千葉大学大学院　自然科学研究科

修士論文

ラムダ項の β 正規形を結合子を用いて求める方法

平成 18 年 3 月

数学・情報数理学 専攻
情報数理学 講座

内藤昭宏

目次

1 はじめに

与えられたラムダ項 M に対して、その正規形 M^* を手動で求めるというのは結構面倒なことである。計算機を使って求める際にも代入を行う時の α 変換などに時間を要するために効率を上げることが難しい。
そこで本論分では、β 正規形を早く求めることを目標に以下の方法についての検証を行ってみた。

1. ラムダ項を一度結合子を用いてラムダを消去した項に変換する。
2. さらにそれについて変換規則を用いてそれ以上変形できない形にする。
3. これを再びラムダ項に変換してから β リダクションを行い、β 正規形を求める。

この方法は結合子の変換規則を用いることによって、β リダクションの回数を減らすことにより速度を早める事を目的としている。
またこの方法を用いて求まったものが M^* と $=_\alpha$ になっていることの証明も行ってみた。

なおこの論文は [1] を参考に古森雄一先生より詳細な方針を与えられ、それらの検討・検証を内藤が行った。

2 CLt- 項

CLt- 項というものを以下のように帰納的に定義する。定数記号には $\mathsf{I}, \mathsf{K}', \mathsf{B}', \mathsf{C}', \mathsf{S}'$ が含まれているものとする。

定義 2.1 (CLt- 項).

1. 変数と定数記号は CLt- 項 である (この形の CLt- 項を**原子** という).
2. M と N が CLt- 項 ならば, (MN) は CLt- 項 である (この形の CLt- 項を**適用** という).

定義通りに書くと括弧が多くなって読みづらくなってしまうので適当に省略する事とする。そして、足りない括弧は左から補うことにする。例えば $\mathsf{S}'xy\mathsf{C}'\mathsf{K}'$ は $((((\mathsf{S}'x)y)\mathsf{C}')\mathsf{K}')$ を表している。

ここでは $\mathsf{I}, \mathsf{K}', \mathsf{B}', \mathsf{C}', \mathsf{S}'$ を基本的な結合子とし CLt- 項の変換規則を以下のように定義する。

定義 2.2 (変換規則).

$$(\mathsf{I})\ \ \mathsf{I}M \rhd_{1wt} M, \quad (\mathsf{K}')\ \ \mathsf{K}'MNRU \rhd_{1wt} MNR, \quad (\mathsf{B}')\ \ \mathsf{B}'MNRU \rhd_{1wt} MN(RU),$$

$$(\mathsf{C}')\ \ \mathsf{C}'MNRU \rhd_{1wt} M(NU)R, \quad (\mathsf{S}')\ \ \mathsf{S}'MNRU \rhd_{1wt} M(NU)(RU)\ .$$

CLt- 項 M に対して上の変換規則を用いて有限回の変形を行い、それ以上変形できなくなったその最終結果を M の正規形と呼び $\overline{M}$ と表すこととする。

3 ラムダ項

ラムダ項を以下のように再帰的に定義する。

定義 3.1 (ラムダ項).

1. 変数は ラムダ項 である (この形の ラムダ項を**原子** という).
2. M と N が ラムダ項 ならば, (MN) は ラムダ項 である (この形の ラムダ項を**適用**という).
3. x が変数 M がラムダ項の時 $(\lambda x.M)$ はラムダ項である (この形のラムダ項を**抽象**という).

CLt- 項の時と同様に括弧を省略し、足りない括弧は左から補うこととする。ただし抽象については一番外側の括弧となっているときだけ省略する。すなわち $(\lambda x.M)$ は $\lambda x.M$ と省略するが,$((\lambda x.M)N)$ を $\lambda x.MN$ とはしない $\lambda x.MN$ は $(\lambda x.(MN))$ の省略形である. また $(\lambda x.(\lambda y.(\lambda z.M)))$ などは $\lambda xyz.M$ と省略する.

ラムダ項 M に含まれる自由変数の集合を $FV(M)$ とする。

定義 3.2 (代入 (substitution)). M, N をラムダ項, x を変数とする. M の中の (自由に現れる) x に N を代入した結果 $[N/x]M$ を次のように帰納的に定義する.

1. $[N/x]x \equiv N$,
2. $[N/x]M \equiv M \quad (x \notin FV(M)$ のとき$)$,
3. $[N/x](QR) \equiv ([N/x]Q\ [N/x]R) \quad (x \in FV(QR)$ のとき$)$,
4. $[N/x](\lambda y.Q) \equiv \lambda y.[N/x]Q \quad (x \in FV(Q)$ かつ $y \notin FV(N)$ のとき$)$,
5. $[N/x](\lambda y.Q) \equiv \lambda z.[N/x][z/y]Q \quad (x \in FV(Q)$ かつ $y \in FV(N)$ のとき$)$.

