mathrm{TpTm}^I_{\lambda\rho}$ satisfies $\mathrm{FV}^a_\lambda(M) = \emptyset$ for each $a \in \mathrm{FV}_\rho(M)$, then $\lambda x.M \in \mathrm{TpTm}^I_{\lambda\rho}$.

4. If $M \in \mathrm{TpTm}^I_{\lambda\rho}$ and $a \in \mathrm{V}_\rho^{\mathrm{Type}(M)}$ then $(aM)^\sigma \in \mathrm{TpTm}^I_{\lambda\rho}$.

5. If $M \in \mathrm{TpTm}^I_{\lambda\rho}$ and $a \in \mathrm{V}_\rho^{\mathrm{Type}(M)}$ then $\rho a.M \in \mathrm{TpTm}^I_{\lambda\rho}$.

Intuitively speaking, a closed lambda-rho-term is in $\mathrm{TpTm}^I_{\lambda\rho}$ if it does not include a subterm of the form

$$\rho a.(\ldots(\lambda x.(\ldots(a(\ldots x \ldots))^\sigma \ldots)\ldots).$$

Note that the term $\lambda y.\rho a.(y(\lambda x.(ax)^\tau))$, which was given in example 3.14 and whose type is $((\sigma \to \tau) \to \sigma) \to \sigma$ (Pierce's formula), is not in $\mathrm{TpTm}^I_{\lambda\rho}$ because $\mathrm{FV}^a_\lambda((ax)^\tau) = \{x\}$.

We define the following term rewriting rules for our system.

Definition 3.16. We write $M \rhd_{1ct} N$, for $M, N \in \mathrm{TpTm}_{\lambda\rho}$, if $M \hookrightarrow_1 N$ can be derived by the following rules:

$$
\begin{array}{rl}
(\tau) & (\lambda x.M)N \hookrightarrow_1 [N/x]M. \\
(\textbf{throw}\ \lambda\text{-app}\ l) & N(aM)^\sigma \hookrightarrow_1 (aM)^\tau \ \text{ where Type}(N) = \sigma \to \tau. \\
(\textbf{throw}\ \lambda\text{-app}\ r) & (aM)^{\sigma\to\tau}N \hookrightarrow_1 (aM)^\tau. \\
(\textbf{throw}\ \lambda\text{-abs}) & \lambda x.(aM)^\tau \hookrightarrow_1 (aM)^{\sigma\to\tau} \ \text{ where Type}(x) = \sigma. \\
(\textbf{throw}\ \rho\text{-app}) & (b(aM)^\sigma)^\tau \hookrightarrow_1 (aM)^\tau. \\
(\textbf{throw}\ \rho\text{-abs}) & \rho a.M \hookrightarrow_1 M \ \text{ if } a \notin \mathrm{FV}_\rho(M). \\
(\textbf{catch}) & \rho a.(aM)^\sigma \hookrightarrow_1 M \ \text{ if } a \notin \mathrm{FV}_\rho(M). \\
(\sigma_\lambda) & \text{If } M \hookrightarrow_1 N \text{ then } RM \hookrightarrow_1 RN \text{ and } MR \hookrightarrow_1 NR. \\
(\xi_\lambda) & \text{If } M \hookrightarrow_1 N \text{ then } \lambda x.M \hookrightarrow_1 \lambda x.N. \\
(\sigma_\rho) & \text{If } M \hookrightarrow_1 N \text{ then } (aM)^\varphi \hookrightarrow_1 (aN)^\varphi. \\
(\xi_\rho) & \text{If } M \hookrightarrow_1 N \text{ then } \rho a.M \hookrightarrow_1 \rho a.N.
\end{array}
$$

In addition, we define the relation $\triangleright_{ct}$ as the reflexive transitive closure of $\triangleright_{1ct}$.

We call the system $\langle \mathrm{TpTm}^I_{\lambda\rho}, \triangleright_{1ct}\rangle$ the *(typed) intuitionistic lambda-rho-calculus.*

Example 3.17. Let $\mathrm{Type}(M) = \sigma$, $\mathrm{Type}(P) = \tau \to \theta \to \sigma$, $\mathrm{Type}(Q) = \theta$, $a \in \mathrm{V}^\sigma_\rho$, $a \notin \mathrm{FV}_\rho(M)$ and $\rho a.P(aM)^\tau Q \in \mathrm{TpTm}^I_{\lambda\rho}$, then we obtain

$$
\begin{array}{rlll}
\rho a.P(aM)^\tau Q & \triangleright_{1ct} & \rho a.(aM)^{\theta\to\sigma}Q & ((\textbf{throw}\ \lambda\text{-app}\ l), (\sigma_\lambda), (\xi_\rho)) \\
 & \triangleright_{1ct} & \rho a.(aM)^\sigma & ((\textbf{throw}\ \lambda\text{-app}\ r), (\xi_\rho)) \\
 & \triangleright_{1ct} & M & ((\textbf{catch})).
\end{array}
$$

3.2.2 Basic properties

This subsection shows some basic properties of our system. The goal of this subsection is to show that $\mathrm{TpTm}^I_{\lambda\rho}$ is closed under the relation $\triangleright_{1ct}$:

Theorem 3.18. If $M \in \mathrm{TpTm}^I_{\lambda\rho}$ and $M \triangleright_{1ct} N$ then $N \in \mathrm{TpTm}^I_{\lambda\rho}$.

Before we prove this theorem, we prepare the following properties.

Lemma 3.19. If $M \in \mathrm{TpTm}^I_{\lambda\rho}$ and $M \triangleright_{1ct} N$ then $\mathrm{FV}_\lambda(M) \supseteq \mathrm{FV}_\lambda(N)$ and $\mathrm{FV}_\rho(M) \supseteq \mathrm{FV}_\rho(N)$.

