1　型付ラムダ計算

ラムダ計算は関数についての表記法を提供する体系である。しかしこれまでのラムダ計算には関数の値域と定義域という概念が無くそのためラムダ項のなかには

$$\lambda x.xx$$

のようにどのような関数を意味しているのかわからないものが含まれている。型付ラムダ計算は（形無し）ラムダ計算の体系に「型」と呼ばれる概念が追加された体系であり、この型システムによりラムダ項を関数として自然に理解することができようになっている。これまでのラムダ計算とこの章で導入されるラムダ計算を区別するためにこれまでのラムダ計算を形無しラムダ計算、この章で導入されるラムダ計算を型付ラムダ計算と呼ぶ。

まずは型付ラムダ計算における型を定義する。

定義 1 型とは以下の規則によって得られる文字列のことである。

- 記号 β は型である。
- A と B が型なら $(A \Rightarrow B)$ も型である。

型は集合を形式的に扱うためのもので、関数の値域や定義域を表現するのに使われる。型 β は基底型と呼ばれ、この資料では自然数全体の集合を表している。また型 $A \Rightarrow B$ は「集合 A から集合 B への関数全体の集合」を表している。型について以下のような略記を用いる。

$$A \Rightarrow B \Rightarrow \cdots C \Rightarrow D = (A \Rightarrow (B \Rightarrow \cdots (C \Rightarrow D) \cdots))。$$

次に型付ラムダ計算におけるラムダ項を定義する。

定義 2 ラムダ項とは以下の規則により得られる文字列である。

- 変数 $x, y, z, \ldots$ はラムダ項である。
- 定数 $c_0, c_1, \ldots$ はラムダ項である。
- M と N がラムダ項なら (MN) はラムダ項である。
- M がラムダ項、x が変数、A が型なら $\lambda x : A.\, M$ はラムダ項である。

型付ラムダ計算のラムダ項と型無しラムダ計算のラムダ項の主な違いはラムダ抽象 $\lambda x : A.\, M$ において変数 x の後に型 A がある点である。この型 A は「関数」$\lambda x : A.\, M$ の定義域を表している。型付ラムダ計算のラムダ項についても型無しラムダ計算のラムダ項と同様の略記法を用いる。例えば MNL は $((MN)L)$ の略記である。型付ラムダ計算のラムダ項と型無しラムダ計算のラムダ項は区別しなくてはならないがこれ以降はどちらのラムダ項をさしているのかが明かな時は単にラムダ項と呼ぶ。

この時点ではラムダ項を機械的に定義しただけであり型の情報を無視しているラムダ項が存在している。例えばラムダ項 $(\lambda x:\beta.\,x)(\lambda y:\beta.\,y)$ は β から β への恒等関数に β から β への恒等関数を関数適用しているが、$(\lambda x:\beta.\,x)$ の定義域は β であり、型の整合性が取れていない。これから型の情報に関して整合性のないラムダ項を除外する仕組みを導入する。

数学の教科書では「x を自然数、y を整数とすると $\cdots$」など、変数がどの値を取り得るかを宣言してから議論を始めるが、型付ラムダ計算においては型環境と呼ばれるものがこの宣言に対応している。

定義 3 型環境とは変数と型の組からなる有限列でその中に同じ変数が 2 度以上現れないものである。

型環境は以下のように表記される。

$$x_1:A_1,\ x_2:A_2,\ \ldots,\ x_n:A_n$$

この型環境は「変数 x_1 は型 A_1 の値を取り、変数 x_2 は型 A_2 の値を取り、$\cdots$ 変数 x_n は型 A_n の値を取る」ことを表している。

型環境には記号 Γ や Γ' を用いる。また $\Gamma=(x_1:A_1,\ x_2:A_2,\ \ldots,\ x_n:A_n)$、$\Gamma=(y_1:B_1,\ y_2:B_2,\ \ldots,\ y_m:B_m)$ のとき $\Gamma,z:C,\Gamma'$ と書いて

$$x_1:A_1,\ x_2:A_2,\ \ldots,\ x_n:A_n,\ z:C,\ y_1:B_1,\ y_2:B_2,\ \ldots,\ y_m:B_m$$