ここで $y \not\equiv x$ は仮定されている. また, z は 今までに現れていない変数である

定義 3.3 (β 変換).
ラムダ項の β 変換 $\rhd_{1\beta}$ を以下のように再帰的に定義する。

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

2. $M \rhd_{1\beta} N$ ならば、$\lambda x.M \rhd_{1\beta} \lambda x.N, \qquad MP \rhd_{1\beta} NP, \qquad PM \rhd_{1\beta} PN.$

β 変換の基になる $(\lambda x.M)N$ の形のラムダ項を β -redex とよぶ.

β 変換を有限回繰り返し実行して P から Q が得られるとき
$P \rhd_{\beta} Q$ とかく. また β 変換可能なラムダ項同士を同一視することによって得られるラムダ項間の等号関係を $=_{\beta}$ で表す. すなわちラムダ項 P, Q に対して

$$P \equiv P_1 \leftrightarrow_{\beta} P_2 \leftrightarrow_{\beta} \cdots \leftrightarrow_{\beta} P_n \equiv Q$$
$$(\text{ただし } M \leftrightarrow_{\beta} N \Longleftrightarrow_{def} (M \rhd_{1\beta} N \text{または} N \rhd_{1\beta} M))$$
$$\text{を満たす } n \geq 1 \text{ と } P_1, P_2, \cdots, P_n \text{ がある時、} P =_{\beta} Q \text{ とかく}$$

ラムダ項 M が β-redex を含まないとき M は β 正規形であるという. また $M \rhd_{\beta} N$ を満たす β 正規形 N がある時 M は β 正規形 N をもつ、あるいは N は M の β 正規形であるという.

定理 3.4 (Church-Rosser の定理). $M \rhd_{\beta} N$ かつ $M \rhd_{\beta} R$ ならば次のような項 T が存在する.
$N \rhd_{\beta} T$ かつ $R \rhd_{\beta} T$.

4 ラムダ項から CLt- 項への変換

定義 4.1 (変換 $\circ$). M をラムダ項とする。このとき CLt- 項への変換 M° を以下のように帰納的に定義する。

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

2. $(MN)^{\circ} \equiv M^{\circ}N^{\circ}$

3. $(\lambda x.M)^{\circ} \equiv \lambda x^{t}.M^{\circ}$

また $\lambda x^{t}.M$ を以下のように帰納的に定義する。

定義 4.2 (変換 t). M を CLt- 項, x を変数とする. 変数 x を含まない項 $(\lambda^{t}x.M)$ を以下のように帰納的に定義する.

1. $(\lambda^t x.x) \equiv \mathsf{I}$,
2. $(\lambda^t x.MUV) \equiv \mathsf{K}'MUV \quad (M$ が閉項かつ $x \notin FV(UV)$ のとき),
3. $(\lambda^t x.MUV) \equiv \mathsf{B}'MU(\lambda^t x.V) \quad (M$ が閉項かつ $x \notin FV(U)$ のとき),
4. $(\lambda^t .MUV) \equiv \mathsf{C}'M(\lambda^t x.U)V) \quad (M$ が閉項かつ $x \notin FV(V)$ のとき),
5. $(\lambda^t x.MUV) \equiv \mathsf{S}'M(\lambda^t x.U)(\lambda^t x.V) \quad (M_1$ が閉項かつ $x \in FV(U)$かつ$x \in FV(V))$ の時),
6. $(\lambda^t x.MN) \equiv (\lambda^t x.IMN)$,
7. $(\lambda^t x.M) \equiv (\lambda^t x.IIM)$.

この変換 t は David Turner によって発見された変換を少し変更したものである（[2] 参照）
ほかの変換方法では $\lambda x_1, \ldots, x_n.M$ を変換したときの変換後の項の大きさが最悪の場合 n^2 に比例した大きさになってしまうのに対し、この変換は最悪の場合でも n に比例した大きさで抑えることができる。

例 4.3 (変換例). 実際に $\lambda xyz.xz(yz)$ を CL-t 項に変換してみる。

$$
\begin{aligned}
\lambda^t z.xz(yz) &\equiv \lambda^t z.\mathsf{I}(xz)(yz) \\
&\equiv \mathsf{S}'\mathsf{I}(\lambda^t z.xz)(\lambda^t z.yz) \\
&\equiv \mathsf{S}'\mathsf{I}(\lambda^t z.\mathsf{I}xz)(\lambda^t z.\mathsf{I}yz) \\
&\equiv \mathsf{S}'\mathsf{I}(\mathsf{B}'\mathsf{I}x(\lambda^t z.z))(\mathsf{B}'\mathsf{I}y(\lambda^t z.z)) \\
&\equiv \mathsf{S}'\mathsf{I}(\mathsf{B}'\mathsf{I}x\mathsf{I})(\mathsf{B}'\mathsf{I}y\mathsf{I})
\end{aligned}
$$