Proof. By induction on the clauses of definition 3.16. The only nontrivial case is the case when $M \hookrightarrow_1 N$ is derived by the rule (**throw** λ-abs). Let $M \equiv \lambda x.(aP)^\sigma$ and $N \equiv (aP)^{\tau\to\sigma}$

where $\mathrm{Type}(x) = \tau$. In this case, from $\lambda x.(aP)^\sigma \in \mathrm{TpTm}^I_{\lambda\rho}$, we obtain $x \notin \mathrm{FV}_\lambda(P)$. Hence $\mathrm{FV}_\lambda(\lambda x.(aP)^\sigma) = \mathrm{FV}_\lambda((aP)^{\tau\to\sigma})$. Furthermore, $\mathrm{FV}_\rho(\lambda x.(aP)^\sigma) = \mathrm{FV}_\rho((aP)^{\tau\to\sigma})$ is obvious. □

Theorem 3.20 (Subject reduction property)**.** If $M \in \mathrm{TpTm}^I_{\lambda\rho}$ and $M \triangleright_{1ct} N$ then $\mathrm{Type}(M) = \mathrm{Type}(N)$.

Proof. Obvious. □

Lemma 3.21.

1. If $x \notin \mathrm{FV}^a_\lambda(M)$ then $\mathrm{FV}^a_\lambda([N/x]M) \subseteq \mathrm{FV}^a_\lambda(M) \cup \mathrm{FV}^a_\lambda(N)$.
2. If a does not occur in M then $\mathrm{FV}^a_\lambda([N/x]M) \subseteq \mathrm{FV}^a_\lambda(N)$.
3. If $x, y \notin \mathrm{FV}^a_\lambda(M)$ and $y \notin \mathrm{FV}^a_\lambda(N)$ then $y \notin \mathrm{FV}^a_\lambda([N/x]M)$.

Proof. (2) and (3) are easy consequences of (1). Then we show (1) by induction on the size of M. We can assume $\mathrm{FV}_\lambda(M) \cap \mathrm{BV}_\lambda(M) = \emptyset$, $\mathrm{FV}_\rho(M) \cap \mathrm{BV}_\rho(M) = \emptyset$ and $a \notin \mathrm{BV}_\rho(M)$.

(A) Suppose $a \notin \mathrm{FV}_\rho(M)$. In this case, we can show $\mathrm{FV}_\lambda([N/x]M) \subseteq \mathrm{FV}^a_\lambda(N)$ by induction on the size of M.

(A-1) If $x \notin \mathrm{FV}_\lambda(M)$, then

$$\mathrm{FV}^a_\lambda([N/x]M) = \mathrm{FV}^a_\lambda(M) = \emptyset \subseteq \mathrm{FV}_\lambda(N).$$

(A-2) If $M \equiv x$, then

$$\mathrm{FV}^a_\lambda([N/x]M) = \mathrm{FV}^a_\lambda(N).$$

(A-3) If $M \equiv PQ$, then

$$\begin{aligned}\mathrm{FV}^a_\lambda([N/x]M) &= \mathrm{FV}^a_\lambda([N/x]P[N/x]Q) = \\ &= \mathrm{FV}^a_\lambda([N/x]P) \cup \mathrm{FV}^a_\lambda([N/x]Q).\end{aligned}$$

Here, by $a \notin \mathrm{FV}^a_\rho(M)$, we have $a \notin \mathrm{FV}^a_\rho(P)$ and $a \notin \mathrm{FV}^a_\rho(Q)$. Then, by induction hypothesis, we have $\mathrm{FV}^a_\lambda([N/x]P) \subseteq \mathrm{FV}^a_\lambda(N)$ and $\mathrm{FV}^a_\lambda([N/x]Q) \subseteq \mathrm{FV}^a_\lambda(N)$. Hence, $\mathrm{FV}^a_\lambda([N/x]M) \subseteq \mathrm{FV}^a_\lambda(N)$.

(A-4) If $M \equiv \lambda y.P$ $(y \not\equiv x)$, then

$$\mathrm{FV}^a_\lambda([N/x]M) = \mathrm{FV}^a_\lambda([N/x]P).$$

By induction hypothesis, we have $\mathrm{FV}^a_\lambda([N/x]P) \subseteq \mathrm{FV}^a_\lambda(N)$. The case when M is either $(bP)^\sigma$ or $\rho b.P$ $(b \not\equiv a)$ can be proved in the same way.

(B) If $M \equiv PQ$, then

$$\begin{aligned}\mathrm{FV}^a_\lambda([N/x]M) &= \mathrm{FV}^a_\lambda([N/x]P[N/x]Q)\\ &= \mathrm{FV}^a_\lambda([N/x]P) \cup \mathrm{FV}^a_\lambda([N/x]Q).\end{aligned}$$

By induction hypothesis, we have $\mathrm{FV}^a_\lambda([N/x]P) \subseteq \mathrm{FV}^a_\lambda(P) \cup \mathrm{FV}^a_\lambda(N)$ and $\mathrm{FV}^a_\lambda([N/x]Q) \subseteq \mathrm{FV}^a_\lambda(Q) \cup \mathrm{FV}^a_\lambda(N)$. Hence we obtain

$$\begin{aligned}\mathrm{FV}^a_\lambda([N/x]M) &\subseteq (\mathrm{FV}^a_\lambda(P) \cup \mathrm{FV}^a_\lambda(N)) \cup (\mathrm{FV}^a_\lambda(Q) \cup \mathrm{FV}^a_\lambda(N))\\ &= (\mathrm{FV}^a_\lambda(P) \cup \mathrm{FV}^a_\lambda(Q)) \cup \mathrm{FV}^a_\lambda(N)\\ &= \mathrm{FV}^a_\lambda(M) \cup \mathrm{FV}^a_\lambda(N).\end{aligned}$$

The case when M is either $\lambda y.P$, $(bP)^\sigma$ or $\rho b.P$ ($y \not\equiv x$, $b \not\equiv a$) can be proved in the same way.