のこととする。Γ,Γ' や $\Gamma,x:A$ なども同様にして得られる型環境とする。型環境は変数と型の組の有限列であるから型環境の長さが 0 の場合 (空列と呼ぶ) も有り得る。Γ が空列の場合は Γ,Γ' は Γ' のことであり、$\Gamma,x:A$ は $x:A$ のことである。

型判定とは型環境 Γ、ラムダ項 M、型 A の組であり $\Gamma\rhd M:A$ と書く。型環境 Γ が空列のときには $\Gamma\rhd M:A$ を単に $\rhd M:A$ と書くこともある。$x_1:A_1,\ldots,x_n:A_n\rhd M:A$ の意味は「変数 x_1 は型 A_1 の値を取り、変数 x_2 は型 A_2 の値を取り、$\cdots$ 変数 x_n は型 A_n の値を取るときラムダ項 M は A の値を取る」というものである。例えば $x:A,y:B\rhd x:A$ は変数 x が A の値を取り変数 y が B の値を取るならラムダ項 x は A の値を取るという意味である。

型判定の中には型の情報に関して整合性の取れていないものがあり、そのような型判定を除外するのが導出可能性である。

定義 4 以下の規則により型判定 $\Gamma\rhd M:A$ が導出可能であることが導けるとき $\Gamma\rhd M:A$ を導出可能な型判定とよぶ。

- $x_1:A_1,\ldots,x_n:A_n\rhd x_i:A_i$ は導出可能
- $\Gamma\rhd c_n:\beta$ は導出可能

- $\Gamma \rhd M : A \Rightarrow B$ と $\Gamma \rhd N : A$ が導出可能なら $\Gamma \rhd MN : B$ は導出可能
- $\Gamma, x : A \rhd M : B$ が導出可能なら $\Gamma \rhd \lambda x : A.\, M : A \Rightarrow B$ は導出可能

型付ラムダ計算においては導出可能な型判定 $\Gamma \rhd M : A$ のみを考え、導出可能でない型判定 $\Gamma \rhd M : A$ は「間違った」ラムダ項として考察しない。

型付ラムダ計算においても型無しラムダ計算の場合と同様に α 同値、代入操作、β 簡約が定義できる。以降はラムダ項は α 同値について同一視をする。また型付ラムダ項 M から N への β 簡約がある時には $M \to_\beta N$ と書く。

幾つか基本的な事実を挙げる。一つめは主部簡約（subject reduction）という性質で β 簡約が導出可能性を保つという主張である。

命題 1 $\Gamma \rhd M : A$ が導出可能で $M \to_\beta N$ なら $\Gamma \rhd N : A$ は導出可能である。

また代入操作も導出可能性を保つ。

命題 2 $\Gamma, x : B, \Gamma' \rhd M : A$ と $\Gamma, \Gamma' \rhd M : B$ が導出可能なら $\Gamma, \Gamma' \rhd M[x := N] : A$ は導出可能。

2　型付ラムダ計算の表示的意味論

この章ではラムダ項を実際に関数として解釈する方法を与えることで記号列の機械的な操作から成り立っているラムダ計算と関数という数学的実体を結び付ける。おおまかな流れとしては

1. 型 A に集合 $[\![A]\!]$ を対応させる。
2. 型環境 Γ に集合 $[\![\Gamma]\!]$ を対応させる。
3. 導出可能な型判定 $\Gamma \rhd M : A$ に関数 $[\![\Gamma \rhd M : A]\!] : [\![\Gamma]\!] \to [\![A]\!]$ を対応させる。

ことを行う。

まず次により型 A について集合 $[\![A]\!]$ を定義する。

$$[\![\beta]\!] = \mathbb{N}$$
$$[\![A \Rightarrow B]\!] = [\![B]\!]^{[\![A]\!]}$$