$$
\begin{aligned}
\lambda^t yz.xz(yz) &\equiv \lambda^t y.\mathsf{S}'\mathsf{I}(\mathsf{B}'\mathsf{I}x\mathsf{I})(\mathsf{B}'\mathsf{I}y\mathsf{I}) \\
&\equiv \mathsf{B}'(\mathsf{S}'\mathsf{I})(\mathsf{B}'\mathsf{I}x\mathsf{I})(\lambda^t y.\mathsf{B}'\mathsf{I}y\mathsf{I}) \\
&\equiv \mathsf{B}'(\mathsf{S}'\mathsf{I})(\mathsf{B}'\mathsf{I}x\mathsf{I})(\mathsf{C}'(\mathsf{B}'\mathsf{I})\mathsf{I}\mathsf{I})
\end{aligned}
$$

$$
\begin{aligned}
\lambda^t xyz.xz(yz) &\equiv \lambda^t x.\mathsf{B}'(\mathsf{S}'\mathsf{I})(\mathsf{B}'\mathsf{I}x\mathsf{I})(\mathsf{C}'(\mathsf{B}'\mathsf{I})\mathsf{I}\mathsf{I}) \\
&\equiv \mathsf{C}'(\mathsf{B}'(\mathsf{S}'\mathsf{I}))(\lambda^t x.\mathsf{B}'\mathsf{I}x\mathsf{I})(\mathsf{C}'(\mathsf{B}'\mathsf{I})\mathsf{I}\mathsf{I}) \\
&\equiv \mathsf{C}'(\mathsf{B}'(\mathsf{S}'\mathsf{I}))(\mathsf{C}'(\mathsf{B}'\mathsf{I})(\mathsf{C}'(\mathsf{B}'\mathsf{I})))
\end{aligned}
$$

よって $(\lambda xyz.xz(yz))^\circ \equiv \mathsf{C}'(\mathsf{B}'(\mathsf{S}'\mathsf{I}))(\mathsf{C}'(\mathsf{B}'\mathsf{I})(\mathsf{C}'(\mathsf{B}'\mathsf{I})))$ となる。

5 CLt- 項からラムダ項への変換

定義 5.1 (CLt- 項からラムダ項への変換). *CLt-* 項からラムダ項への変換 $\sharp$ を以下のように帰納的に定義する。

$x^\sharp \equiv x$

$\mathsf{I}^\sharp \equiv \lambda x.x$

$\mathsf{K}'^\sharp \equiv \lambda xyzu.xyz$

$\mathsf{B}'^\sharp \equiv \lambda xyzu.xy(zu)$

$\mathsf{C}'^\sharp \equiv \lambda xyzu.x(yu)z$

$\mathsf{S}'^\sharp \equiv \lambda xyzu.x(yu)(zu)$

$(MN)^\sharp \equiv M^\sharp N^\sharp$

6 $=_\alpha$ であることの証明

ここではラムダ項 M に対し、この方法を用いて求まったものが、その正規形 M^* と β 同値であることの証明を行う。まずは準備として以下の補題を示す。
注意として、これ以降の証明では S' 等の基本結合子をラムダ項に変換した際に現れる変数（例えば S' を変換した $\lambda xyzu.x(yu)(zu)$ の $xyzu$）はすべて今までに現れてない変数 (new-variable) であるとする。

補題 6.1. 任意の *CLt-* 項 M に対し

$$(\lambda^t x.M)^\sharp x =_\beta M^\sharp$$

が成り立つ。

Proof. M に関する帰納法で示す。

■ $M \equiv x$ の時

$$\begin{array}{rcl} (\lambda^t x.x)^\sharp x & \equiv & \mathsf{I}^\sharp x \\ & \equiv & (\lambda x.x)x \\ & \rhd_{1\beta} & x \end{array}$$

■ $M \equiv M_1M_2M_3$ (M_1 が閉項かつ $x \notin FV(M_2M_3)$) の時

$$\begin{array}{rcl} (\lambda^t x.M_1M_2M_3)^\sharp x & \equiv & (\mathsf{K}'M_1M_2M_3)^\sharp x \\ & \equiv & (\lambda xyzu.xyz)M_1^\sharp M_2^\sharp M_3^\sharp x \\ & \rhd_{1\beta} & M_1^\sharp M_2^\sharp M_3^\sharp \\ & \equiv & (M_1M_2M_3)^\sharp \end{array}$$

■ $M \equiv M_1 M_2 M_3$ (M_1 が閉項かつ $x \notin FV(M_2)$) の時

$$\begin{array}{rcl}(\lambda^t x.M_1 M_2 M_3)^\sharp x & \equiv & (\mathsf{B}' M_1 M_2 (\lambda^t x.M_3))^\sharp x \\ & \equiv & (\lambda xyzu.xy(zu)) M_1^\sharp M_2^\sharp (\lambda^t x.M_3)^\sharp x \\ & \rhd_{1\beta} & M_1^\sharp M_2^\sharp ((\lambda^t x.M_3)^\sharp x)\end{array}$$