(C) Suppose $M \equiv (aP)^\sigma$. In this case, from the condition $x \notin \mathrm{FV}^a_\lambda(M)$, we have $x \notin \mathrm{FV}_\lambda(M)$. Therefore we obtain $\mathrm{FV}_\lambda([N/x]M) = \mathrm{FV}_\lambda(M)$.

$\square$

Lemma 3.22. If M and N are both in $\mathrm{TpTm}^I_{\lambda\rho}$ then $[N/x]M$ is also in $\mathrm{TpTm}^I_{\lambda\rho}$.

Proof. By induction on the size of M. The only nontrivial case is the case when M is a λ-abstraction. Suppose $M \equiv \lambda y.P$. We can assume $y \notin \mathrm{FV}_\lambda(N)$. By induction hypothesis, $[N/x]P$ is in $\mathrm{TpTm}^I_{\lambda\rho}$. Furthermore, because $M \in \mathrm{TpTm}^I_{\lambda\rho}$, we obtain $y \notin \bigcup_{a \in \mathrm{FV}_\rho(P)} \mathrm{FV}^a_\lambda(P)$. Then, with lemma 3.21-(3), we obtain

$$y \notin \bigcup_{a \in \mathrm{FV}_\rho([N/x]P)} \mathrm{FV}^a_\lambda([N/x]P)$$

Therefore we obtain $\lambda y.[N/x]P \in \mathrm{TpTm}^I_{\lambda\rho}$. $\square$

Then we show the set $\mathrm{TpTm}^I_{\lambda\rho}$ is closed under the relation $\triangleright_{1ct}$ (theorem 3.18)

Proof of theorem 3.18. Suppose $M \in \mathrm{TpTm}^I_{\lambda\rho}$ and $M \triangleright_{1ct} N$. We show, by induction on the clauses of definition 3.16, the following conditions simultaneously.

[♯1] $\mathrm{FV}^a_\lambda(M) \supseteq \mathrm{FV}^a_\lambda(N)$ for any $a \in \mathrm{FV}_\rho(M)$.

[♯2] $N \in \mathrm{TpTm}^I_{\lambda\rho}$.

The nontrivial cases are the following two cases.

1. Suppose $M \rhd_{1ct} N$ is derived by (τ), that is, there exist P and Q such that $M \equiv (\lambda x.P)Q \rhd_{1ct} [Q/x]P \equiv N$. [♯2] is obtained from lemma 3.22. Then we show [♯1].

   Suppose $a \in \mathrm{FV}_\rho(P)$. Because $\lambda x.P \in \mathrm{TpTm}^I_{\lambda\rho}$, we have $x \notin \mathrm{FV}^a_\lambda(P)$. Then, from lemma 3.21-(1), we obtain

$$\mathrm{FV}^a_\lambda([Q/x]P) \subseteq \mathrm{FV}^a_\lambda(P) \cup \mathrm{FV}^a_\lambda(Q) = \mathrm{FV}^a_\lambda((\lambda x.P)Q).$$

   Suppose $a \notin \mathrm{FV}_\rho(P)$. Then, from lemma 3.21-(2), we obtain

$$\mathrm{FV}^a_\lambda([Q/x]P) \subseteq \mathrm{FV}^a_\lambda(Q) = \mathrm{FV}^a_\lambda((\lambda x.P)Q).$$

2. Suppose $M \rhd_{1ct} N$ is derived by (ξ_λ), that is, there exist x, M' and N' such that $M \equiv \lambda x.M'$, $N \equiv \lambda x.N'$ and $M' \rhd_{1ct} N'$. By induction hypothesis, we have $N' \in \mathrm{TpTm}^I_{\lambda\rho}$. [♯1] is obvious. Then we show [♯2].

   Suppose $a \in \mathrm{FV}_\rho(N')$ and suppose $x \in \mathrm{FV}^a_\lambda(N')$. From theorem 3.20 and [♯1], we obtain $x \in \mathrm{FV}^a_\lambda(M')$. However, this contradicts to $M \equiv \lambda x.M' \in \mathrm{TpTm}^I_{\lambda\rho}$. Hence $x \notin \mathrm{FV}^a_\lambda(N')$. We therefore obtain $N \equiv \lambda x.N' \in \mathrm{TpTm}^I_{\lambda\rho}$.

□

3.2.3 Correspondence to intuitionistic logic

In this subsection, we prove that the intuitionistic lambda-rho-calculus corresponds to intuitionistic logic [4] . We first prepare the following concept.

Definition 3.23 (lambda-rho-context)**.** We define the set $\mathcal{C}$ of all *lambda-rho-contexts* and a mapping $\mathrm{Type}^c : \mathcal{C} \to \mathrm{Tp}_\to$ as follows (we write $C[\ ]_\varphi : \sigma$ if both $C[\ ]_\varphi \in \mathcal{C}$ and $\mathrm{Type}^c(C[\ ]_\varphi) = \sigma$ hold).

($c0$) $[\![\ ]\!]_\varphi : \varphi$.

($c1$) $C[\ ]_\varphi : \sigma,\ M \in \mathrm{TpTm}_{\lambda\rho},\ \mathrm{Type}(M) : \sigma \to \tau \implies MC[\ ]_\varphi : \tau$.

($c2$) $C[\ ]_\varphi : \sigma \to \tau,\ M \in \mathrm{TpTm}_{\lambda\rho},\ \mathrm{Type}(M) = \sigma \implies C[\ ]_\varphi M : \tau$.