集合 $[\![A]\!]$ を型 A の解釈と呼ぶ。ここで $[\![B]\!]^{[\![A]\!]}$ は集合 $[\![A]\!]$ から集合 $[\![B]\!]$ への関数全体の集合である。この定義は「型 A の構成についての帰納的な定義」であり、あたえられた型 A に対し $[\![A]\!]$ を求めるには $[\![A]\!]$ を上の 2 つの等式をつかって書き換えていけばよい。例えば $[\![(\beta \Rightarrow \beta) \Rightarrow \beta]\!]$ は

$$[\![(\beta \Rightarrow \beta) \Rightarrow \beta]\!] = [\![\beta]\!]^{[\![\beta \Rightarrow \beta]\!]} = [\![\beta]\!]^{([\![\beta]\!]^{[\![\beta]\!]})}$$

となる。次に型環境 Γ について $[\![\Gamma]\!]$ を以下で与える。

$$[\![\Gamma]\!] = \begin{cases} [\![A_1]\!] \times \cdots \times [\![A_n]\!], & (\Gamma = (x_1 : A_1, \ldots, x_n : A_n)) \\ \{0\}, & (\Gamma \text{ が長さ } 0 \text{ の列の時}) \end{cases}$$

最後に導出可能な型判定 $\Gamma \rhd M : A$ に対し関数 $[\![\Gamma \rhd M : A]\!] \colon [\![\Gamma]\!] \to [\![A]\!]$ を定義するが、そのための準備として基本的な関数と関数についての基本的な操作を幾つか準備する。

恒等関数　集合 X から集合 X の関数 id_X を以下で定義する。

$$\mathrm{id}_X(x) = x$$

定数関数　y を集合 Y の元とする。集合 X から集合 X の関数 k_y を以下で定義する。

$$k_y(x) = y$$

対角関数　集合 X から集合 $X \times X$ の関数 δ_X を以下で定義する。

$$\delta_X(x) = (x, x)$$

射影関数　直積集合 $X_1 \times \cdots \times X_n$ から X_i への関数 $\mathrm{proj}_{n,i}$ を以下で定義する。

$$\mathrm{proj}_{n,i}(x_1, \ldots, x_n) = x_i$$

評価関数　関数 $\mathrm{evl}_{X,Y} \colon Y^X \times X \to Y$ ををを以下で定義する。

$$\mathrm{evl}_{X,Y}(f, x) = f(x)$$

関数合成　関数 $f \colon X \to Y$ と関数 $g \colon Y \to Z$ に対し関数関数 $g \circ f \colon X \to Z$ を以下で定義する。

$$(g \circ f)(x) = g(f(x))$$

関数の積　関数 $f \colon X \to Y$ と関数 $g \colon Z \to W$ に対し関数 $f \times g \colon X \times Z \to Y \times W$ を以下で定義する。

$$(f \times g)(x, z) = (f(x), g(z))$$

カリー化 (Curry 化)　関数 $f\colon X \times Y \to Z$ に対し関数 $\hat{f}\colon X \to Z^Y$ を以下で定義する。

$$(\hat{f}(x))(y) = f(x, y)$$

つまり、$\hat{f}(x)\colon Y \to Z$ は $y \in Y$ に対して $f(x, y)$ を返す関数である。

ようやく関数 $[\![\Gamma \rhd M : A]\!]\colon [\![\Gamma]\!] \to [\![A]\!]$ の定義を与える。導出可能な型判定 $\Gamma \rhd M : A$ に対し関数 $[\![\Gamma \rhd M : A]\!]\colon [\![\Gamma]\!] \to [\![A]\!]$ を次の規則により与える。

- M が変数 x の時は、型環境は $x_1 : A_1, \ldots, x_n : A_n$ というかたちでしかも $x_i = x$ となる i が唯 1 つ存在する。このとき $[\![\Gamma \rhd M : A]\!]\colon [\![\Gamma]\!] \to [\![A]\!]$ を

  $$\mathrm{proj}_{n,i}\colon [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![A_i]\!]$$

