

ラムダ計算と型の理論

2020/09/25 竹内泉

1 関数の代数

関数を対象とし、関数という性質に注目した代数を作る。

関数を対象とするということは、関数を代数の台集合の元と見做すということである。これは以下を意味する。

数学系の言葉で言えば、〈点と写像を同一視する〉ということであり、工学系の言葉で言えば、〈データと操作を同一視する〉ということである。

通常、点（数学系）／データ（工学系）や写像（数学系）／手続（工学系）とは以下のようなものである。

5…数直線上の点（数学系）／データ（工学系）

3を足す…写像（数学系）手続（工学系）

そして関数適用は以下のように記述される。

$$\langle 3 \text{ を足す} \rangle(5) = 8$$

関数を点／データと見做すということは、関数に作用する関数をも考えるということである。

作用される関数の例として以下を挙げる。

3を足す…関数空間中の点（数学系）／関数型のデータ（工学系）

作用する関数の例として以下を挙げる。

2回繰り返す…関数空間上の写像（数学系）／手続に対する手続（工学系）

第一の関数に第二の関数を適用すると、以下を得る。

$$\langle 2 \text{ 回繰り返す} \rangle(\langle 3 \text{ を足す} \rangle) = \langle 3 \text{ を足して } 3 \text{ を足す} \rangle$$

〈3を足して3を足す〉と〈6を足す〉はある意味で等しい。

どういう意味で等しいかというと、

写像として等しい（数学系）／入出力の関係が等しい（工学系）

という意味である。これは外延的同等性と呼ばれる。

関数の等しさの基準は多数ある。外延的同等性は関数の等しさの基準の中の一つである。

〈2回繰り返す〉と〈3を足す〉から〈2回繰り返す〉(〈3を足す〉)

即ち〈3を足して3を足す〉を得るのは、函数適用という操作である。

ここでの目標は、函数の特徴だけを抽出して代数を作ることである。

函数の特徴は、函数適用の他に函数抽象がある。

函数抽象とは、新しく函数を定義する際の操作である。

例えば、以下のような函数を定義する。

f に対して $f(1)$ となる函数

f という入力に $f(1)$ と出力する函数

f という引数に $f(1)$ という返り値を返す函数

これはどれも同じ定義の様々な表現方法である。

これには幾つかの記法がある。

$f \mapsto f(1)$ 、 $(f, f(1))$ 等々

ここでは

$\lambda f.f1$

という記法を採用する。

この記法に従うと、二倍するという函数は $\lambda x.2 \cdot x$ と書かれる。

函数 $\lambda f.f1$ を入力 $\lambda x.2 \cdot x$ に適用すると、

$$(\lambda f.f1)(\lambda x.2 \cdot x) = (\lambda x.2 \cdot x)1 = 2 \cdot 1 = 2$$

のように計算され。

このような記法によって作られた函数の代数をラムダ計算と呼ぶ。

2 ラムダ計算

以降に、項文字列の代数を構成し、それを α 同値で割ることによって項の代数 Λ を構成する

α 同値は項文字列の上の関係であるのに対し、 β 変換、 η 変換等は Λ の上関係である

2.1 項文字列

2.1.1 文字列

項文字列とはある種の文字列

使う文字集合

ラムダ： λ

括弧： $(,)$

変数文字： v_1, v_2, v_3, \dots 可算無限個あり、番号が付いている

項文字列は文脈自由文法で定義される

1. 変数文字は項文字列

2. M が項文字列で x が変数文字なら、文字の接続 $(\lambda x M)$ も項文字列

3. M, N が項文字列なら、文字の接続 (MN) も項文字列

$M ::= x \mid (\lambda x M) \mid (MM)$ 但し x は変数文字

2.1.2 演算

項文字列の上にラムダ抽象と函数適用という演算を定義する

(ラムダ抽象) M が項文字列で x が変数文字なら $\lambda x.M$ は文字の接続 $(\lambda x M)$

(函数適用) M, N が項文字列なら MN は文字の接続 (MN)

函数適用は左結合的である、即ち $LMN = (LM)N$

ラムダ抽象よりも函数適用の方が結合力が強い、即ち $\lambda x.MN = \lambda x.(MN)$

$\lambda x_1 x_2 \dots x_n.M$ は $\lambda x_1. \lambda x_2. \dots \lambda x_n.M$ を表す

例

$x = v_1, y = v_2, M = (v_1 v_2)$ のとき

$\lambda x.yM = (\lambda v_1 (v_2 (v_1 v_2)))$

$x\lambda y.M = (v_1 (\lambda v_2 (v_1 v_2)))$

2.1.3 名前換え

項文字列の上に、名前換えという演算を定義する

M が項文字列で $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$ が変数文字であり、 x_1, x_2, \dots, x_n が全て相異なる時、 $M\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\}$ は以下のように定義される

表記上は、この演算の結合力は函数適用より強い

1. $x_i\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} = y_i$ 但し $i \in \{1, 2, \dots, n\}$

2. $z\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} = z$ 但し z は変数文字で $z \notin \{x_1, x_2, \dots, x_n\}$

3. $(MN)\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} =$

$M\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\}N\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\}$

4. $(\lambda z.M)\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} = \lambda z.M\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\}$

但し $z \notin \{x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n\}$

5. $(\lambda x_i.M)\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} =$
 $\lambda x_i.M\{y_1/x_1, y_2/x_2, \dots, y_{i-1}/x_{i-1}, y_{i+1}/x_{i+1}, \dots, y_n/x_n\}$
 但し $i \in \{1, 2, \dots, n\}$ 、 $x_i \notin \{y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_n\}$
6. $(\lambda y_i.M)\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} =$
 $\lambda z.M\{y_1/x_1, y_2/x_2, \dots, y_n/x_n, z/y_i\}$
 但し $i \in \{1, 2, \dots, n\}$ 、 $y_i \notin \{x_1, x_2, \dots, x_n\}$ 、 z は $M, x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$
 に現れない、番号の最も若い変数文字
7. $(\lambda x_i.M)\{y_1/x_1, y_2/x_2, \dots, y_n/x_n\} =$
 $\lambda z.M\{y_1/x_1, y_2/x_2, \dots, y_{i-1}/x_{i-1}, z/x_i, y_{i+1}/x_{i+1}, \dots, y_n/x_n\}$
 但し $i \in \{1, 2, \dots, n\}$ 、 $x_i \in \{y_1, y_2, \dots, y_{n-1}, y_{i+1}, \dots, y_n\}$ 、
 z は $M, x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$ に現れない、番号の最も若い変数文字

例

$x = \boxed{v1}$ 、 $y = \boxed{v2}$ 、 $z = \boxed{v3}$ のとき、

$$x\{y/x\} = y$$

$$xy\{y/x, x/y\} = yx$$

$$\lambda x.xy\{y/x, z/y\} = \lambda x.xz$$

$$\lambda x.xy\{x/y\} = \lambda z.zx$$

2.1.4 α 同値

項文字列の間に α 同値という関係を定義する

二つの項文字列 M と N が α 同値であることを $M \overset{\alpha}{\sim} N$ と書く

$\overset{\alpha}{\sim}$ とは、以下の規則を有限回適用して得られる関係である

$$\frac{}{\lambda x.M \overset{\alpha}{\sim} \lambda y.M\{y/x\}} \quad \text{但し } M \text{ に } y \text{ は現れない}$$

$$\frac{}{M \overset{\alpha}{\sim} M} \quad \frac{M \overset{\alpha}{\sim} N}{N \overset{\alpha}{\sim} M} \quad \frac{L \overset{\alpha}{\sim} M \quad M \overset{\alpha}{\sim} N}{L \overset{\alpha}{\sim} N}$$

$$\frac{M \overset{\alpha}{\sim} M' \quad N \overset{\alpha}{\sim} N'}{MN \overset{\alpha}{\sim} M'N'} \quad \frac{M \overset{\alpha}{\sim} M'}{\lambda x.M \overset{\alpha}{\sim} \lambda x.M'}$$