($c3$) $C[\ ]_\varphi : \tau,\ x \in \mathrm{V}^\sigma_\lambda \implies \lambda x.C[\ ]_\varphi : \sigma \to \tau$.

($c4$) $C[\ ]_\varphi : \sigma,\ a \in \mathrm{V}^\sigma_\rho \implies (aC[\ ]_\varphi)^\tau : \tau$.

($c5$) $C[\ ]_\varphi : \sigma,\ a \in \mathrm{V}^\sigma_\rho \implies \rho a.C[\ ]_\varphi : \sigma$.

[4] This thesis prove the fact by use of the contraction $\rhd_{1ct}$, but a more proof-theoretic proof was given in (Matsuda, 2015a).

We use metavariables $C, D, \dots$ to stand for arbitrary contexts. Parentheses are omitted under the convention of association to the left. We obtain a typed lambda-rho-term $C[M]_\varphi$, for each M satisfying $\mathrm{Type}(M) = \varphi$ and each $C[\ ]_\varphi \in \mathcal{C}$, as follows.

$(c0)'$ $C[\ ]_\varphi \equiv [\![\ ]\!]_\varphi \implies C[M]_\varphi \equiv M$.

$(c1)'$ $C[\ ]_\varphi \equiv ND[\ ]_\varphi \implies C[M]_\varphi \equiv ND[M]_\varphi$.

$(c2)'$ $C[\ ]_\varphi \equiv D[\ ]_\varphi N \implies C[M]_\varphi \equiv D[M]_\varphi N$.

$(c3)'$ $C[\ ]_\varphi \equiv \lambda x.D[\ ]_\varphi \implies C[M]_\varphi \equiv \lambda x.D[M]_\varphi$.

$(c4)'$ $C[\ ]_\varphi \equiv (bD[\ ]_\varphi)^\tau \implies C[M]_\varphi \equiv (bD[M]_\varphi)^\tau$.

$(c5)'$ $C[\ ]_\varphi \equiv \rho b.D[\ ]_\varphi \implies C[M]_\varphi \equiv \rho b.D[M]_\varphi$.

Theorem 3.24. If $\rho a.C[(aM)^\varphi]_\varphi \in \mathrm{TpTm}^I_{\lambda\rho}$ and $\mathrm{FV}_\rho(M) = \emptyset$, then

$$\rho a.C[(aM)^\varphi]_\varphi \rhd_{ct} M.$$

Proof. It suffices to show that $C[(aM)^\varphi]_\varphi \rhd_{ct} (aM)^\psi$, here

$$\psi = \mathrm{Type}(C[(aM)^\varphi]_\varphi).$$

The proof is given by induction on the size of $C[\ ]_\varphi$.

1. $C[\ ]_\varphi \equiv [\![\ ]\!]_\varphi$. Because $\rho a.C[(aM)^\varphi]_\varphi \in \mathrm{TpTm}^I_{\lambda\rho}$, we obtain $\varphi \equiv \psi$, and then we obtain $C[(aM)^\varphi]_\varphi \equiv (aM)^\psi$.

2. Suppose that $C[\ ]_\varphi \equiv ND[aM]_\varphi$. Let $\mathrm{Type}(N) = \sigma \to \tau$. By induction hypothesis, we obtain $D[(aM)^\varphi]_\varphi \rhd_{ct} (aM)^\sigma$. Furthermore, by the rule (**throw** λ-app l), we obtain $N(aM)^\sigma \rhd_{ct} (aM)^\tau$. The cases when $C[\ ]_\varphi$ is constructed by either of the rules $(c2)$-$(c4)$ can be proved in the same way.

3. Suppose that $C[\ ]_\varphi \equiv \rho b.D[\ ]$ and $\mathrm{Type}(b) = \psi$. By induction hypothesis, we obtain $D[(aM)^\varphi]_\varphi \rhd_{ct} (aM)^\psi$. Furthermore, because $b \notin \mathrm{FV}_\rho(M)$, we obtain $\rho b.(aM)^\psi \rhd_{1ct} (aM)^\psi$.

□

Lemma 3.25. Let $M \in \mathrm{TpTm}^I_{\lambda\rho}$ be closed, then there exists a closed lambda-term N such that $M \rhd_{ct} N$.

Proof. By induction on the number of rho-symbols in M.

Suppose there are no rho-applications in M. Take a rho-abstraction $\rho a.N$ occurring in M and let $M \equiv C[\rho a.N]_\varphi$. By the assumption, we have $\mathrm{FV}_\rho(N) = \emptyset$. Hence we have $M \triangleright_{1ct} C[N]_\varphi$. By theorem 3.20, we can see that $C[N]_\varphi$ is closed. Then, by induction hypothesis, $C[N]_\varphi$ can be reduced to some closed lambda-term.

Suppose there is a rho-application in M. Then there exists a rho-application $aN\ M$ such that $\mathrm{FV}_\rho(N) = \emptyset$. Because M is closed, $M \equiv C[\rho a.D[(aN)^\psi]_\psi]_\varphi$ for some contexts $C[\ ]_\varphi, D[\ ]_\psi$. We have $M \triangleright_{ct} C[N]_\varphi$ by theorem 3.24. We can see, by theorem 3.20, $C[N]_\varphi$ is closed. Then, by induction hypothesis, $C[N]_\varphi$ can be reduced to some closed lambda-term. □

Then we show that the intuitionistic lambda-rho-calculus corresponds to the implicational fragment of intuitionistic logic.

Theorem 3.26. For each $\varphi \in \mathrm{Tp}_\to$, $\vdash_{\mathbf{HK}_\supset} \varphi$ if and only if there exists a closed term $M \in \mathrm{TpTm}^I_{\lambda\rho}$ such that $\mathrm{Type}(M) = \varphi$.

Proof. Because $\mathrm{TpTm}_\lambda \subseteq \mathrm{TpTm}^I_{\lambda\rho}$, the "only if" part is clear.