  と定義する。

- M が定数 c_n の時は、$A = \beta$ であり、$[\![\Gamma \rhd M : A]\!]\colon [\![\Gamma]\!] \to [\![A]\!]$ を

  $$k_n\colon [\![\Gamma]\!] \to [\![A]\!]$$

  と定義する。

- M が $N_0 N_1$ の形の時は $\Gamma \rhd N_0 : B \Rightarrow A$ と $\Gamma \rhd N_1 : B$ がともに導出可能となる型 B が唯 1 つ存在する。このとき $[\![\Gamma \rhd M : A]\!]\colon [\![\Gamma]\!] \to [\![A]\!]$ を

  $$\mathrm{evl}_{[\![B]\!],[\![A]\!]} \circ ([\![\Gamma \rhd N_0 : B \Rightarrow A]\!] \times [\![\Gamma \rhd N_1 : B]\!]) \circ \delta_{[\![\Gamma]\!]}$$

  と定義する。

- M が $\lambda x : B.\, N$ の形の時は、$\Gamma, x : B \rhd N : C$ が導出可能な型 C で $A = B \Rightarrow C$ となるものが存在する。型環境 Γ の長さが 1 以上であれば $[\![\Gamma \rhd M : A]\!]$ を $[\![\Gamma, x : B \rhd N : C]\!]$ のカリー化：

  $$[\![\Gamma, x : \widehat{B \rhd N} : C]\!]$$

  と定義する。また型環境 Γ の長さが 0 つまり Γ が空列であれば $[\![\Gamma \rhd M : A]\!]$ を $f = [\![\Gamma, x : B \rhd N : C]\!] \circ u$ のカリー化：

  $$\widehat{f}$$

  と定義する。ただし、u は $\{0\} \times [\![B]\!]$ から $[\![B]\!]$ への $u(0, x) = x$ で与えられる関数である。

関数 $[\![\Gamma \rhd M : A]\!]$ を型判定 $\Gamma \rhd M : A$ の解釈と呼ぶ。この定義は「型判定 $\Gamma \rhd M : A$ の導出に関する帰納的な定義」であり、与えられた型判定 $\Gamma \rhd M : A$

に対し $[\![\Gamma \rhd M : A]\!]$ を求めるには規則に従って $[\![\Gamma \rhd M : A]\!]$ を書き換えていけばよい。たとえば

$$[\![x : \beta \rhd \lambda f : \beta \Rightarrow \beta.\, f(fx) : (\beta \Rightarrow \beta) \Rightarrow \beta]\!]$$
$$= [\![x : \beta, f : \beta \Rightarrow \beta \rhd f(fx) : \beta]\!] \text{ のカリー化}$$

であり、$[\![x : \beta, f : \beta \Rightarrow \beta \rhd f(fx) : \beta]\!]$ を求めればよい。型環境 Γ を $x : \beta, f : \beta \Rightarrow \beta$ とおくと

$$[\![\Gamma \rhd f(fx) : \beta]\!] = \mathrm{evl}_{[\![\beta]\!],[\![\beta]\!]} \circ ([\![\Gamma \rhd f : \beta \Rightarrow \beta]\!] \times [\![\Gamma \rhd fx : \beta]\!]) \circ \delta_{[\![\Gamma]\!]}$$

となっている。$[\![\Gamma \rhd f : \beta \Rightarrow \beta]\!]$ は

$$[\![\Gamma \rhd f : \beta \Rightarrow \beta]\!] = \mathrm{proj}_{2,2} \colon [\![\beta]\!] \times [\![\beta]\!]^{[\![\beta]\!]} \to [\![\beta]\!]^{[\![\beta]\!]}$$

である。一方、

$$[\![\Gamma \rhd fx : \beta]\!] = \mathrm{evl}_{[\![\beta]\!],[\![\beta]\!]} \circ ([\![\Gamma \rhd f : \beta \Rightarrow \beta]\!] \times [\![\Gamma \rhd x : \beta]\!]) \circ \delta_{[\![\Gamma]\!]}$$

であり、

$$[\![\Gamma \rhd f : \beta \Rightarrow \beta]\!] = \mathrm{proj}_{2,2} \colon [\![\beta]\!] \times [\![\beta]\!]^{[\![\beta]\!]} \to [\![\beta]\!]^{[\![\beta]\!]}$$
$$[\![\Gamma \rhd x : \beta]\!] = \mathrm{proj}_{2,1} \colon [\![\beta]\!] \times [\![\beta]\!]^{[\![\beta]\!]} \to [\![\beta]\!]$$