帰納法の仮定から $(\lambda^t x.M_3)^\sharp x =_\beta M_3^\sharp$ 、よって

$$\begin{array}{rl} =_\beta & M_1^\sharp M_2^\sharp M_3^\sharp \\ \equiv & (M_1 M_2 M_3)^\sharp\end{array}$$

■ $M \equiv M_1 M_2 M_3$ (M_1 が閉項かつ $x \notin FV(M_3)$) の時
上と同様に

$$\begin{array}{rcl}(\lambda^t x.M_1 M_2 M_3)^\sharp x & \equiv & (\mathsf{C}' M_1 (\lambda^t x.M_2) M_3)^\sharp x \\ & \equiv & (\lambda xyzu.x(yu)z) M_1^\sharp (\lambda^t x.M_2) M_2^\sharp M_3^\sharp x \\ & \rhd_{1\beta} & M_1^\sharp ((\lambda^t x.M_2)^\sharp x) M_3^\sharp \\ & =_\beta & M_1^\sharp M_2^\sharp M_3^\sharp \\ & \equiv & (M_1 M_2 M_3)^\sharp\end{array}$$

■ $M \equiv M_1 M_2 M_3$ (M_1 が閉項かつ $x \in FV(M_2)$かつ$x \in FV(M_3)$) の時
上と同様に

$$\begin{array}{rcl}(\lambda^t x.M_1 M_2 M_3)^\sharp x & \equiv & (\mathsf{S}' M_1 (\lambda^t x.M_2)(\lambda^t x.M_3))^\sharp x \\ & \equiv & (\lambda xyzu.x(yu)(zu)) M_1^\sharp (\lambda^t x.M_2)^\sharp (\lambda^t x.M_3)^\sharp x \\ & \rhd_{1\beta} & M_1^\sharp ((\lambda^t x.M_2)^\sharp x)((\lambda^t x.M_3)^\sharp x) \\ & =_\beta & M_1^\sharp M_2^\sharp M_3^\sharp \\ & \equiv & (M_1 M_2 M_3)^\sharp\end{array}$$

■ $M \equiv M_1 M_2$ の時

$$(\lambda^t x.M_1 M_2)^\sharp x \quad \equiv \quad (\lambda^t x.\mathsf{I} M_1 M_2)^\sharp x$$

I は閉項なので、上のいずれかと同じ議論をすればよい。

■ $M \equiv M_1$ の時

$$(\lambda^t x.M_1)^\sharp x \quad \equiv \quad (\lambda^t x.\mathsf{II} M_1)^\sharp x$$

I は閉項なので、上のいずれかと同じ議論をすればよい。

$\square$

定理 6.2. 任意のラムダ項 M に対し

$$(M^\circ)^\sharp =_\beta M$$

が成り立つ。

Proof. M に関する帰納法で示す。

■ $M \equiv x$ の時

$$\begin{array}{rcl} (x^\circ)^\sharp & \equiv & x^\sharp \\ & \equiv & x \end{array}$$

■ $M \equiv M_1 M_2$の時

帰納法の仮定から

$$\begin{array}{rcl} (M_1^\circ)^\sharp & \equiv & M_1 \\ (M_2^\circ)^\sharp & \equiv & M_2 \end{array}$$

よって

$$\begin{array}{rcl} ((M_1 M_2)^\circ)^\sharp & \equiv & (M_1^\circ M_2^\circ)^\sharp \\ & \equiv & (M_1^\circ)^\sharp (M_2^\circ)^\sharp \\ & \equiv & M_1 M_2 \end{array}$$

■ $M \equiv \lambda x.N$ の時

$$\begin{array}{rcl} M^\circ & \equiv & (\lambda x.N)^\circ \\ & \equiv & \lambda^t x.N^\circ \end{array}$$

□ $N^\circ \equiv x$ の時

$$\begin{array}{rcl} (\lambda^t x.x)^\sharp & \equiv & \mathsf{I}^\sharp \\ & \equiv & \lambda x.x \end{array}$$

□ $N^\circ \equiv N_1 N_2 N_3$ (N_1 が閉項かつ $x \notin FV(M_2 M_3)$) の時

$$\begin{array}{rcl}
(\lambda^t x.N_1N_2N_3)^\sharp & \equiv & (\mathsf{K}'N_1N_2N_3)^\sharp \\
& \equiv & (\lambda xyzu.xyz)N_1^\sharp N_2^\sharp N_3^\sharp \\
& \rhd_{1\beta} & \lambda u.N_1^\sharp N_2^\sharp N_3^\sharp \\
& =_\alpha & \lambda x.N_1^\sharp N_2^\sharp N_3^\sharp \quad (\text{u は new-variable なので } N_1,N_2,N_3 \text{ には出現しない}) \\
& \equiv & \lambda x.(N_1N_2N_3)^\sharp \\
& \equiv & \lambda x.(N^\circ)^\sharp \\
& =_\beta & \lambda x.N \quad (\text{帰納法の仮定より})
\end{array}$$