We show the "if" part. Let M be a closed term in $\mathrm{TpTm}^I_{\lambda\rho}$ such that $\mathrm{Type}(M) = \varphi$. By lemma 3.25, there exists a closed lambda-term N such that $M \triangleright_{ct} N$. Furthermore, from theorem 3.20, we obtain $\mathrm{Type}(M) = \varphi$. Therefore, φ is intuitionistically valid. □

3.2.4 Catch and throw in the intuitionistic lambda-rho-calculus

We can give the intuitionistic lambda-rho-terms which work as the **catch** operator and the **throw** operator as follows.

$$\textbf{catch}\ a\ \textbf{in}\ M \equiv \rho a.M, \quad \textbf{throw}\ N\ \textbf{to}\ a \equiv (aN)^\varphi$$

where φ is an appropriate type which depends on the context. We can easily show that if $\mathrm{Type}(a) = \sigma$ and $\mathrm{FV}_\mu(N) = \emptyset$ then, for any lambda-rho-context $C[\ ]_\tau : \sigma$ such that $\textbf{catch}\ a\ \textbf{in}\ C[\textbf{throw}\ N\ \textbf{to}\ a]_\tau \in \mathrm{TpTm}^I_{\lambda\rho}$,

$$\textbf{catch}\ a\ \textbf{in}\ C[\textbf{throw}\ N\ \textbf{to}\ a]_\tau \triangleright_{ct} N.$$

See also example 3.17.

3.2.5 Strong normalization

This subsection gives a proof of the strong normalization property of our system:

Theorem 3.27 (Strong normalization theorem)**.** For any $M \in \mathrm{TpTm}^I_{\lambda\rho}$, there are no infinite $\triangleright_{1ct}$-sequences starting from M.

Our contraction relation $\triangleright_{1ct}$ is regarded as a kind of symmetric reduction. In general, symmetric contraction rules make the strong normalization proof complicated (a detailed consideration for this topic was given in (David & Nour, 2005)). However, in our system, the proof can be given in the standard way, so-called the reducibility method [5] (see (Hindley & Seldin, 2008, Appendix A3)). It tells us that our contraction relation is easier to treat than Parigot's contraction $\triangleright_{1p}$ in some sense. In the following, we write the set of subterms of M as $\mathrm{Sub}(M)$, that is, $\mathrm{Sub}(M) = \{N \in \mathrm{TpTm}^I_{\lambda\rho} \mid N \text{ occurs in } M \text{ as a subterm}\}$. The key of our proof is the following property.

Lemma 3.28. Suppose $M \in \mathrm{TpTm}^I_{\lambda\rho}$, $M \triangleright_{ct} N$ and $(aP)^\varphi \in \mathrm{Sub}(N)$, then there exists a subterm of M which has the form $(aQ)^\psi \in \mathrm{Sub}(M)$ for some Q such that $Q \triangleright_{ct} P$.

Proof. The proof is given by induction on the length of the $\triangleright_{1ct}$-sequence from M to N. We can assume $a \notin \mathrm{BV}_\rho(M)$.

1. Suppose $M \equiv N$. Then the lemma is obvious.

2. Suppose $M \triangleright_{1ct} N$. The case when $M \triangleright_{1ct} N$ is derived without the rule (τ) is obvious. Then suppose that there exist R, S such that $M \equiv (\lambda x.R)S$ and $N \equiv [S/x]R$. In this case, because $M \in \mathrm{TpTm}^I_{\lambda\rho}$, we have $x \notin \mathrm{FV}^a_\lambda(R)$. Then we can check

$$(aP)^\varphi \in \mathrm{Sub}([S/x]R) \implies (aP)^\varphi \in \mathrm{Sub}(R) \cup \mathrm{Sub}(S)$$

   by easy induction on the size of R.

3. Suppose there exists R such that $M \triangleright_{ct} R \triangleright_{1ct} N$. Suppose $(aP)^\varphi \in \mathrm{Sub}(N)$. Then, there exists $(aS)^\tau \in \mathrm{Sub}(R)$ such that $S \triangleright_{ct} P$. In addition, by induction hypothesis, there exists $(aQ)^\psi \in \mathrm{Sub}(M)$ such that $Q \triangleright_{ct} S$. This $(aQ)^\psi$ satisfies the required condition.

$\square$

Then we start to prove the theorem.

Definition 3.29 (Computable terms)**.** We define the set SN of *strongly normalizable terms* as follows.

$$\mathrm{SN} = \{M \in \mathrm{TpTm}^I_{\lambda\rho} \mid \text{there are no infinite } \triangleright_{1ct}\text{-sequences starting from } M\}.$$

[5] Yamagata (Yamagata, 2001) showed the strong normalization property of Parigot's symmetric contraction by applying the proof method given in (Barbanera & Berardi, 1996). Their proof was given with the reducibility method. However, their definition of strong computable terms (strong reducible terms) were very complicated. In our proof, on the other hand, the strong computable terms can be defined simply.

Then we define the set $\mathrm{SC} \subseteq \mathrm{TpTm}^I_{\lambda\rho}$ of *strongly computable terms* as follows.

$M \in \mathrm{SC}$ if either of the following conditions holds.

(1) $\mathrm{Type}(M)$ is an atomic type and $M \in \mathrm{SN}$.

(2) $\mathrm{Type}(M) = \sigma \to \tau$ and $MN \in \mathrm{SC}$ for every $N \in \mathrm{SC}$ such that $\mathrm{Type}(N) = \sigma$.

Note 3.30. The following properties can be easily checked (see (Hindley & Seldin, 2008)).