項文字列から、それと α 同値な項文字列に書き換えることを α 変換と呼ぶ

2.1.5 項

α 同値は同値関係である

項文字列全体の集合を α 同値で割った商空間を Λ と書き、その元を項と呼ぶ
変数文字 x の α 同値による同値類は $\{x\}$ である

$\{\{x\} \mid x \text{ は変数文字}\}$ を Var と書く

Var の元を変数と呼ぶ

ラムダ抽象と函数適用の演算は Λ に自然に導入される

即ち、項 $M, N \in \Lambda$ に対して、それらに属する項文字列 $M', M'' \in M, N', N'' \in N$ に関しては $M' \simeq M'', N' \simeq N''$ が成り立ち、 $M'N' \simeq M''N''$ となる

よって、 $M, N \in \Lambda$ の函数適用の結果として、 $MN = \{L \mid L \simeq M'N'\} \in \Lambda$ とすることが出来る

ラムダ抽象についても同様である

変数 $x \in Var$ と項 $M, N \in \Lambda$ に対して $\lambda x.M, MN$ は自然に導入されたラムダ抽象と函数適用の演算を表す

項 $M \in \Lambda$ と項文字列 $M', M'' \in M$ に対して、 M' と M'' の文字列の長さは一致する

この文字列の長さを M の長さと呼ぶ

以降の証明は主に項の長さに関する数学的帰納法によってなされる

2.1.6 自由変数

項文字列 M に対して変数文字の集合 $FV(M)$ を定義する

1. $FV(x) = \{x\}$ 但し x は変数文字

2. $FV(\lambda x.M) = FV(M) - \{x\}$

3. $FV(MN) = FV(M) \cup FV(N)$

項文字列 M に対して $FV(M)$ の元を M の自由変数と呼ぶ

自由変数の定義は自然に項に導入される

$M \in \Lambda$ ならば、任意の $M', M'' \in M$ に対して $FV(M') = FV(M'')$

よって $FV(M) = \{\{x\} \mid x \in FV(M'), M' \in M\} \subset Var$ と定義する

項 M に対しても $FV(M)$ の元を M の自由変数と呼ぶ

$FV(M) = \emptyset$ であるような項 M を閉項と呼ぶ

例

$FV(\lambda x.xy) = \{y\}$ 、 $FV(\lambda xy.xy) = \emptyset$ 、 $FV(x\lambda x.x) = \{x\}$

2.1.7 代入

項文字列 M 、 N と変数文字 x に対して代入 $M[N/x]$ を定義する
表記上、代入の結合力は函数適用より強い

1. $x[N/x] = N$
2. $y[N/x] = y$ 但し y は x 以外の変数文字
3. $(\lambda y.M)[N/x] = \lambda y.M[N/x]$ 但し $y \neq x$ かつ $y \notin FV(N)$
4. $(\lambda x.M)[N/x] = \lambda x.M$
5. $(\lambda y.M)[N/x] = \lambda z.M\{z/y\}[N/x]$

但し $y \in FV(N)$ かつ、 z は M, N, x に現れない、最も番号の若い変数文字

6. $(MM')[N/x] = M[N/x]M'[N/x]$

任意の項文字列 M と任意の変数文字 x 、 y に対して $M[y/x] = M\{y/x\}$

代入は項に自然に導入される

変数 $x \in Var$ と項 $M, N \in A$ に対して $M[N/x]$ は自然に導入された代入の演算を表す

同時代入 $M[N_1/x_1, N_2/x_2, \dots, N_n/x_n]$ も同様に定義する

同時代入と逐次代入は一般には同じではない

即ち $(xy)[xy/x, yx/y] = xy(yx)$ 、しかし $(xy)[xy/x][yx/y] = x(yx)(yx)$

例

$x, y, z \in Var$ に対して、

$$(\lambda x.xy)[\lambda x.xy/y] = \lambda x.x\lambda x.xy, \quad (\lambda x.xy)[\lambda y.xy/y] = \lambda z.z\lambda y.xy$$

2.2 β 変換

2.2.1 β 変換

A の中に二項関係 $\xrightarrow{\beta}$ を定義する

これは以下の規則を有限回適用して得られる関係である

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

$$\xi. M \xrightarrow{\beta} N \Rightarrow \lambda x.M \xrightarrow{\beta} \lambda x.N$$

$$\mu \& \nu. M \xrightarrow{\beta} N \Rightarrow LM \xrightarrow{\beta} LN \& ML \xrightarrow{\beta} NL$$

$M \xrightarrow{\beta} N$ の時、 M を N に書き換える操作を β 変換と呼ぶ

$\xrightarrow{\beta}$ の推移的反射的閉包を $\xrightarrow{\beta^*}$ と書く

$M \xrightarrow{\beta^*} N$ の時、 M を N に書き換える操作を β 簡約と呼ぶ

$\neg \exists N \in \Lambda. M \xrightarrow{\beta} N$ である時、 M は β 正規であると言う
 $M \xrightarrow{\beta^*} N$ かつ N が β 正規である時、 N を M の β 正規形と呼ぶ
 M に β 正規形がある時、 M は正規化可能であると言う
 例
 $(\lambda x.x)y \xrightarrow{\beta} y, (\lambda xy.xy)z \xrightarrow{\beta} \lambda y.zy, (\lambda xy.xy)y \xrightarrow{\beta} \lambda z.yz,$
 $(\lambda xy.xy)\lambda y.xy \xrightarrow{\beta} \lambda y.(\lambda y.xy)y$

2.2.2 合流性

定理

$\forall M, M', M'' \in \Lambda. M \xrightarrow{\beta^*} M' \& M \xrightarrow{\beta^*} M'' \Rightarrow \exists N \in \Lambda. M' \xrightarrow{\beta^*} N \& M'' \xrightarrow{\beta^*} N$
 N と N' が共に M の β 正規形ならば $N = N'$
 このような性質を合流性と呼ぶ

2.2.3 β 同値

$\xrightarrow{\beta}$ の推移的反射的閉包を β 同値と呼び、 \sim と書く
 正規化可能な二つの項に対して、それが β 同値であることとそれらの β 正規形が等しいことは必要十分である

2.3 η 変換

2.3.1 η 変換

Λ の中に二項関係 $\xrightarrow{\eta}$ を定義する
 これは以下の規則を有限回適用して得られる関係である
 $\eta. (\lambda x.Mx) \xrightarrow{\eta} M$ 但し $x \notin FV(M)$
 $\xi. M \xrightarrow{\eta} N \Rightarrow \lambda x.M \xrightarrow{\eta} \lambda x.N$
 $\mu \& \nu. M \xrightarrow{\eta} N \Rightarrow LM \xrightarrow{\eta} LN \& ML \xrightarrow{\eta} NL$
 $M \xrightarrow{\eta} N$ の時、 M を N に書き換える操作を η 変換と呼ぶ
 $\xrightarrow{\eta}$ の推移的反射的閉包を $\xrightarrow{\eta^*}$ と書く
 $M \xrightarrow{\eta^*} N$ の時、 M を N に書き換える操作を η 簡約と呼ぶ
 $\neg \exists N \in \Lambda. M \xrightarrow{\eta} N$ である時、 M は η 正規であると言う
 $M \xrightarrow{\eta^*} N$ かつ N が η 正規である時、 N を M の η 正規形と呼ぶ
 例
 $\lambda z.xyz \xrightarrow{\eta} xy, \lambda y.xyy \not\xrightarrow{\eta} xy$

2.3.2 合流性

定理

任意の $M, M', M'' \in \Lambda$ に対して $M \xrightarrow{\eta} M' \& M \xrightarrow{\eta} M''$ ならば、 $M' = M''$ または $\exists N \in \Lambda. M' \xrightarrow{\eta} N \& M'' \xrightarrow{\eta} N$

定理

$\forall M, M', M'' \in \Lambda. M \xrightarrow{\eta^*} M' \& M \xrightarrow{\eta^*} M'' \Rightarrow \exists N \in \Lambda. M' \xrightarrow{\eta^*} N \& M'' \xrightarrow{\eta^*} N$

任意の $M \in \Lambda$ に対して M の η 正規形は唯一存在する

2.3.3 η 同値

$\xrightarrow{\eta}$ の推移的反射的閉包を η 同値と呼び、 \sim と書く

二つの項に対して、それが η 同値であることとそれらの η 正規形が等しいことは必要十分である

2.3.4 $\beta \eta$ 変換

$M \xrightarrow{\beta} N$ または $M \xrightarrow{\eta} N$ であることを $M \xrightarrow{\beta\eta} N$ と書く