□ $N^\circ \equiv N_1N_2N_3$ (N_1 が閉項かつ $x \notin FV(N_2)$) の時

$$\begin{array}{rcl}
(\lambda^t x.N_1N_2N_3)^\sharp & \equiv & (\mathsf{B}'N_1N_2(\lambda^t x.N_3))^\sharp \\
& \equiv & (\lambda xyzu.xy(zu))N_1^\sharp N_2^\sharp(\lambda^t x.N_3)^\sharp \\
& \rhd_{1\beta} & \lambda u.N_1^\sharp N_2^\sharp((\lambda^t x.N_3)^\sharp u) \\
& =_\alpha & \lambda x.N_1^\sharp N_2^\sharp((\lambda^t x.N_3)^\sharp x) \\
& =_\beta & \lambda x.N_1^\sharp N_2^\sharp N_3^\sharp \quad (\text{補題 6.1より}) \\
& \equiv & \lambda x.(N_1N_2N_3)^\sharp \\
& \equiv & \lambda x.(N^\circ)^\sharp \\
& =_\beta & \lambda x.N \quad (\text{帰納法の仮定より})
\end{array}$$

□ $N^\circ \equiv N_1N_2N_3$ (N_1 が閉項かつ $x \notin FV(N_3)$) の時
上と同様に

$$\begin{array}{rcl}
(\lambda^t x.N_1N_2N_3)^\sharp & \equiv & (\mathsf{C}'N_1(\lambda^t x.N_2)N_3)^\sharp \\
& \equiv & (\lambda xyzu.x(yu)z)N_1^\sharp(\lambda^t x.N_2)^\sharp N_3^\sharp \\
& \rhd_{1\beta} & \lambda u.N_1^\sharp((\lambda^t x.N_2)^\sharp u)N_3^\sharp \\
& =_\alpha & \lambda x.N_1^\sharp((\lambda^t x.N_2)^\sharp x)N_3^\sharp \\
& =_\beta & \lambda x.N_1^\sharp N_2^\sharp N_3^\sharp \\
& \equiv & \lambda x.(N_1N_2N_3)^\sharp \\
& \equiv & \lambda x.(N^\circ)^\sharp \\
& =_\beta & \lambda x.N
\end{array}$$

□ $N^\circ \equiv N_1N_2N_3$ (M_1 が閉項かつ $x \in FV(N_2)$かつ$x \in FV(N_3)$) の時
上と同様に

$$
\begin{aligned}
(\lambda^t x.N_1 N_2 N_3)^\sharp &\equiv (\mathsf{S}' N_1(\lambda^t x.N_2)(\lambda^t x.N_3)^\sharp \\
&\equiv (\lambda xyzu.x(yu)(zu))N_1^\sharp(\lambda^t x.N_2)^\sharp(\lambda^t x.N_3)^\sharp \\
&\rhd_{1\beta} \lambda u.N_1^\sharp((\lambda^t x.N_2)^\sharp u)((\lambda^t x.N_3)^\sharp u) \\
&=_\alpha \lambda x.N_1^\sharp((\lambda^t x.N_2)^\sharp x)((\lambda^t x.N_3)^\sharp x) \\
&=_\beta \lambda x.N_1^\sharp N_2^\sharp N_3^\sharp \\
&\equiv \lambda x.(N_1 N_2 N_3)^\sharp \\
&\equiv \lambda x.(N^\circ)^\sharp \\
&=_\beta \lambda x.N
\end{aligned}
$$

□ $N^\circ \equiv N_1 N_2$ の時

$$
(\lambda^t x.N_1 N_2)^\sharp \equiv (\lambda^t x.\mathsf{I} N_1 N_2)^\sharp
$$

I は閉項なので上のいずれかと同じ議論をすればよい。

□ $N^\circ \equiv N_1$ の時

$$
(\lambda^t x.N_1)^\sharp \equiv (\lambda^t x.\mathsf{II} N_1)^\sharp
$$

I は閉項なので上のいずれかと同じ議論をすればよい。

□

定理 6.3. *CLt-* 項 M,N に対し $M \rhd_{1wt} N \Rightarrow M^\sharp \rhd_\beta N^\sharp$

Proof. (略証)