(1) Suppose $\mathrm{Type}(M) = \sigma_1 \to \ldots \to \sigma_n \to t$ for some $\sigma_1, \ldots, \sigma_n \in \mathrm{Tp}_{\to}$ and $t \in \mathrm{AT}$. $M \in \mathrm{SC}$ if and only if $MN_1 \ldots N_n \in \mathrm{SN}$ for every $N_1, \ldots, N_n \in \mathrm{SC}$ such that $\mathrm{Type}(N_i) = \sigma_i$.

(2) If $M \in \mathrm{SN}$ and $M \rhd_{1ct} N$ then $N \in \mathrm{SN}$.

(3) If $M \in \mathrm{SN}$ then every subterm of M is in SN, because any infinite $\rhd_{1ct}$-sequence from a subterm of M gives rise to an infinite sequence from M.

(4) Suppose $[N/x]M \in \mathrm{SN}$. Then $M \in \mathrm{SN}$, because any infinite $\rhd_{1ct}$-sequence from M gives rise to an infinite sequence from $[N/x]M$. Furthermore, if $x \in \mathrm{FV}_\lambda(M)$ then $N \in \mathrm{SN}$.

(5) If $Mx \in \mathrm{SN}$ then $M \in \mathrm{SN}$.

(6) If $(aM)^\sigma \in \mathrm{SN}$ then $(aM)^\tau \in \mathrm{SN}$.

Lemma 3.31. Every term of the form $(aM_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} M_1 \ldots M_n$ is in SN, if $M_0, \ldots, M_n \in \mathrm{SN}$ and $(aM_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} M_1 \ldots M_n \in \mathrm{TpTm}^I_{\lambda\rho}$.

Proof. By induction on n.

First, we consider the case when $n = 0$. Suppose that there is an infinite $\rhd_{1ct}$-sequence Σ starting from $(aM_0)^\tau$. From the form of reduction rules, the form of Σ is either of the followings.

$$(aM_0)^\tau \rhd_{1ct} (aN_1)^\tau \rhd_{1ct} (aN_2)^\tau \rhd_{1ct} \ldots \qquad (M_0 \rhd_{1ct} N_1,\ N_j \rhd_{1ct} N_{j+1}).$$

$$(aM_0)^\tau \rhd_{1ct} \ldots \rhd_{1ct} (a(bP)^\varphi)^\tau \rhd_{1ct} (bP)^\tau \rhd_{1ct} \ldots \qquad (M_0 \rhd_{ct} (bP)^\varphi).$$

The former sequence contradicts the condition $M_0 \in \mathrm{SN}$. On the other hand, because $M_0 \rhd_{ct} (bP)^\varphi$ and $M_0 \in \mathrm{SN}$ and note 3.30-(6), there are no $\rhd_{1ct}$-sequences starting from $(bP)^\tau$. Hence $(aM_0)^\tau \in \mathrm{SN}$.

Let $M \equiv (aM_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} M_1 \ldots M_n$ and suppose that there is an infinite $\rhd_{1ct}$-sequence Σ starting from M. Because each M_i is in SN, there are no infinite $\rhd_{1ct}$-sequences of the following form.

$$\begin{aligned} M &\rhd_{1ct} (a_1 N_{01})^{\sigma_1 \to \ldots \to \sigma_n \to \tau} N_{11} \ldots N_{n1} \\ &\rhd_{1ct} (a_2 N_{02})^{\sigma_1 \to \ldots \to \sigma_n \to \tau} N_{12} \ldots N_{n2} \\ &\rhd_{1ct} \ldots \end{aligned}$$

$$(\exists j \text{ such that } j = k \text{ implies } M_j \rhd_{1ct} N_{j1} \text{ and } j \neq k \text{ implies } M_k \equiv N_{k1})$$
$$(\forall i, \exists j \text{ such that } j = k \text{ implies } N_{ji} \rhd_{1ct} N_{j(i+1)} \text{ and } j \neq k \text{ implies } N_{ki} \equiv N_{k(i+1)})$$

Hence, σ has either of the following forms.

$$\begin{aligned} M \rhd_{1ct} \ldots &\rhd_{1ct} (bP_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} P_1 P_2 \ldots P_n \\ &\rhd_{1ct} (bP_0)^{\sigma_2 \to \ldots \to \sigma_n \to \tau} P_2 \ldots P_n \\ &\rhd_{1ct} \ldots \end{aligned}$$
$$(M_j \rhd_{ct} P_j,\ (aM_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} \rhd_{ct} (bP_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau})$$

$$\begin{aligned} M \rhd_{1ct} \ldots &\rhd_{1ct} (cQ_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} Q_1 Q_2 \ldots Q_{i-1} (dR)^{\sigma_i} Q_{i+1} \ldots Q_n \\ &\rhd_{1ct} (dR)^{\sigma_{i+1} \to \ldots \to \sigma_n \to \tau} Q_{i+1} \ldots Q_n \\ &\rhd_{1ct} \ldots \end{aligned}$$
$$(M_i \rhd_{ct} (dR)^{\sigma_i},\ M_j \rhd_{ct} Q_j,\ (aM_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau} \rhd_{ct} (cQ_0)^{\sigma_1 \to \ldots \to \sigma_n \to \tau})$$

Here, from note 3.30-(2) and (3), we obtain $P_j, Q_k, R \in$ SN. Therefore, by induction hypothesis, there are no infinite sequences of the above forms. □

Lemma 3.32. If $M_1, \ldots, M_n \in$ SN and $xM_1 \ldots M_n \in \mathrm{TpTm}^I_{\lambda\rho}$, then $xM_1 \ldots M_n \in$ SN.

Proof. In the same way as the lemma 3.31. □

Lemma 3.33.

(1) Every term of the form $xM_1 \ldots M_n$ such that $\mathrm{Type}(xM_1 \ldots M_n) = \varphi$ is in SC, if $M_1, \ldots, M_n \in$ SN.