$\xrightarrow{\beta\eta}$ の推移的反射的閉包を $\xrightarrow{\beta\eta^*}$ と書く

$M \xrightarrow{\beta\eta^*} N$ の時、 M を N に書き換える操作を $\beta \eta$ 簡約と呼ぶ

$\neg \exists N \in \Lambda. M \xrightarrow{\beta\eta} N$ である時、 M は $\beta \eta$ 正規であると云う

$M \xrightarrow{\beta\eta^*} N$ かつ N が $\beta \eta$ 正規である時、 N を M の $\beta \eta$ 正規形と呼ぶ

2.3.5 $\beta \eta$ 変換の合流性

定理

$\forall M, M', M'' \in \Lambda. M \xrightarrow{\beta\eta^*} M' \& M \xrightarrow{\beta\eta^*} M'' \Rightarrow \exists N \in \Lambda. M' \xrightarrow{\beta\eta^*} N \& M'' \xrightarrow{\beta\eta^*} N$

N と N' が共に M の $\beta \eta$ 正規形ならば $N = N'$

2.3.6 $\beta \eta$ 同値

$\xrightarrow{\beta\eta}$ の推移的反射的閉包を $\beta \eta$ 同値と呼び、 $\overset{\beta\eta}{\sim}$ と書く

正規化可能な二つの項に対して、それが $\beta \eta$ 同値であることとそれらの $\beta \eta$ 正規形が等しいことは必要十分である

2.3.7 後置性

$\forall M, M' \in \Lambda. M \xrightarrow{\beta\eta^*} M' \Rightarrow \exists N \in \Lambda. M \xrightarrow{\beta^*} N \xrightarrow{\eta^*} M'$
 $M \xrightarrow{\eta^*} M'$ であり、 M が β 正規ならば、 M' も β 正規
 $M \xrightarrow{\beta\eta} M'$ であり、 M' が β 正規ならば、 β 正規な N があって $M \xrightarrow{\beta^*} N \xrightarrow{\eta} M'$

2.4 自然数の模倣

\mathbf{N} は $\{0, 1, 2, \dots\}$ を表す

$n \in \mathbf{N}$ と $M, N \in \Lambda$ に対し、 $M^n N \in \Lambda$ を定義する

$$M^0 N = N, M^{n+1} N = M(M^n N) = M^n(MN)$$

$n \in \mathbf{N}$ に対し $\bar{n} = \lambda xy. x^n y$ と定義する

\bar{n} は β 正規

$\text{suc} = \lambda zxy. zx(xy)$, $\text{suc}' = \lambda zxy. x(zxy)$ と置くと

$$\text{suc}^n \bar{0} \xrightarrow{\beta} \text{suc}'^n \bar{0} \xrightarrow{\beta^*} \bar{n}$$

$$(\lambda v w xy. vx(wxy)) \bar{m} \bar{n} \xrightarrow{\beta^*} \lambda xy. \bar{m}.x(\bar{n}xy) \xrightarrow{\beta^*} \overline{m+n}$$

$$(\lambda v wx. v(wx)) \bar{m} \bar{n} \xrightarrow{\beta^*} \lambda x. \bar{m}(\bar{n}x) \xrightarrow{\beta^*} \overline{m\bar{n}}$$

$$(\lambda v. v) \bar{m} \bar{n} \xrightarrow{\beta^*} \bar{m} \bar{n} \xrightarrow{\beta^*} \overline{n^m}$$

定理

任意の m 引数部分再帰関数 f に対して $\bar{f} \in \Lambda$ があって

1. $(n_1, n_2, \dots, n_m) \in \mathbf{N}^m$ が f の定義域に入ることと、

ある n があって $\bar{f} \bar{n}_1 \bar{n}_2 \dots \bar{n}_m \xrightarrow{\beta\eta} \bar{n}$ となることは同値

2. (n_1, n_2, \dots, n_m) が f の定義域に入るならば、 $\bar{f} \bar{n}_1 \bar{n}_2 \dots \bar{n}_m \xrightarrow{\beta^*} \overline{f(n_1, n_2, \dots, n_m)}$

定理

任意の $m \in \mathbf{N}$ と任意の $M \in \Lambda$ に対して m 引数部分再帰関数 f_M があって

1. $(n_1, n_2, \dots, n_m) \in \mathbf{N}^m$ が f_M の定義域に入ることと、

ある n があって $M \bar{n}_1 \bar{n}_2 \dots \bar{n}_m \xrightarrow{\beta\eta} \bar{n}$ となることは同値

2. (n_1, n_2, \dots, n_m) が f_M の定義域に入るならば、

$$M \bar{n}_1 \bar{n}_2 \dots \bar{n}_m \xrightarrow{\beta^*} \overline{f_M(n_1, n_2, \dots, n_m)}$$

$$P = \lambda xyp. pxy, L = \lambda p. p\lambda xy. x, R = \lambda p. p\lambda xy. y$$

と置くと

$$L(Pxy) \xrightarrow{\beta^*} x, R(Pxy) \xrightarrow{\beta^*} y$$

$$\text{Pred} = \lambda z. L(z(\lambda p. P(Rp)(\text{suc}(Rp)))(P\bar{0}\bar{0}))$$

と置くと

$$Pred \bar{0} \xrightarrow{\beta^*} \bar{0}, \quad Pred \overline{n+1} \xrightarrow{\beta^*} \bar{n}$$

$$(\lambda v.v \text{ Pred}) \bar{m} \bar{n} \xrightarrow{\beta^*} \overline{m-n} \text{ 但し } m \geq n$$

$$(\lambda v.v \text{ Pred}) \bar{m} \bar{n} \xrightarrow{\beta^*} \bar{0} \text{ 但し } m < n$$

3 単純型付計算

3.1 型

3.1.1 型

Atom は可算集合であり、番号が付いている。この集合の元を原子型と呼ぶ。

Type は、生成元 *Atom* と二項演算 \rightarrow から作られる自由代数である。

Atom の元を原始型と呼び、二項演算 \rightarrow を矢印と呼び、*Type* の元を型と呼ぶ。

\rightarrow は右結合的である、即ち $T \rightarrow U \rightarrow V = T \rightarrow (U \rightarrow V)$

Type の文字列による実例は、例えば以下のように構成される。

使う文字集合

矢印： \rightarrow

括弧： $(,)$

原子型： $\boxed{T1}, \boxed{T2}, \boxed{T3}, \dots$ 可算個あり、番号が付いている。

型は文脈自由文法で定義される。

1. 原子型は型

2. T, U が型なら、文字の接続 $\boxed{(T \rightarrow U)}$ も型

$T ::= A \mid \boxed{(T \rightarrow T)}$ 但し $A \in Atom$

3.1.2 階級

Type から \mathbf{N} への関数 *rank* を以下のように定義する。

1. $rank(A) = 0$ 但し $A \in Atom$

2. $rank(T \rightarrow U) = \max\{rank(T) + 1, rank(U)\}$

$rank(T)$ を T の階級と呼ぶ。

3.2 型付

3.2.1 環境

$Var \times Type$ の有限部分集合 $\Gamma = \{(x_1, T_1), (x_2, T_2), \dots, (x_n, T_n)\}$ であって、各 $x \in Var$ に対して $(x, T) \in \Gamma$ となる $T \in Type$ が高々一ヶしかないような Γ を環境と呼ぶ

環境全体の集合を Env と書く

$Var \times Type$ の有限部分集合 $\Gamma = \{(x_1, T_1), (x_2, T_2), \dots, (x_n, T_n)\}$ を環境と見做した場合には

$$\{x_1 : T_1, x_2 : T_2, \dots, x_n : T_n\}$$

または単に

$$x_1 : T_1, x_2 : T_2, \dots, x_n : T_n$$

と書く

環境 Γ 、 $x \in Var$ 、 $T \in Type$ に対して、

$x : T \in \Gamma$ 即ち $(x, T) \in \Gamma$ のことを $x : T \in \Gamma$ と書く

この時、 $\Gamma(x)$ は T を表す

$\Gamma, \Delta \in Env$ 、 $x \in Var$ 、 $T \in Type$ に対して、

$\Gamma \cup \Delta$ 、 $\Gamma \cup \{x : T\}$ がまた環境である場合には

Γ, Δ は $\Gamma \cup \Delta$ を表し、 $\Gamma, x : T$ 、 $x : T, \Gamma$ は $\Gamma \cup \{x : T\}$ を表す