$M \rhd_{1wt} N$ について使われた規則が (S′) の時を考える. このとき M は $\mathsf{S}' M_1 M_2 M_3 M_4$ を含んでいる. つまり

$$
M \equiv \cdots(\cdots(\mathsf{S}' M_1 M_2 M_3 M_4 \cdots)\cdots)\cdots
$$

とあらわせる. よって N はこれを変換したもの、つまり

$$
N \equiv \cdots(\cdots(M_1(M_2 M_4)(M_3 M_4)\cdots)\cdots)\cdots
$$

とあらわせる. よって

$$
\begin{array}{rcl}
M^\sharp & \equiv & \cdots^\sharp(\cdots^\sharp((\mathsf{S}')^\sharp M_1^\sharp M_2^\sharp M_3^\sharp M_4^\sharp \cdots^\sharp)\cdots^\sharp)\cdots^\sharp \\
& \equiv & \cdots^\sharp(\cdots^\sharp((\lambda xyzu.x(yu)(zu))M_1^\sharp M_2^\sharp M_3^\sharp M_4^\sharp \cdots^\sharp)\cdots^\sharp)\cdots^\sharp \\
& \rhd_\beta & \cdots^\sharp(\cdots^\sharp(M_1^\sharp(M_2^\sharp M_4^\sharp)(M_3^\sharp M_4^\sharp)\cdots^\sharp)\cdots^\sharp)\cdots^\sharp \\
& \equiv & N^\sharp
\end{array}
$$

使われた変換規則がほかの場合も同様である.

□

定理 6.2, 定理 6.3とラムダ項のチャーチ · ロッサー性から以下のことが成り立つ。

定理 6.4. ラムダ項 M に対し

$$M^* =_\alpha ((\overline{M^\circ})^\sharp)^*$$

が成り立つ。

7　β 正規形の求まる速度

上記の方法を用いて β 正規形を求めるとき、実際に普通に求めるのと比較してどれくらい差があるのか、上記の方法を行う C プログラムを作成して検証してみた. そのプログラム及びマニュアルは以下のページにのせておく予定である.

http://www.math.s.chiba-u.ac.jp/~04um0304/

このプログラムではラムダ項及び CLt- 項を tree 型のグラフとして扱っている. また CLT-項に変換して求める場合の最後の β- リダクションについては、直接求める場合と同じ関数を用いて計算している. また代入の時は必ず α 変換を行うように作っており、代入を後回しにしたりせずに最後まで行ってから次に進むようにしてある.

例 7.1 (実行例 1). $(\lambda xyz.xz(yz))(\lambda xy.y)(uu)$ の β 正規形を求めてみる

以下プログラムの入出力である.

> (lambda x y z . x z (y z))(lambda x y . y)(u u);

——————-kekka————————-

(lambda x . u u x)

0 SEC

chokusetu:(lambda x . u u x)

0 SEC

——————————————————-

—*kekka*— 以降がプログラムの出力である。また SEC は β 正規形を求めるまでにかかった時間であり、 CLt- 項になおして求める方法は CLt 項に変換を行うところから β 正規形を求めるまでにかかった時間である．また小数点以下は切り捨ててある. 確かに直接求める方法と $=_\alpha$ となっていることがわかる.
またこのラムダ項には η *-redex* が含まれているが η リダクションは行われていないこともわかる.
なお η リダクションとは以下のようなものであり余分な抽象を消し去る意味を持つ

$$(\lambda x.Mx) \quad \rhd_\eta \quad M(x \notin FV(M))$$

例 7.2 (ラムダ項における２進計算の足し算). 次の検証に使用するものは 2004 年度の修士論文「ラムダ計算における２進計算」 *([3] 参照)* より、加算を行うラムダ項を、このラムダ項が正しい入力に対し正しく動くかどうかという確認も込めて、使用させていただきました. 以下に簡単にその定義を紹介します

基本的なラムダ項と整数 n の２進表記をラムダ項で表したものは以下のように定義する.

定義 7.3 (真偽値). ラムダ項 T, F を次のように定める。

$$\begin{array}{rcl} T & \equiv & \lambda xy.x \\ F & \equiv & \lambda xy.y \end{array}$$

T, F をこのように定めることで、ラムダ項 LMN に対して、 L が T の場合は M、 L が F の場合は N とすることができる。

定義 7.4 (1 桁の数). 整数 $1, 0, -1$ に対応するラムダ項 $\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ を次のように定める。

$$\begin{array}{rcl} \widetilde{1}^* & \equiv & \lambda xyz.x \\ \widetilde{0}^* & \equiv & \lambda xyz.y \\ \widetilde{-1}^* & \equiv & \lambda xyz.z \end{array}$$

定義 7.5 (数の表現). 整数 n が次のように表せるとする。

$$n = 2^i * x_i + 2^{i-1} * x_{i-1} + 2^{i-2} * x_{i-2} + \ldots + 2^0 * x_0 \quad i \geq 0$$
$$x_j \ (0 \leq j \leq i) \text{ は } 1 \text{ または } 0 \text{ または } -1$$