から $[\![\Gamma \rhd fx : \beta]\!] \colon [\![\beta]\!] \times [\![\beta]\!]^{[\![\beta]\!]} \to [\![\beta]\!]$ は

$$[\![\Gamma \rhd fx : \beta]\!](u, n) = u(n)$$

で与えられる関数であることが分かる。よって $[\![\Gamma \rhd f(fx) : \beta]\!] \colon [\![\beta]\!] \times [\![\beta]\!]^{[\![\beta]\!]} \to [\![\beta]\!]$ は

$$[\![\Gamma \rhd fx : \beta]\!](u, n) = u(u(n))$$

で与えられる関数であり、そのカリー化

$$[\![x : \beta \rhd \lambda f : \beta \Rightarrow \beta.\, f(fx) : (\beta \Rightarrow \beta) \Rightarrow \beta]\!] \colon [\![\beta]\!] \to [\![\beta]\!]^{([\![\beta]\!]^{[\![\beta]\!]})}$$

は各 n に対して “関数 $f \colon \mathbb{N} \to \mathbb{N}$ を受け取ると $f(f(n))$ を返す関数” を返す関数である。

型判定の解釈はラムダ項の直感的な意味に沿った自然な形で与えられており、実は β 簡約の不変量をあたえていることが示せる。

定理 1 (健全性) 型判定 $\Gamma \rhd M : A$ が導出可能であり $M \to_\beta N$ なら $[\![\Gamma \rhd M : A]\!] = [\![\Gamma \rhd N : A]\!]$ である。

3　型付ラムダ計算の拡張

これまで扱ってきた型付ラムダ計算の表現力は非常に限られたものである。ここでは型付ラムダ計算を拡張の一例として λ_+ を導入する。

まず、λ_+ の型は型付ラムダ計算の型である。つぎに λ_+ のラムダ項は以下の規則で与えられる。

定義 5 ラムダ項とは以下の規則により得られる文字列である。

- 変数 $x, y, z, \ldots$ はラムダ項である。
- 定数 $c_0, c_1, \ldots$ はラムダ項である。
- M と N がラムダ項なら (MN) はラムダ項である。
- M がラムダ項、x が変数、A が型なら $\lambda x : A.\, M$ はラムダ項である。
- M がラムダ項なら $\mathbf{succ}(M)$ と $\mathbf{pred}(M)$ はラムダ項である。
- M と N と L がラムダ項なら $\mathbf{if}\ M\ \mathbf{then}\ N\ \mathbf{else}\ L$ はラムダ項である。
- M がラムダ項なら $\mathbf{fix}(M)$ はラムダ項である。

最後の 3 項目が新な追加である。$\mathbf{succ}(M)$ と $\mathbf{pred}(M)$ はそれぞれ足す 1 と引く 1 を意味している。$\mathbf{if}\ M\ \mathbf{then}\ N\ \mathbf{else}\ L$ は条件分岐で M の値が 0 なら N、M の値が 0 でないなら L を意味している。最後の $\mathbf{fix}(M)$ は形無しラムダ計算における不動点演算子に対応したものである。

型環境は以前と同様、変数と型の組からなる有限列でその中に同じ変数が 2 度以上現れないものとする。型判定とは型環境 Γ、λ_+ のラムダ項 M、型 A の組であり $\Gamma \rhd M : A$ と書く。型環境 Γ の長さが 0 のときには $\Gamma \rhd M : A$ を単に $\rhd M : A$ と書くこともある。型判定のうちで次の定義の意味で導出可能なものが「正しい」λ_+ のラムダ項である。

定義 6 以下の規則により型判定 $\Gamma \rhd M : A$ が導出可能であることが導けるとき $\Gamma \rhd M : A$ を導出可能な型判定とよぶ。