$\Gamma = \{x_1 : T_1, x_2 : T_2, \dots, x_n : T_n\} \in Env$ に対して $Dom(\Gamma)$ は $\{x_1, x_2, \dots, x_n\} \subset Var$ を表す

3.2.2 型付規則

$\Gamma \in Env$ 、 $M \in \Lambda$ 、 $T \in Type$ に対して、三項関係 $\Gamma \vdash x : T$ は以下の規則を有限回適用して得られるものである

$$\frac{}{\Gamma \vdash x : T} (x : T \in \Gamma) \quad \frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x.M : T \rightarrow U} \quad \frac{\Gamma \vdash M : T \rightarrow U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U}$$

$\emptyset \vdash M : T$ を $\vdash M : T$ と書く

以下が成り立つ

$$\Gamma \subset \Delta \ \& \ \Gamma \vdash M : T \Rightarrow \Delta \vdash M : T$$

$$\Gamma \vdash M : T \rightarrow U \ \& \ \Delta \vdash N : T \Rightarrow \Gamma, \Delta \vdash MN : U$$

$$\Gamma \vdash M : T \Rightarrow FV(M) \subset Dom(\Gamma)$$

特に、 $\vdash M : T$ ならば M は閉項

ある $\Gamma \in Env$ と $T \in Type$ があって $\Gamma \vdash M : T$ となる時、 M は型付可能であると言う

以下が成り立つ

$\Gamma \vdash M : T$ かつ M は β 正規であり、任意の $x : U \in \Gamma$ に対し $rank(U) < rank(T)$ ならば、ある $y \in Dom(\Gamma)$ 、 $N \in \Lambda$ があって $M = \lambda y. N$

例

$x : T \rightarrow U \vdash \lambda y. xy : T \rightarrow U$ の導出

1. $x : T \rightarrow U \vdash x : T \rightarrow U$
2. $y : T \vdash y : T$
3. $x : T \rightarrow U, y : T \vdash xy : U$; 1, 2 より
4. $x : T \rightarrow U \vdash \lambda y. xy : T \rightarrow U$; 3 より

$x : (T \rightarrow T) \rightarrow U \vdash x \lambda y. y : U$ の導出

1. $x : (T \rightarrow T) \rightarrow U \vdash x : (T \rightarrow T) \rightarrow U$
2. $y : T \vdash y : T$
3. $\vdash \lambda y. y : T \rightarrow T$; 2 より
4. $x : (T \rightarrow T) \rightarrow U \vdash x \lambda y. y : U$; 1, 3 より

3.3 型付計算の代入

$\Gamma, x : U \vdash M : T$ かつ $\Gamma \vdash N : U$ ならば $\Gamma, \Gamma' \vdash M[N/x] : T$

同時代入についても同様である

3.4 主部簡約定理

$\Gamma \vdash M : T$ & $M \xrightarrow{\beta\eta^*} N \Rightarrow \Gamma \vdash N : T$

3.5 正規化定理

定理

$\Gamma \vdash M : T$ ならば M から始まる β 変換が無限に続くことはない

即ち、 M は正規化可能である

このような性質を停止性と呼ぶ

3.6 自然数の表現

$\vdash \bar{n} : (T \rightarrow T) \rightarrow T \rightarrow T$

$A \in Atom$ 、 $\vdash M : (A \rightarrow A) \rightarrow A \rightarrow A$ ならばある $n \in \mathbf{N}$ があって $M \stackrel{\beta\eta}{\sim} \bar{n}$

$A \in Atom$, $\hat{0} = A$, $\widehat{n+1} = \hat{n} \rightarrow \hat{n}$, $M = \lambda z x. z(\lambda w y. x(wy))\lambda x y. y$

と置くと、

$M\bar{m} \stackrel{\beta^*}{\rightarrow} \bar{m}$, $\vdash M : \widehat{n+1} \rightarrow \hat{n}$ ($n \geq 2$)

$\vdash \lambda v w x y. v x(w x y) : \hat{2} \rightarrow \hat{3} = \hat{2} \rightarrow \hat{2} \rightarrow \hat{2}$

$\vdash \lambda v w. v(w x) : \hat{2} \rightarrow \hat{3} = \hat{2} \rightarrow \hat{2} \rightarrow \hat{2}$

$\vdash \lambda v. v : \hat{4} = \hat{3} \rightarrow \hat{2} \rightarrow \hat{2}$

$\vdash M : \hat{n} \rightarrow \hat{2}$ ($n \geq 2$) である時、函数 $f_M : \mathbf{N} \rightarrow \mathbf{N}$ があって

$M\bar{m} \stackrel{\beta\eta}{\sim} \overline{f_M(m)}$ 、かつ、 $f_M(m)$ の計算量は $O(m) \uparrow^n O(m)$ である

$\vdash Pred : (((\hat{2} \rightarrow \hat{3}) \rightarrow \hat{2}) \rightarrow (\hat{2} \rightarrow \hat{3}) \rightarrow \hat{2}) \rightarrow \hat{2}$

$(\lambda v. v Pred)\bar{m}\bar{n}$ は型付可能ではない

3.7 文脈自由文法

環境 $\Gamma = \{x_1 : T_1, x_2 : T_2, \dots, x_n : T_n\}$ に於いて全ての T_n で $rank(T) \leq 1$ であり、 $rank(T_0) = 0$ 即ち $T_0 \in Atom$ である時、 $\{M \mid \Gamma \vdash M : T_0, M \text{ は } \beta \text{ 正規}\}$ は文脈自由文法の構造を持つ

例えば

$A \in Atom$, $\Gamma = \{x : A \rightarrow A, y : A\}$

である時、

$\Gamma \vdash M : A$ 、かつ M は β 正規ならば $M = x^n$

これは

1. y は L の元
2. M が L ならば xM は L の元

$L ::= y \mid xL$

という文脈自由文法の構造を持つ

また別の例では

$A, B \in Atom$, $\Gamma = \{w : A \rightarrow B \rightarrow A, x : B \rightarrow A \rightarrow B, y : A, z : B\}$

である時、

$\Gamma \vdash M : A$ かつ β 正規な M の集合は

1. y は A の元
 2. z は B の元
 3. $M \in A, N \in B$ ならば wMN は A の元
 3. $M \in A, N \in B$ ならば xNM は B の元
- $A ::= y|wAB, B ::= z|xBA$
 という文脈自由文法で定義される A の構造を持つ

4 定数付計算

4.1 定義

以前のラムダ計算を定数無しの計算と呼ぶ

定数付計算は、定数無しの計算から、用語の意味を少々変更し、ある種の制限を加え、計算規則を追加することによって得られる

定数無しの計算の項の変数の内、一部を定数と呼び、残りを改めて変数と呼ぶ
 定数全体の集合を $Cons$ と書き、新しく定義した変数全体の集合を改めて Var と書く

新しい定義での Var は以前の定義での $Var - Cons$ であり、以前の Var は、新しい定義の許での $Cons \cup Var$ である

定数は自由変数に含めない

即ち、新しい定義での $FV(M)$ は以前の定義の $FV(M) - Cons$ である

定数無しの計算の項の上の演算に制限を加える

項の集合 A の上の演算の内、代入については、新たに定義した Var に対する代入のみ可能である

同時代入についても同様である

またラムダ抽象 λ については、 $\lambda x.M$ の作用は $x \notin Cons$ の場合のみ可能である
 規則集合 $Rule$ は $A \times A$ の部分集合であり、以下の性質を持つ

$Rule$ の元 (M, N) を $M \rightarrow N$ と書き、計算規則と呼ぶ

$M \rightarrow N \in Rule$ の M はある $c \in Cons$ があって $cM_1M_2\dots M_n$ という形をしている

A 上の二項関係 \xrightarrow{Rule} は、以下の規則を有限回適用して得られる関係である

$$\frac{}{M \xrightarrow{Rule} N}$$

但し、ある $M' \rightarrow N' \in Rule$ と同時代入 θ があって $M = M'\theta, N = N'\theta$

この代入は定数に対する代入ではない

$$\frac{M \xrightarrow{Rule} N}{\lambda x.M \xrightarrow{Rule} \lambda x.N} \quad \frac{M \xrightarrow{Rule} N}{ML \xrightarrow{Rule} NL} \quad \frac{M \xrightarrow{Rule} N}{LM \xrightarrow{Rule} LN}$$