(2) Every term $M \in \mathrm{SC}$ such that $\mathrm{Type}(M) = \varphi$ is in SN.

Proof. We prove (1) and (2) simultaneously by induction on the size of φ.

Let $\varphi \equiv t \in \mathrm{AT}$. We obtain (1) from lemma 3.32, and obtain (2) from the definition of SC.

Let $\varphi \equiv \sigma \to \tau$. We first show (1). Take an arbitrary term N of type σ. Then, by induction hypothesis, we obtain $N \in \mathrm{SN}$ and, by induction hypothesis again, $xM_1 \ldots M_n N \in \mathrm{SC}$. We therefore obtain $xM_1 \ldots M_n \in \mathrm{SC}$. Then we show (2). Take $x \in \mathrm{V}_\lambda^\sigma$ and consider Mx. We obtain $Mx : \tau$, and obtain $Mx \in \mathrm{SN}$ by induction hypothesis. Hence, with note 3.30, $M \in \mathrm{SN}$. □

Lemma 3.34.

(1) If $([M_1/x]M_0)M_2 \ldots M_n \in \mathrm{SN}$ and $M_1 \in \mathrm{SN}$ then $(\lambda x.M_0)M_1M_2 \ldots M_n \in \mathrm{SN}$.

(2) If $([N/x]M) \in \mathrm{SC}$ and $N \in \mathrm{SC}$ then $(\lambda x.M)N \in \mathrm{SC}$.

Proof. In the same way as lemma 3.31 and (Hindley & Seldin, 2008). □

Then we show the strong normalization theorem for our system.

Proof of the strong normalization theorem. From lemma 3.33, it suffices to show the following property.

($\sharp$) For all $x_1, \ldots, x_n$ ($x_i \in \mathrm{V}_\lambda^{\varphi_i}$) and all $N_1, \ldots, N_n \in \mathrm{SC}$ ($\mathrm{Type}(N_i) = \varphi_i$) such that none of $x_1, \ldots, x_{i-1}$ occurs free in N_i, $[N_1/x_1] \ldots [N_n/x_n]M \in \mathrm{SC}$.

The proof of ($\sharp$) is given by induction on the size of $M \in \Lambda_\rho$. The case when M is neither a rho-application nor a rho-abstraction can be shown in the standard way. The case when M is a ρ-application is easily verified with lemma 3.31 and the induction hypothesis.

Suppose $M \equiv \rho a.M'$. We can assume a does not occur free in any of $N_1, \ldots, N_n$. In this case, we have

$$[N_1/x_1] \ldots [N_n/x_n]M \equiv \rho a.[N_1/x_1] \ldots [N_n/x_n]M'.$$

By induction hypothesis, $[N_1/x_1] \ldots [N_n/x_n]M' \in \mathrm{SC}$. Let $\mathrm{Type}(M) = \sigma_1 \to \ldots \to \sigma_m \to t$ ($t \in \mathrm{AT}$) and take $P_1, \ldots, P_m \in \mathrm{SC}$ such that $\mathrm{Type}(P_i) = \sigma_i$. Suppose that there is an infinite $\rhd_{1ct}$-sequence Σ starting from

$$(\rho a.[N_1/x_1] \ldots [N_n/x_n]M')P_1 \ldots P_m,$$

then Σ has the following form (we can see, in the same way as lemma 3.34, the other cases cannot happen).

$$\begin{aligned}(\rho a.[N_1/x_1]\dots[N_n/x_n]M')P_1\dots P_m \rhd_{1ct} \dots \rhd_{1ct} &(\rho a.(aQ)^{\varphi})R_1\dots R_m \\ \rhd_{1ct} &\, QR_1\dots R_m \\ \rhd_{1ct} &\dots \\ ([N_1/x_1]\dots[N_n/x_n]M' \rhd_{ct} (aQ)^{\varphi},\ P_i \rhd_{ct} R_i,\ &\varphi \equiv \sigma_1 \to \dots \to \sigma_m \to p)\end{aligned}$$

On the other hand, from lemma 3.28, there exists $(aS)^{\psi} \in \mathrm{Sub}(M')$ such that

$$[N_1/x_1]\dots[N_n/x_n]S \rhd_{ct} Q.$$

By induction hypothesis and note 3.30, we have $[N_1/x_1]\dots\ [N_n/x_n]S \in \mathrm{SC}$, but this contradicts the form of Σ. Therefore there are no infinite $\rhd_{1ct}$-sequence starting from $(\rho a.[N_1/x_1]\dots[N_n/x_n]M')P_1\dots P_m$. □

3.3 Conclusion and future work

In this chapter, we give a new typed lambda-calculus, called the intuitionistic lambda-rho-calculus, which corresponds to the implicational fragment of intuitionistic logic and has more expressive power than the lambda-beta-calculus.

We expect that we can show some properties of intuitionistic logic with the intuitionistic lambda-rho-calculus. For example, in (Matsuda, 2015b), the author show the correspondence between the intuitionistic lambda-rho-calculus and a proof system called tree sequent calculus of intuitionistic logic, and show some properties of the tree sequent calculus with the correspondence.

Bibliography

Abelson, H., Sussman, G., & Sussman, J. (1996). *Structure and Interpretation of Computer Programs* (2 edition). MIT Press.

Baba, K., Hirokawa, S., Kashima, R., Komori, Y., & Takeuti, I. (2000). Case Calculus for Classical Logic. *DOI Technical Report*, *178*.

Barbanera, F., & Berardi, S. (1996). A symmetric lambda calculus for classical program extraction. *Information and computation*, *125*(2), 103–117.

Barendregt, H. P. (1984). *The lambda calculus*. North-Holland Amsterdam.

Bierman, G. M. (1998). *A computational interpretation of the $\lambda\mu$-calculus*. Springer.