この整数 n に対応するラムダ項 $\widetilde{n}$ を次のように定める。

$$\widetilde{n} \quad \equiv \quad < \cdots <<< \widetilde{0}, \widetilde{x_i} >, \widetilde{x_{i-1}} >, \widetilde{x_{i-2}} >, \cdots >, \widetilde{x_0} >$$

ただし、x_k $(0 \leq k \leq i)$ から x_i には連続して $\widetilde{0}$ が現れないこと、また x_k には $\widetilde{1}$ と $\widetilde{-1}$ が同時には現れないこととする。また $< M, N >$ は対を表しており以下のものの省略表現として用いている.

$$Pair \stackrel{\mathrm{def}}{=} \lambda x.xMN \quad M, N\text{はラムダ項}$$

定義 7.6. $\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ の場合は $\widetilde{0}^*$、$\widetilde{n} =< M, N >$ の場合は M を返すようなラムダ項 Hdr を次のように定める。

$$Hdr \quad \stackrel{\mathrm{def}}{=} \quad \lambda x.(Isn\, x) \quad \widetilde{0}^* \quad (xK)$$

また、$\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ の場合はそのまま、$\widetilde{n} =< M, N >$ の場合は N を返すようなラムダ項 Ftr を次のように定める。

$$Ftr \quad \stackrel{\mathrm{def}}{=} \quad \lambda x.(Isn\, x) \quad x \quad (x(KI))$$

定理 7.7. $\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ の場合には T、対の場合には F を返すようなラムダ項 Isn は次のように定められる。

$$Isn = \lambda x.xK_3^3KKIFT$$
$$x\text{には任意のラムダ項を入力。}$$

ただし $K_3^3 \equiv \lambda xyz.z$ である.

定義 7.8. *1* 桁の数同士を加算した結果、繰り上がりの値を返すラムダ項 Add_1 と、繰り上がりを切り捨てた値を返すラムダ項 Add_2 を以下のように定める。

$$\begin{array}{rcllll} Add_1 & \stackrel{\mathrm{def}}{=} & \lambda xyz.x\,(y & (z\,\widetilde{1}^*\,\widetilde{1}^*\,\widetilde{0}^*) & (z\,\widetilde{1}^*\,\widetilde{0}^*\,\widetilde{0}^*) & \widetilde{0}^*) \\ & & (y & (z\,\widetilde{1}^*\,\widetilde{0}^*\,\widetilde{0}^*) & \widetilde{0}^* & (z\,\widetilde{0}^*\,\widetilde{0}^*\,\widetilde{-1}^*)) \\ & & (y & \widetilde{0}^* & (z\,\widetilde{0}^*\,\widetilde{0}^*\,\widetilde{-1}^*) & (z\,\widetilde{0}^*\,\widetilde{-1}^*\,\widetilde{-1}^*)) \end{array}$$

$$\begin{array}{rcllll} Add_2 & \stackrel{\mathrm{def}}{=} & \lambda xyz.x\,(y & (z\,\widetilde{1}^*\,\widetilde{0}^*\,\widetilde{1}^*) & (z\,\widetilde{0}^*\,\widetilde{1}^*\,\widetilde{0}^*) & z) \\ & & (y & (z\,\widetilde{0}^*\,\widetilde{1}^*\,\widetilde{0}^*) & z & (z\,\widetilde{0}^*\,\widetilde{-1}^*\,\widetilde{0}^*)) \\ & & (y & z & (z\,\widetilde{0}^*\,\widetilde{-1}^*\,\widetilde{0}^*) & (z\,\widetilde{-1}^*\,\widetilde{0}^*\,\widetilde{-1}^*)) \end{array}$$

なお上において、x は繰り上がり、y, z は加算する数をあらわし、それぞれ $\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ を入力。

これらの定義を用いて、ラムダ項 $\widetilde{n_1}, \widetilde{n_2}$ を加算するラムダ項を再帰的に定義する。

定義 7.9. 繰り上がりおよび *2* つのラムダ項 $\widetilde{n_1}, \widetilde{n_2}$ を加算した結果を返すラムダ項 Add^* を次のように定める。

$$
\begin{array}{l}
Add^* = \lambda xyz. \\
\quad (Isn\ y) \\
\quad\quad ((Isn\ z) \\
\quad\quad\quad (Pair\ (Add_1\ x\ y\ z)\ (Add_2\ x\ y\ z)) \\
\quad\quad\quad (Pair\ (Add^*\ (Add_1\ x\ (Ftr\ y)\ (Ftr\ z))\ (Hdr\ y)\ (Hdr\ z)) \\
\quad\quad\quad\quad (Add_2\ x\ (Ftr\ y)\ (Ftr\ z)))) \\
\quad\quad (Pair\ (Add^*\ (Add_1\ x\ (Ftr\ y)\ (Ftr\ z))\ (Hdr\ y)\ (Hdr\ z)) \\
\quad\quad\quad (Add_2\ x\ (Ftr\ y)\ (Ftr\ z)))
\end{array}
$$

x は繰り上がりを表すラムダ項 $\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ で、 y, z はラムダ項 $\widetilde{n_1}, \widetilde{n_2}$。