$Cons$ と $Rule$ を指定することによって定数付計算が特定される。

項全体の集合の名前には Λ を用いず、 $(Cons, Rule)$ に応じて新しい名前を用いることもある

ある定数付計算 $(Cons, Rule)$ に対して、 Λ の上の二項関係である $\xrightarrow{\beta}, \xrightarrow{\eta}, \xrightarrow{Rule}$ の合併の反射的推移的閉包を $\xrightarrow{*}$ と書き、この関係を計算と呼ぶ

正規、正規形等は同様に定義する

合流性は成り立つとは限らない

例

定数の集合 $Cons = \{P, L, R\}$ 、計算規則 $Rule = \{L(Pxy) \rightarrow x, R(Pxy) \rightarrow y\}$

4.2 自然数の表現

定数の集合 $\{0, \sigma, \rho\}$ 、計算規則 $\{\rho 0xy \rightarrow y, \rho(\sigma z)xy \rightarrow xz(\rho zxy)\}$

項全体の集合 $\Lambda_{\mathbf{N}}$

この計算は合流性を充たす

本節では \bar{n} は $\sigma^n 0$ を表す

$$Add = \lambda x.\rho x \lambda z.\sigma, \quad Add \bar{m}\bar{n} \xrightarrow{*} \overline{m+n}$$

$$Mult = \lambda xy.\rho x(\lambda z.Add y)\bar{0}, \quad Mult \bar{m}\bar{n} \xrightarrow{*} \overline{m\bar{n}}$$

$$Pred = \lambda x.\rho x \lambda xy.y, \quad Pred \bar{0} \xrightarrow{*} \bar{0}, \quad Pred \bar{n} \xrightarrow{*} \overline{n-1} \quad (n \geq 1)$$

$$Minus = \lambda x.\rho x \lambda z.Pred, \quad Minus \bar{m}\bar{n} \xrightarrow{*} \overline{m-n} \quad (m \geq n), \quad Minus \bar{m}\bar{n} \xrightarrow{*} \bar{0} \quad (m < n)$$

任意の m 引数部分再帰関数 f に対して $\bar{f} \in \Lambda_{\mathbf{N}}$ があって

1. $(n_1, n_2, \dots, n_m) \in \mathbf{N}^m$ が f の定義域に入ることと、

ある n があって $\bar{f}\bar{n}_1\bar{n}_2\dots\bar{n}_m \xrightarrow{*} \bar{n}$ となることとは同値

2. (n_1, n_2, \dots, n_m) が f の定義域に入るならば、 $\bar{f}\bar{n}_1\bar{n}_2\dots\bar{n}_m \xrightarrow{*} \overline{f(n_1, n_2, \dots, n_m)}$

任意の $m \in \mathbf{N}$ と任意の $M \in \Lambda_{\mathbf{N}}$ に対して m 引数部分再帰関数 f_M があって

1. $(n_1, n_2, \dots, n_m) \in \mathbf{N}^m$ が f_M の定義域に入ることと、

ある n があって $M\bar{n}_1\bar{n}_2\dots\bar{n}_m \xrightarrow{*} \bar{n}$ となることとは同値

2. (n_1, n_2, \dots, n_m) が f の定義域に入るならば、 $M\bar{n}_1\bar{n}_2\dots\bar{n}_m \xrightarrow{*} \overline{f_M(n_1, n_2, \dots, n_m)}$

4.3 二分木の表現

定数の集合 $Cons = \{nil, cons, car, cdr, nilp\}$ 、

規則 $Rule = \{car\ nil \rightarrow nil, car(cons\ xy) \rightarrow x, cdr\ nil \rightarrow nil, cdr(cons\ xy) \rightarrow y, nilp\ nil\ xy \rightarrow x, nilp(cons\ vw)xy \rightarrow y\}$

4.4 型付計算

定数付計算の定数 $Cons$ に対して、定数環境とは以下を充たす部分集合 $\Gamma_c \subset Cons \times Type$ である

各 $c \in Cons$ に対して唯一の $T \in Type$ があって $(c, T) \in \Gamma_c$

定数付計算では、 Γ を環境と呼んだ場合には $Dom(\Gamma) \cap Cons = \emptyset$ を仮定する

定数 $Cons$ と定数環境 Γ_c の許で三項関係 $\Gamma \vdash M : T$ が成り立つのは以下が共に成り立つ時である

1. ある有限部分集合 $\Gamma' \subset \Gamma_c$ があって $\Gamma, \Gamma' \vdash M : T$
2. Γ は定数 $Cons$ の許での環境

定数 $Cons$ と定数環境 Γ_c の許での型付定数付計算では、各計算規則 $M \rightarrow N \in Rule$ に対して以下を仮定する

環境 Γ と型 T があって $\Gamma \vdash M : T$ かつ $\Gamma \vdash N : T$

計算規則 $M \rightarrow N \in Rule$ から関係 \xrightarrow{Rule} を定義する際の同時代入は型付計算の同時代入である

4.5 自然数の表現

原子型の集合 $Atom = \{N\}$

型の集合 $Type$ の元は N と \rightarrow から作られる

定数の集合 $\{0, \sigma\} \cup \{\rho_T | T \in Type\}$ 、

計算規則 $\{\rho_T 0xy \rightarrow y, \rho_T(\sigma z)xy \rightarrow xz(\rho_T zxy) | T \in Type\}$

定数の型

$$\Gamma_c = \{0 : N, \sigma : N \rightarrow N\} \cup \{\rho_T : N \rightarrow (N \rightarrow T \rightarrow T) \rightarrow T \rightarrow T | T \in Type\}$$

この計算を原始再帰的計算類の計算と呼ぶ

ある計算量以下の関数はこの計算によって模倣される

4.6 二分木の表現

原子型の集合 $Atom = \{B\}$

型の集合 $Type$ の元は B と \rightarrow から作られる

定数の集合 $Cons = \{nil, cons\} \cup \{brec_T | T \in Type\}$

計算規則

$$\{brec_T nil\ xy \rightarrow y, brec_T (cons\ vw)\ xy \rightarrow xvw(brec_T vxy)(brec_T wxy) | T \in Type\}$$

定数の型 $\Gamma_c = \{nil : B, cons : B \rightarrow B \rightarrow N\}$

$$\cup \{brec_T : B \rightarrow (B \rightarrow B \rightarrow T \rightarrow T \rightarrow T) \rightarrow T \rightarrow T | T \in Type\}$$

原始再帰的計算類の計算量以下の関数はこの計算によって模倣される

$$car = \lambda x. brec_B x(\lambda v w y z. v) nil \quad cdr = \lambda x. brec_B x(\lambda v w y z. w) nil$$

$$nilp_T = \lambda x. brec_{T \rightarrow T \rightarrow T} x(\lambda v v' w w' y z. y) \lambda y z. z$$

と置くと

$$\vdash car : B \rightarrow B, \vdash cdr : B \rightarrow B, \vdash nilp_T : B \rightarrow T \rightarrow T \rightarrow T$$

$$car\ nil \overset{*}{\rightarrow} nil, car(cons\ xy) \overset{*}{\rightarrow} x, cdr\ nil \overset{*}{\rightarrow} nil, cdr(cons\ xy) \overset{*}{\rightarrow} y,$$

$$nilp_T\ nil\ xy \overset{*}{\rightarrow} x, nilp_T(cons\ vw)\ xy \overset{*}{\rightarrow} y\}$$

5 データ構造

データ構造を利用する際には、そのデータ構造の要素を作る操作と、そのデータ構造の要素から情報を取り出す操作が必要である

型付計算でデータ構造を表す型をデータ型と呼ぶ

関数では、要素を作る操作はラムダ抽象であり、情報を取り出す操作は関数適用である

対では、要素を作る操作は $P = \lambda xyz. zxy$ であり、情報を取り出す操作は $L = \lambda z. z\lambda xy. x$ と $R = \lambda z. z\lambda xy. y$ である