Buss, S. R. (1998). *Handbook of proof theory*. Elsevier.

Church, A. (1944). *THE CALCULI OF LAMBDA-CONVERSION*. Princeton university press.

Church, A., & Rosser, J. B. (1936). Some properties of conversion. *Transactions of the American Mathematical Society*, *39*(3), 472–482.

Curry, H. B., Feys, R., Craig, W., & Craig, W. (1958). *Combinatory logic, vol. 1*. North-Holland Publ.

David, R., & Nour, K. (2005). Why the usual candidates of reducibility do not work for the symmetric $\lambda\mu$-calculus. *Electronic Notes in Theoretical Computer Science*, *140*, 101–111.

Dehornoy, P., & van Oostrom, V. (2008). Z: Proving Confluence by Monotonic Single-Step Upperbound Functions. *Logical Models of Reasoning and Computation*.

Fujita, K. (2015). On Least Upper Bounds on the Church-Rosser Theorem. Talk in LLS 2015, Kusatsu.

Fujita, K., Kashima, R., Komori, Y., & Matsuda, N. (2015). Reduction rules for intuitionistic $\lambda\rho$-calclus. *Studia Logica*, *103*(6), 1225–1244.

Gentzen, G. (1935). Untersuchungen über das logische Schließen. I. *Mathematische zeitschrift*, *39*(1), 176–210.

Graham, P. (1996). *ANSI common lisp. Series in Artificial Intelligence.* Prentice Hall, Englewood Cliffs, New Jersey.

Griffin, T, G. (1989). A formulae-as-type notion of control. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pp. 47–58.

Gunter, C. A. (1992). *Semantics of programming languages: structures and techniques.* MIT press.

Hindley, J. R., & Seldin, J. P. (2008). *Lambda-calculus and combinators: an introduction.* Cambridge University Press Cambridge.

Howard, W. A. (1980). The formulae-as-types notion of construction. In Curry, H. B., Hindley, J. R., & Seldin, J. P. (Eds.), *To H.B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 479–490. Academic Press, New York.

Kleene, S. C. (1936). λ-definability and recursiveness. *Duke mathematical journal*, *2*(2), 340–353.

Kobayashi, S. (1997). Monad as modality. *Theoretical Computer Science*, *2175*, 29–74.

Komori, Y. (2013). $\lambda\rho$-calculus II. *Tsukuba Journal of Mathematics*, *37*, 307–320.

Komori, Y., Matsuda, N., & Yamakawa, F. (2014). A Simplified Proof of the Church-Rosser Theorem. *Studia Logica*, *102*(1), 175–183.

Komori, Y., & Ono, H. (2010). *Gendaisuurironrigakujosetsu.* Nihonhyouronsha. in Japanese.

Matsuda, N. (2015a). Intuitionistic fragment of the $\lambda\mu$-calculus. Tech. rep. C-282, Tokyo Institute of Technology.

Matsuda, N. (2015b). Intuitionistic tree sequent calculus and intuitionistic lambda-rho-calculus. In *Post-proceedings of the RIMS workshop "proof theory, computability theory and related issues"*.

Matsuda, N. (2015c). A simple extension of the Curry-Howard correspondence with intuitionistic lambda rho calculus. In *International Workshop on Rewriting Techniques for Program Transformations and Evaluation (WPTE)*, pp. 21–32.

Miyamoto, K., & Igarashi, A. (2004). A modal foundation for secure information flow. In *Workshop on Foundations of Computer Security*, pp. 187–203.

Nakazawa, K., & Fujita, K. (2015). Compositional Z: Confuluence Proofs for Permutative Conversion. In *Proceedings of te 32th conference of Japan society for software science and technology.*

Nakazawa, K., & Katsumata, S. (2012). Extensional models of untyped Lambda-mu calculus. *arXiv preprint arXiv:1210.3116.*

Nakazawa, K., & Nagai, T. (2014). Reduction system for extensional lambda-mu calculus. In *Rewriting and Typed Lambda Calculi*, pp. 349–363. Springer.

Nakazawa, K., & Naya, H. (2015). Strong Reduction of Combinatory Calculus with Streams. *Studia Logica*, *103*(2), 375–387.

Parigot, M. (1992). $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Logic programming and automated reasoning*, pp. 190–201. Springer.

Parigot, M. (1993). Classical proofs as programs. In *Computational logic and proof theory*, pp. 263–276. Springer.

Parigot, M. (2000). On the computational interpretation of negation. In *Computer Science Logic*, pp. 472–484. Springer.

Pierce, B. C. (2002). *Types and programming languages.* MIT press.

Prawitz, D. (1965). *NATURAL DEDUCTION: A Proof-Theoretical Study.* Almqvist and Wiksell.

Saurin, A. (2005). Separation with streams in the $\Lambda\mu$-calculus. In *Logic in Computer Science, LICS 2005*, pp. 356–365. IEEE.

Sørensen, M. H., & Urzyczyn, P. (2006). *Lectures on the Curry-Howard isomorphism.* Elsevier.

Takahashi, M. (1989). Parallel reductions in λ-calculus. *Journal of Symbolic Computation*, *7*(2), 113–123.

Takahashi, M. (1991). *Keisanron: Keisankanouseitoramudakeisan.* Gendaikagakusha. in Japanese.

Takahashi, M. (1995). Parallel reductions in λ-calculus. *Information and computation*, *118*(1), 120–127.

Yamagata, Y. (2001). Strong normalization of second order symmetric lambda-mu calculus. In *Theoretical Aspects of Computer Software*, pp. 459–467. Springer.

Yamakawa, F., & Komori, Y. (2015). An extension to predicate logic of $\lambda\rho$-calculus. In *Proceedings of the RIMS workshop "Logics, Algebras and Languages in Computer Science"*.