定義 7.10. *2* つのラムダ項 $\widetilde{n_1}, \widetilde{n_2}$ を加算するラムダ項 Add を次のように定める。

$$Add \stackrel{\text{def}}{=} Add^*\, \widetilde{0}^*$$

例 7.11. 以上のラムダ項を用いて

1. $\widetilde{1}^* + \widetilde{1}^*$,
2. $< \widetilde{1}^*, \widetilde{1}^* > + << \widetilde{1}^*, \widetilde{0}^* >, \widetilde{0}^* >$,
3. $<<<<<<< \widetilde{1}^*, \widetilde{0}^* >, \widetilde{-1}^* >, \widetilde{1}^* >, \widetilde{0}^* >, \widetilde{0}^* >, \widetilde{-1}^* >, \widetilde{1}^* >$
   $+$
   $<<<<<<<<<< \widetilde{1}^*, \widetilde{-1}^* >, \widetilde{0}^* >, \widetilde{-1}^* >, \widetilde{1}^* >, \widetilde{1}^* >, \widetilde{1}^* >, \widetilde{0}^* >, \widetilde{-1}^* >, \widetilde{-1}^* >$
   $, \widetilde{0}^* >$

の計算を行ってみた. ただし *3*に関しては直接求める方法では計算にかなりの時間がかかると思われるので行っていない. 以下がその出力である. ただし入力における $N1, N0, Nm1$ はそれぞれ $\widetilde{1}^*, \widetilde{0}^*, \widetilde{-1}^*$ をあらわす

1. *> Add N1 N1;* $\cdots$ 入力
   ——————-kekka———————-
   (lambda x . x (lambda x y z . x) (lambda x y z . y))
   0 SEC

chokusetu:(lambda x . x (lambda x y z . x) (lambda x y z . y))
27 SEC
————————————————————-
>

2. > *Add (lambda x . x N1 N1)(lambda x . x (lambda x . x N1 N0)N0);* ··· 入力
———————-*kekka*————————-
(lambda x . x (lambda x . x (lambda x . x (lambda x y z . y) (lambda x y z . x)) (lambda x y z . x)) (lambda x y z . x))
0 SEC
chokusetu:(lambda x . x (lambda x . x (lambda x . x (lambda x y z . y) (lambda x y z . x)) (lambda x y z . x)) (lambda x y z . x))
536 SEC
————————————————————-
>

3. > *Add*
(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x N1 N0)Nm1)N1)N0)N0)Nm1)N1)
(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x(lambda x.x N1 Nm1)N0)Nm1)N1)N1)N1)N0)Nm1)Nm1)N0);
··· 入力
———————-*kekka*————————-
(lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x . x (lambda x y z . y) (lambda x y z . x)) (lambda x y z . z)) (lambda x y z . y)) (lambda x y z . y)) (lambda x y z . x)) (lambda x y z . x)) (lambda x y z . y)) (lambda x y z . z)) (lambda x y z . y)) (lambda x y z . y)) (lambda x y z . x))
3 SEC
————————————————————-
>

計算自体は正しく行われているので足し算に関しては正しく定義されているようである. また *1,2* からわかるように直接求めるよりも明らかに早く求まっていることがわかる. このように再帰を使っていたりするラムダ項に関しては, 直接求めるよりもかなり早く求めるこ

とができそうである.

8 終わりに

ラムダ計算において再帰を定義するために必要なラムダ項として Y コンビネータというものがある. 実際に今回検証に利用した 2 進数の足し算を行うラムダ項も再帰があるので, この Y コンビネータを使っている. この論文では Y コンビネータに関する変換規則を定義していないのだが, Y コンビネータに関する規則を CLt 項の変換規則に加えてやればさらに効率よく求めることができそうである.

9　謝辞

本論文作成にあたり、指導教官の古森雄一先生、ならびに古森研究室の皆様に多大なご協力を頂きましたことを心から感謝いたします。

参考文献

[1] S.C.Kleene,「Proof by Cases in Formal Logic」, The Annals of Mathematics, 2nd Ser., Vol.35,No3(Jul,1934)529-544, P537 の定理 6V

[2] 千葉大学理学部 2005 年度　前期情報数学 1 配布資料 古森雄一著

[3] 千葉大学自然科学研究科 2004 年度修士論文, 大家幸敏,「ラムダ計算における 2 進計算」 http://www.math.s.chiba-u.ac.jp/~komori/masters_thesis/2004

[4] 高橋正子,「計算論 - 計算可能性とラムダ計算 -」近代科学社 1991