自然数では、要素を作る操作は 0 と σ であり、情報を取り出す操作は ρ である

二分木では、要素を作る操作は nil と $cons$ であり、情報を取り出す操作は car 、 cdr 、 $nilp$ 、 $brec_T$ 等である

6 二階型付計算

本章では二階型付計算 F について述べる

用語の定義は変更される

6.1 型文字列

6.1.1 型文字列

型文字列とはある種の文字列

使う文字集合

パイ： Π

矢印： \rightarrow

括弧： $(,)$

型変数文字： $tv1, tv2, tv3, \dots$ 可算無限個あり、番号が付いている

項文字列は文脈自由文法で定義される

1. 型変数文字は型文字列

2. T が型文字列で X が型変数文字なら、文字の接続 (ΠXT) も型文字列

3. T, U が型文字列なら、文字の接続 $(T \rightarrow U)$ も型文字列

$T ::= X \mid (\Pi XT) \mid (T \rightarrow T)$ 但し X は型変数文字

6.1.2 型文字列の上の演算

型文字列の上にパイ抽象と関数型生成という演算を定義する

(パイ抽象) T が項文字列で X が型変数文字なら $\Pi X.T$ は文字の接続

(ΠXT)

(関数型生成) T, U が型文字列なら $T \rightarrow U$ は文字の接続 $(T \rightarrow U)$

関数型合成は右結合的である、即ち $T \rightarrow T' \rightarrow T'' = T \rightarrow (T' \rightarrow T'')$

関数型合成よりも型ラムダ抽象の方が結合力が強い、

即ち $\Pi X. T \rightarrow U = (\Pi X.T) \rightarrow U$

$\Pi X_1 X_2 \dots X_n. T$ は $\Pi X_1. \Pi X_2. \dots \Pi X_n. T$ を表す

6.1.3 名前換え

型文字列の上に、名前換えという演算を定義する

T が型文字列で $X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n$ が型変数文字であり、 X_1, X_2, \dots, X_n は全て相異なる時、 $T\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\}$ は以下のように定義される

但しこの演算の結合力は \rightarrow より強い

1. $X_i\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} = X_i$ 但し $i \in \{1, 2, \dots, n\}$
2. $Z\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} = Z$
但し Z は型変数文字で $Z \notin \{X_1, X_2, \dots, X_n\}$
3. $(T \rightarrow U)\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} =$
 $T\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} \rightarrow U\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\}$
4. $(\Pi Z.T)\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} = \Pi Z.T\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\}$
但し $Z \notin \{X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n\}$
5. $(\Pi X_i.T)\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} =$
 $\Pi X_i.T\{Y_1/X_1, Y_2/X_2, \dots, Y_{i-1}/X_{i-1}, Y_{i+1}/X_{i+1}, \dots, Y_n/X_n\}$
但し $i \in \{1, 2, \dots, n\}$ 、 $X_i \notin \{Y_1, Y_2, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_n\}$
6. $(\Pi Y_i.T)\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} =$
 $\Pi Z.M\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n, Z/Y_i\}$
但し $i \in \{1, 2, \dots, n\}$ 、 $Y_i \notin \{X_1, X_2, \dots, X_n\}$ 、
 Z は $T, X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n$ に現れない、番号の最も若い型変数文字
7. $(\Pi X_i.T)\{Y_1/X_1, Y_2/X_2, \dots, Y_n/X_n\} =$
 $\Pi Z.T\{Y_1/X_1, Y_2/X_2, \dots, Y_{i-1}/X_{i-1}, Z/X_i, Y_{i+1}/X_{i+1}, \dots, Y_n/X_n\}$
但し $i \in \{1, 2, \dots, n\}$ 、 $X_i \in \{Y_1, Y_2, \dots, Y_{n-1}, Y_{i+1}, \dots, Y_n\}$ 、
 Z は $T, X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_n$ に現れない、番号の最も若い変数

6.1.4 代入

型文字列 T 、 U と型変数文字 X に対して代入 $T[U/x]$ を定義する

代入の結合力は \rightarrow より強い

1. $X[U/X] = U$
2. $Y[U/X] = Y$ 但し Y は X 以外の型変数文字
3. $(\Pi Y.T)[U/X] = \Pi Y.T[U/X]$ 但し $Y \neq X$ かつ Y は N の中に現れない
4. $(\Pi X.T)[U/X] = \Pi X.T$
5. $(\Pi Y.M)[U/X] = \Pi Z.T\{Z/Y\}[U/X]$

但し Y は N に現れ、かつ、 Z は T, U, X に現れない、最も番号の若い型変数文字

6.2 項文字列

6.2.1 環境文字列

使う文字集合

パイ： Π

矢印： \rightarrow

括弧： $(,)$

変数文字： v_1, v_2, v_3, \dots 可算無限個あり、番号が付いている

型変数文字： tv_1, tv_2, tv_3, \dots

コロン： $:$

読点： $,$

型文字列 T_1, T_2, \dots, T_n と互いに異なる変数文字 x_1, x_2, \dots, x_n があって文字列の接続で $x_1:T_1, x_2:T_2, \dots, x_n:T_n$ となっている文字列を環境文字列と呼ぶ

空文字列もまた環境文字列と呼ぶ

変数文字 x と型文字列 T に対して $x:T$ とは文字列の接続 $x:T$ という環境文字列となる演算である

環境文字列の上の部分的に定義される演算 $,$ を定義する

空でない環境文字列 Γ と Γ' に対して、もし文字列の接続 Γ, Γ' が環境文字列ならば、 Γ, Γ' とは、文字列の接続 Γ, Γ' という環境文字列となる

環境文字列 Γ と Γ' の片方のみが空である場合、 Γ, Γ' とは、空でない方の環境文字列となる

空文字列 Γ に対しては $\Gamma, \Gamma = \Gamma$

環境文字列 Γ と変数文字 x との間の演算 $\Gamma - x$ を以下のように定義する。

Γ が空文字列ならば $\Gamma - x = \Gamma$

$x:T - x$ は空文字列

$y:T - x = y:T$ 但し $y \neq x$

$\Gamma, x:T - x = \Gamma$

$\Gamma, y:T - x = (\Gamma - x), y:T$ 但し $y \neq x$

6.2.2 演算

使う文字集合

ラムダ : λ

大文字ラムダ : Λ

パイ : Π

矢印 : \rightarrow

括弧 : $(,)$

変数文字 : $v1, v2, v3, \dots$

型変数文字 : $tv1, tv2, tv3, \dots$

コロンの : $:$

以上の文字からなる文字列の上に以下の演算を定義する

(変数生成) 変数文字 x と型文字列 T に対して x^T は文字列の接続 $(x:T)$

x^T を略して単に x と書くこともある

(ラムダ抽象) 変数文字 x と型文字列 T と文字列 M に対して $\lambda x:T.M$ は文字列の接続 $(\lambda x:T.M)$

(函数適用) 文字列 M, N に対して MN は文字列の接続 (MN)

(型抽象) 型変数文字 X と文字列 M に対して $\Lambda X.M$ は文字列の接続

$(\Lambda X.M)$

(型適用) 文字列 M と型文字列 T に対して MT は文字列の接続 (MT)

函数適用、型適用は左結合的である

ラムダ抽象、型抽象の結合力は函数適用、型適用より弱い

6.2.3 型付

環境文字列 Γ と型文字列 T と文字列 M との間の三項関係 $\Gamma \vdash M : T$ は以下の規則を有限回適用して得られるものである

$\overline{\Gamma \vdash x^T : T}$ 但し Γ は $x : T$ を部分列として含む

$$\frac{\Gamma \vdash M : U}{\Gamma - x \vdash \lambda x : T.M : T \rightarrow U}$$

$$\frac{\Gamma \vdash M : T \rightarrow U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U} \quad \text{但し、} x \text{ が } \Gamma \text{ に現れるならば、それは } x : T \text{ という形で現れる}$$

$$\frac{\Gamma \vdash M : T}{\Gamma \vdash \Lambda X.M : \Pi X.T} \quad \text{但し、} X \text{ は型変数文字であって } \Gamma \text{ に登場しない}$$

$$\frac{\Gamma \vdash M : \Pi X.T}{\Gamma \vdash MU : T[U/X]}$$

ある環境文字列 Γ と型文字列 T があって $\Gamma \vdash M : T$ となるような文字列 M を項文字列と呼ぶ

6.2.4 名前換えと代入

項文字列に対する名前換え $\{ / \}$ と代入 $[/]$ は同様に定義する

6.3 型と項

6.3.1 α 同値

型文字列の上の同値関係 $\overset{\alpha}{\sim}$ と項文字列の上の同値関係 $\overset{\alpha}{\simeq}$ を同様に定義し、 α 同値と呼ぶ