- $x_1 : A_1, \ldots, x_n : A_n \rhd x_i : A_i$ は導出可能
- $\Gamma \rhd c_n : \beta$ は導出可能
- $\Gamma \rhd M : A \Rightarrow B$ と $\Gamma \rhd N : A$ が導出可能なら $\Gamma \rhd MN : B$ は導出可能
- $\Gamma, x : A \rhd M : B$ が導出可能なら $\Gamma \rhd \lambda x : A.\, M : A \Rightarrow B$ は導出可能
- $\Gamma \rhd M : \beta$ が導出可能なら $\Gamma \rhd \mathbf{succ}(M) : \beta$ と $\Gamma \rhd \mathbf{pred}(M) : \beta$ は導出可能

- $\Gamma \rhd M : \beta$ と $\Gamma \rhd N : A$ と $\Gamma \rhd L : A$ の 3 つが導出可能なら $\Gamma \rhd$ **if** M **then** N **else** $L : A$ は導出可能
- $\Gamma \rhd M : A \Rightarrow A$ が導出可能なら $\Gamma \rhd \mathbf{fix}(M) : A$ は導出可能

最後に β 簡約を拡張する。

定義 7 λ_+ のラムダ項 M と N に対し以下の規則で $M \to_\beta N$ が得られたとき M から β 簡約により N が得られたといい $M \to_\beta N$ と書くことにする。

- $(\lambda x : A.\, M)N \to_\beta M[x := N]$
- $\mathbf{succ}(c_n) \to_\beta c_{n+1}$
- $\mathbf{pred}(c_n) \to_\beta \begin{cases} c_{n-1}, & (n > 0) \\ c_0, & (n = 0) \end{cases}$
- **if** c_0 **then** M **else** $N \to_\beta M$
- **if** c_{n+1} **then** M **else** $N \to_\beta N$
- $\mathbf{fix}(M) \to_\beta M(\mathbf{fix}(M))$
- $M \to_\beta M'$ なら
  - $MN \to_\beta M'N$
  - $NM \to_\beta NM'$
  - $\lambda x : A.\, M \to_\beta \lambda x : A.\, M'$
  - $\mathbf{succ}(M) \to_\beta \mathbf{succ}(M')$
  - $\mathbf{pred}(M) \to_\beta \mathbf{pred}(M')$
  - $\mathbf{fix}(M) \to_\beta \mathbf{fix}(M')$
  - **if** M **then** N **else** $L \to_\beta$ **if** M' **then** N **else** L
  - **if** N **then** M **else** $L \to_\beta$ **if** N **then** M' **else** L
  - **if** L **then** N **else** $M \to_\beta$ **if** L **then** N **else** M'

M から 0 回以上 β 簡約して N が得られたときに $M \to_\beta^* N$ と書くのはこれまでと同様である。**fix** を使えば足し算やアッカーマン関数は形無しラムダ計算の場合と同様にラムダ項で書くことができる。

$$\mathbf{add} = \mathbf{fix}(\lambda f : \beta \Rightarrow \beta \Rightarrow \beta.\, \lambda x : \beta.\, \lambda y : \beta.\, \mathbf{if}\ x\ \mathbf{then}\ y\ \mathbf{else}\ \mathbf{succ}(f\ (\mathbf{pred}(x))\ y))$$

4　領域理論

λ_+ に対しても型を集合、導出可能な型判定を関数として解釈することは可能であろうか。例として

$$\triangleright \lambda x : \beta. \mathbf{succ}(x) : \beta \Rightarrow \beta$$

を考える。このラムダ項の自然な意味は x に対して $x+1$ を返す関数 $f : \mathbb{N} \to \mathbb{N}$ であろう。しかしながらこの関数には $fx = x$ をみたす自然数 x が存在しないため、型判定

$$\triangleright \mathbf{fix}(\lambda x : \beta. \mathbf{succ}(x)) : \beta$$

の解釈を自然に与える方法は無さそうである。この問題に対し Dana Scott は

$$\text{型} \implies \text{集合} \qquad \text{型判定} \implies \text{関数}$$

という解釈のスタイルを

$$\text{型} \implies \omega \text{ 完備半順序集合} \qquad \text{型判定} \implies \text{連続関数}$$