例

$x = \boxed{v1}$ 、 $y = \boxed{v2}$ 、 $X = \boxed{tv1}$ 、 $Y = \boxed{tv2}$ の時、
 $\Pi X.(X \rightarrow X) \overset{\alpha}{\simeq} \Pi Y.(Y \rightarrow Y)$ 、 $\Pi X.\Pi Y.(X \rightarrow Y) \overset{\alpha}{\simeq} \Pi Y.\Pi X.(Y \rightarrow X)$
 $x^{\Pi X.(X \rightarrow X)} \overset{\alpha}{\simeq} x^{\Pi Y.(Y \rightarrow Y)}$ 、 $\lambda x^X.x^X \overset{\alpha}{\simeq} \lambda y^X.y^X$ 、 $\Pi X.\lambda x^X.x^X \overset{\alpha}{\simeq} \Pi Y.y^Y.y^Y$

6.3.2 型

型文字列の α 同値による同値類を型と呼ぶ

型全体の集合を $Type$ と書く

型変数文字 X の α 同値による同値類は $\{X\}$ である

これを型変数と呼ぶ

型変数全体の集合を $TVar$ と書く

パイ抽象、関数化型生成、代入の演算は型に自然に導入される

$X \in TVar$ 、 $T, U \in Type$ に対して $\Pi X.T$ 、 $T \rightarrow U$ 、 $T[U/X]$ は自然に導入されたパイ抽象、関数化型生成、代入を表す

6.3.3 項

項文字列の α 同値による同値類を項と呼ぶ

項全体の集合を $Term$ と書く

x^T の α 同値による同値類は $\{x^{T'} | T \overset{\alpha}{\simeq} T'\}$ である

これを変数と呼ぶ

変数文字 x と型 T に対して、文字列の集合 $\{x^{T'} | T' \in T\}$ を x^T と書き、また省略して x と書く

変数 x^T に対して、変数文字 x を x^T の変数名と呼び、 $Name(x^T)$ と書く
 変数全体の集合を Var と書く

ラムダ抽象、関数適用、型抽象、型適用、代入の演算は項形に自然に導入される
 $x \in Var$ 、 $M, N \in Term$ 、 $X \in TVar$ 、 $T \in Type$ に対して $\lambda x : T.M$ 、 MN 、 $\Lambda X.M$ 、 MT 、 $M[N/x]$ 、 $M[T/X]$ は自然に導入されたラムダ抽象、関数適用、型抽象、型適用、項代入、型の代入、を表す

項の代入はいつも定義されるとは限らない

項の中の変数に項を代入した結果が項形とならないこともある

6.3.4 自由変数と自由型変数

自由変数は同様に定義する

自由型変数もまた、自由変数と同様に定義する

$M \in Term$ に対して $FV(M)$ は M の自由変数を表す

$M \in Term$ 、 $T \in Type$ に対して $FTV(M)$ 、 $FTV(T)$ は M と T の自由型変数を表す

例

$$FV(\lambda y^Y . x^{Y \rightarrow X} y^Y) = \{x^{Y \rightarrow X}\}$$

$$FTV(\Pi Y.(X \rightarrow Y)) = \{X\}$$
, $FTV(\Lambda X. \lambda x^{Y \rightarrow X} . x^{Y \rightarrow X} y^Y) = \{Y\}$

6.3.5 環境

Var の有限部分集合 $\Gamma \subset Var$ であって、各 $x, x' \in \Gamma$ に対して $x \neq x'$ ならば $Name(x) \neq Name(x')$ となるような Γ を環境と呼ぶ

環境全体の集合を Env と書く

Var の有限部分集合 $\Gamma = \{x_1^{T_1}, x_2^{T_2}, \dots, x_n^{T_n}\}$ を環境と見做した場合には

$$\{x_1 : T_1, x_2 : T_2, \dots, x_n : T_n\}$$

または単に

$$x_1 : T_1, x_2 : T_2, \dots, x_n : T_n$$

と書く

$\Gamma \in Env$ 、 $x^T \in Var$ に対して、

$x : T \subset \Gamma$ 即ち $x^T \in \Gamma$ のことを $x : T \in \Gamma$ と書く

この時、 $\Gamma(x)$ は T を表す

環境 Γ と Δ 、及び $x^T \in Var$ に対して、

$\Gamma \cup \Delta$ 、 $\Gamma \cup \{x^T\}$ がまた環境である場合には
 Γ, Δ は $\Gamma \cup \Delta$ を表し、 $\Gamma, x : T$ 、 $x : T, \Gamma$ は $\Gamma \cup \{x^T\}$ を表す
環境 $\Gamma = \{x_1 : T_1, x_2 : T_2, \dots, x_n : T_n\}$ に対して
 $Dom(\Gamma)$ は $\{x_1, x_2, \dots, x_n\} \subset Var$ を表し
 $FTV(\Gamma)$ は $\bigcup_i FTV(T_i)$ を表す
環境文字列 $\Gamma = x_1 : T_1, x_2 : T_2, \dots, x_n : T_n$ に対して
 $\bar{\Gamma} = \{x_1 : U|U \overset{\alpha}{\sim} T_1\}, \{x_2 : U|U \overset{\alpha}{\sim} T_2\}, \dots, \{x_n : U|U \overset{\alpha}{\sim} T_n\}$
は環境である
ここに $\{x_i\} \in Var$ 、 $\bar{T}_i \in Type$
この $\bar{\Gamma}$ を環境文字列 Γ から得られる環境と呼ぶ

6.3.6 型付規則

$\Gamma \in Env$ 、 $M \in Term$ 、 $T \in Type$ に対して
ある環境文字列 Γ' 、項文字列 $M' \in M$ 、型文字列 $T' \in T$ があって
 Γ は Γ' から得られた環境であり、かつ $\Gamma' \vdash M' : T'$ であることを
 $\Gamma \vdash M : T$ と書く
この三項関係は、以下の規則を有限回適用したものとして定義される関係と一致する

$$\frac{}{\Gamma \vdash x : T} \text{ 但し } x : T \in \Gamma \qquad \frac{\Gamma \vdash M : T}{\Gamma' \vdash M : T} \text{ 但し } \Gamma \subset \Gamma'$$

$$\frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x : T. M : T \rightarrow U} \qquad \frac{\Gamma \vdash M : T \rightarrow U \quad \Gamma' \vdash N : T}{\Gamma, \Gamma' \vdash MN : U}$$

$$\frac{\Gamma \vdash M : T}{\Gamma \vdash \Lambda X. M : \Pi X. T} \text{ 但し } X \notin FTV(\Gamma) \qquad \frac{\Gamma \vdash M : \Pi X. T}{\Gamma \vdash MU : T[U/X]}$$

ラムダ抽象、関数適用、型抽象、型適用、項の代入、型の代入の演算は項に自然に導入される

項の代入 $M[N/x]$ は、 $\Gamma, x : U \vdash M : T$ かつ $\Gamma \vdash N : U$ の場合のみ定義される自由変数、自由型変数もまた項に自然に導入される

例

$y : Y \vdash$

$$(\Lambda X. \lambda x^X. x)(\Pi X. ((Y \rightarrow X) \rightarrow X)) \Lambda X. \lambda x^{Y \rightarrow X}. xy : \Pi X. (X \rightarrow Y) \rightarrow Y$$

の規則からの導出

1. $x : X \vdash x : X$
2. $\vdash \lambda x^X . x : (X \rightarrow X)$; 1 より
3. $\vdash \Lambda X . \lambda x^X . x : \Pi X . (X \rightarrow X)$; 2 より
4. $\vdash (\Lambda X . \lambda x^X . x)(\Pi X . ((Y \rightarrow X) \rightarrow X)) :$
 $\Pi X . ((Y \rightarrow X) \rightarrow X) \rightarrow \Pi X . ((Y \rightarrow X) \rightarrow X)$; 3 より
5. $x : Y \rightarrow X \vdash x : Y \rightarrow X$
6. $y : Y \vdash y : Y$
7. $x : Y \rightarrow X, y : Y \vdash xy : X$; 5, 6 より
8. $y : Y \vdash \lambda x^{Y \rightarrow X} . xy : (Y \rightarrow X) \rightarrow X$; 7 より
9. $y : Y \vdash \Lambda X . \lambda x^{Y \rightarrow X} . xy : \Pi X . ((Y \rightarrow X) \rightarrow X)$; 7 より
10. $y : Y \vdash (\Lambda X . \lambda x^X . x)(\Pi X . ((Y \rightarrow X) \rightarrow X)) \Lambda X . \lambda x^{Y \rightarrow X} . xy :$
 $\Pi X . (X \rightarrow Y) \rightarrow Y$; 4, 9 より

6.4 計算

6.4.1 β 変換

項 M 、 N に対して、二項関係 $M \xrightarrow{\beta} N$ は以下の規則を有限回適用して得られる関係である

$$\frac{}{(\lambda x . : U.M)N \xrightarrow{\beta} M[N/x]} \quad \frac{}{(\Lambda X.M)U \xrightarrow{\beta} M[U/X]}$$

$$\frac{M \xrightarrow{\beta} N}{\lambda x : T.M \xrightarrow{\beta} \lambda x : T.N} \quad \frac{M \xrightarrow{\beta} M'}{MN \xrightarrow{\beta} M'N} \quad \frac{N \xrightarrow{\beta} N'}{MN \xrightarrow{\beta} MN'}$$

$$\frac{M \xrightarrow{\beta} M'}{\Lambda X.M \xrightarrow{\beta} \Lambda X.M'} \quad \frac{M \xrightarrow{\beta} M'}{MU \xrightarrow{\beta} M'U}$$

M は項なので、ある $\Gamma \in Env$ 、 $T \in Type$ があって $\Gamma \vdash M : T$

$M \xrightarrow{\beta} N$ 、 $\Gamma \vdash M : T$ ならば $\Gamma \vdash N : T$

$\xrightarrow{\beta}$ は合流性と停止性を充たす

β 正規は同様に定義する

例

$$\begin{aligned} & (\Lambda X . \lambda x^{(Y \rightarrow X)} . xy)Y((\Lambda X . \lambda x^X . x)Y) \xrightarrow{\beta} (\lambda x^{(Y \rightarrow Y)} . xy)((\Lambda X . \lambda x^X . x)Y) \\ & \xrightarrow{\beta} (\lambda x^{(Y \rightarrow Y)} . xy)\lambda x^Y . x \xrightarrow{\beta} (\lambda x^Y . x)y \xrightarrow{\beta} y \end{aligned}$$

6.4.2 η 変換

項 M 、 N に対して、二項関係 $M \xrightarrow{\eta} N$ は以下の規則を有限回適用して得られる関係である

$$\frac{}{\lambda x : T.Mx \xrightarrow{\eta} M} \text{但し } x \notin FV(M) \quad \frac{}{\Lambda X.MX \xrightarrow{\eta} M} \text{但し } X \notin FTV(M)$$

$$\frac{M \xrightarrow{\eta} N}{\lambda x : T.M \xrightarrow{\eta} \lambda x : T.N} \quad \frac{M \xrightarrow{\eta} M'}{MN \xrightarrow{\eta} M'N} \quad \frac{N \xrightarrow{\eta} N'}{MN \xrightarrow{\eta} MN'}$$

$$\frac{M \xrightarrow{\eta} M'}{\Lambda X.M \xrightarrow{\eta} \Lambda X.M'} \quad \frac{M \xrightarrow{\eta} M'}{MU \xrightarrow{\eta} M'U}$$

$M \xrightarrow{\eta} N$ 、 $\Gamma \vdash M : T$ ならば $\Gamma \vdash N : T$

$\xrightarrow{\eta}$ は合流性と停止性を満たす

η 正規は同様に定義する

6.5 各種のデータ型

6.5.1 対の型

$T, U \in Type$ に対して、以下の型と項を定義する

$$T \times U = \Pi X.(T \rightarrow U \rightarrow X) \rightarrow X$$

$$P_{T,U} = \lambda x^T y^U \Lambda X. \lambda z^{T \rightarrow U \rightarrow X}. zxy$$

$$L_{T,U} = \lambda z^{T \times U}. zT \lambda x^T y^U .x$$

$$R_{T,U} = \lambda z^{T \times U}. zT \lambda x^T y^U .y$$

すると $\vdash P_{T,U} : T \rightarrow U \rightarrow T \times U$ 、 $\vdash L_{T,U} : T \times U \rightarrow T$ 、 $\vdash R_{T,U} : T \times U \rightarrow U$

$$L_{T,U}(P_{T,U}MN) \xrightarrow{\beta^*} M \text{、} R_{T,U}(P_{T,U}MN) \xrightarrow{\beta^*} N$$

6.5.2 三つ組の型

$T, U, V \in Type$ に対して、以下の型と項を定義する

$$\times(T, U, V) = \Pi X.(T \rightarrow U \rightarrow V \rightarrow X) \rightarrow X$$

$$Triple_{T,U,V} = \lambda x^T y^U z^V \Lambda X. \lambda w^{T \rightarrow U \rightarrow V \rightarrow X}. wxyz$$

$$Fst_{T,U,V} = \lambda w^{\times(T,U,V)}. zT \lambda x^T y^U z^V .x$$

$$Snd_{T,U,V} = \lambda w^{\times(T,U,V)}. zU \lambda x^T y^U z^V .y$$

$$Trd_{T,U,V} = \lambda w^{\times(T,U,V)}. zV \lambda x^T y^U z^V .z$$

すると

$$\begin{aligned}
&\vdash \text{Triple}_{T,U,V} : T \rightarrow U \rightarrow V \rightarrow \times(T,U,V), \vdash \text{Fst}_{T,U,V} : \times(T,U,V) \rightarrow T, \\
&\vdash \text{Snd}_{T,U,V} : \times(T,U,V) \rightarrow U, \vdash \text{Trd}_{T,U,V} : \times(T,U,V) \rightarrow V \\
&\text{Fst}_{T,U,V}(\text{Triple}_{T,U,V}MM'M'') \xrightarrow{\beta^*} M, \text{Snd}_{T,U,V}(\text{Triple}_{T,U,V}MM'M'') \xrightarrow{\beta^*} M', \\
&\text{Trd}_{T,U,V}(\text{Triple}_{T,U,V}MM'M'') \xrightarrow{\beta^*} M''
\end{aligned}$$

6.5.3 自然数の型

$\mathbf{N} = \Pi X.(X \rightarrow X) \rightarrow X \rightarrow X$ と置き、更に以下の項を定義する

$$0 = \Pi X.\lambda x^{X \rightarrow X} y^X.y$$

$$\sigma = \lambda n^{\mathbf{N}}.\Pi X.\lambda x^{X \rightarrow X} y^X.x(nXxy)$$

$$\rho_T = \lambda n^{\mathbf{N}} f^{\mathbf{N} \rightarrow T \rightarrow T} g^T.R_{\mathbf{N},T}(n(\mathbf{N}, T)$$

$$(\lambda y^{\mathbf{N} \times T}.P_{\mathbf{N},T}(\sigma(L_{\mathbf{N},T}y))(f(L_{\mathbf{N} \times T}y)(R_{\mathbf{N},T}y)))$$

$$(P_{\mathbf{N},T}0g))$$

すると

$$\vdash 0 : \mathbf{N}, \vdash \sigma : \mathbf{N} \rightarrow \mathbf{N}, \rho_T : \mathbf{N} \rightarrow (\mathbf{N} \rightarrow T \rightarrow T) \rightarrow T \rightarrow T$$

$$\rho_T 0xy \xrightarrow{\beta^*} y, \rho_T(\sigma^{(n+1)}0)xy \stackrel{\beta}{\sim} x(\sigma^n 0)(\rho_T(\sigma^n 0)xy)$$

参考文献

Hindley, J. R., & Seldin, J. P.: ‘Lambda-Calculus and Combinators: An Introduction’, Cambridge University Press, 2008.

Barendregt, H.: ‘Lambda Calculi with Types’, In Handbook of Logic in Computer Science, Vol. II, Oxford University Press, 1992.

高橋正子、「計算論」、近代科学社、1991年