に取り替える解決法を与えた。基本的なアイデアは $\mathbf{fix}(M)$ を $M(M(M(\cdots)))$ というラムダ項であると考え $M(M(M(\cdots)))$ を

$$\bot \qquad M(\bot) \qquad M(M(\bot)) \qquad M(M(M(\bot))) \qquad \cdots$$

というラムダ項からなるある種の近似列の極限として捉えるというものである。この章では ω 完備半順序集合と連続関数の世界における不動点定理を紹介し、λ_+ の $\mathbf{fix}(M)$ との対応を観察する。

定義 8　半順序集合とは集合 X と X 上の 2 項関係の組で以下を満すもののことである。

- すべての $x \in X$ について $x \leq x$
- すべての $x, y, z \in X$ について $x \leq y$ かつ $y \leq z$ が成り立つなら $x \leq z$
- すべての $x, y \in X$ について $x \leq y$ かつ $y \leq x$ なら $x = y$

$(X, \leq)$ を半順序集合とする。X の元からなる無限上昇列

$$x_1 \leq x_2 \leq x_3 \leq \cdots$$

のことを $(X, \leq)$ の ω-chain とよぶ。

定義 9　$(X, \leq)$ を半順序集合とする。$(X, \leq)$ の *ω-chain*

$$x_1 \leq x_2 \leq x_3 \leq \cdots$$

に対して $x \in X$ が

- すべての n で $x_n \leq x$
- すべての n で $x_n \leq y$ がなりたつなら、$x \leq y$

をみたすとき x を ω-*chain* $x_1 \leq x_2 \leq x_3 \leq \cdots$ の最小上界と呼ぶ。

ω-chain

$$x_1 \leq x_2 \leq x_3 \leq \cdots$$

の最小上界をしばしば $\bigvee_n x_n$ と書く。

定義 10 半順序集合 $(X, \leq)$ で任意の $(X, \leq)$ の ω-*chain* が最小上界をもつものを ω 完備半順序集合と呼ぶ。ω 完備半順序集合 $(X, \leq)$ が最小元を持つとき $(X, \leq)$ を点つき ω 完備半順序集合と呼ぶ。

定義 11 $(X, \leq)$、$(Y, \sqsubseteq)$ を半順序集合とし、f を X から Y への関数とする。すべての $x, x' \in X$ について $x \leq x'$ を満たすなら $fx \sqsubseteq fx'$ がなりたつとき f は $(X, \leq)$ から $(Y, \sqsubseteq)$ への単調関数であるという。

f は $(X, \leq)$ から $(Y, \sqsubseteq)$ への単調関数であるとき f は $(X, \leq)$ の ω-chain を $(Y, \sqsubseteq)$ の ω-chain にうつすことが容易に確認出来る。

定義 12 $(X, \leq)$、$(Y, \sqsubseteq)$ を ω 完備半順序集合とし、f を $(X, \leq)$ から $(Y, \sqsubseteq)$ への単調関数とする。すべての $(X, \leq)$ の ω-*chain* $x_1 \leq x_2 \leq \cdots$ に対して $f(\bigvee_n x_n) = \bigvee_n f(x_n)$ がなりたつとき f を $(X, \leq)$ から $(Y, \sqsubseteq)$ への連続関数と呼ぶ。

定理 2 $(X, \leq)$ を点つき ω 完備半順序集合とし、f を $(X, \leq)$ から $(X, \leq)$ への連続関数とする。このとき f は最小不動点を持つ。つまり

- $fx = x$
- すべての $y \in X$ について $fy = y$ なら $x \leq y$

をみたす $x \in X$ が存在する。

$(X, \leq)$ の最小元を $\bot$ としたとき f の最小不動点は ω-chain

$$\bot \leq f\bot \leq f(f\bot) \leq f(f(f\bot)) \leq \cdots$$

の最小上界として得られる。

ω 完備半順序集合と連続関数の世界にも直積や連続関数のなす ω 完備半順序集合、射影関数や評価関数など、拡張する以前の型付ラムダ計算の解釈を定義するのに使用した関数に対応した連続関数がすべて備わっており、それらとこの不動点定理を組合せることで λ_+ の解釈を与えることが可能である。またその解釈は β 簡約の不変量となっている。

ω 完備半順序集合の例を与える。

例 1 集合 A を $\{\bot, 0, 1, 2, \cdots\}$ とし、その上の 2 項関係 $\leq$ を

$$x \leq y \iff x = \bot \text{ または } x = y$$

で定義する。このとき $(A, \leq)$ は ω 完備半順序集合である。

例 2 集合 B を $(A, \leq)$ から $(A, \leq)$ への連続関数全体の集合としその上の 2 項関係 $\sqsubseteq$ を

$$f \sqsubseteq g \iff \text{すべての } x \in X \text{ で } fx \leq gx$$

とする。このとき $(B, \sqsubseteq)$ は ω 完備半順序集合である。$(B, \sqsubseteq)$ の *ω-chain*

$$f_1 \sqsubseteq f_2 \sqsubseteq f_3 \sqsubseteq \cdots$$

の最小上界 g は

$$g(x) = \bigvee_n f_n(x)$$

で与えられる関数である。

最後に最小不動点と $\mathbf{fix}(M)$ の対応みる。

例 3 つぎのようなラムダ項を考える。

$$\triangleright\, \lambda f : \beta \Rightarrow \beta.\, \lambda x : \beta.\, \mathbf{if}\ x\ \mathbf{then}\ c_0\ \mathbf{else}\ (\mathbf{add}\ x\ (f(\mathbf{pred}(x))))$$
$$: (\beta \Rightarrow \beta) \Rightarrow (\beta \Rightarrow \beta)$$

このラムダ項を M とおく。このとき

$$\mathbf{fix}(M)\ c_n \to_\beta^* \begin{cases} c_0 & (n = 0) \\ \mathbf{add}\ c_n\ (\mathbf{fix}(M)\ c_{n-1}) & (n > 0) \end{cases}$$

が成り立つことが分るので、帰納法を用いれば

$$\mathbf{fix}(M)\ c_n \to_\beta^* c_{0+1+\cdots+n}$$

が示せる。実はこのラムダ項 $\mathbf{fix}(M)$ の意味（解釈）は連続関数 $f\colon B \to B$

$$f(u) = n \text{ を受け取ると} \left\{ \begin{array}{l} n \text{ が } 0 \text{ なら } 0 \text{ を返す} \\ n \text{ が } 1 \text{ 以上の自然数なら } n \oplus u(n-1) \text{ を返す} \\ n \text{ が } \bot \text{ なら } \bot \text{ を返す} \end{array} \right\} \text{関数}$$

の最小不動点として与えることができる。ただし $x \oplus y$ は x と y がともに自然数ならそれらの和を返し、x と y のいずれかが $\bot$ なら $\bot$ を返す関数である。この関数 f がラムダ項 M に対応したものであるということは何となく

納得出来るだろう。では f の最小不動点を求めてみよう。まず B の最小元 d は常に $\perp$ を返す定数関数である。そして f の最小不動点は ω-*chain*

$$d \leq f(d) \leq f(f(d)) \leq f(f(f(d))) \leq \cdots$$

の最小上界として得られる。$f(f(\cdots(f(d))\cdots))$ は A から A への連続関数で以下の様になっている。

$$\overbrace{f(f(\cdots(f}^{n}(d))\cdots))(m) = \begin{cases} \perp & (m = \perp) \\ \sum_{k=0}^{m} k & (m \text{ は } 0 \text{ 以上 } n-1 \text{ 以下の自然数}) \\ \perp & (m \text{ は } n \text{ 以上の自然数}) \end{cases}$$

よってその最小上界 $g \in B$ は

$$g(n) = \begin{cases} \perp & (n = \perp) \\ \sum_{k=0}^{n} k & (n \text{ は自然数}) \end{cases}$$

となり、確かに $g(n)$ は $\mathbf{fix}(M)$ c_n 同様に 0 から n までの総和を与えている。

5 参考文献

- 大堀淳 プログラミング言語の基礎理論 （情報数学講座） 共立出版
- H. Barendregt. The lambda calculus, Its Syntax and Semantics.