4　領域理論

ラムダ計算の体系がもつ構造と同種の数学的構造を，ラムダ計算の世界以外に見出すことはできるのか?

$$\Downarrow$$

1969 年に D. Scott によって肯定的に解かれる.

- プログラムの扱うデータの集合を**領域**（domain）と呼ぶ.
- 領域はどのように抽象化して定義したらよいか?

4.1　データの近似と極限

計算機の中で，無限の長さのデータを表現することはできないが，次のように扱うことができる.

1. 円の面積（数値計算）： $\pi = 3.14...$ は表現できないが，円周率の少数点以下第 n 桁目の数値を求めるプログラムを作成することはできる.
   例： 半径 r と有効桁数 n を入力として面積を求めるプログラム P として，$r=1$ のとき，
   $$P(1,1) = 3,\ P(1,2) = 3.1,\ P(1,3) = 3.14,\ ...$$
   すなわち，正確な面積は，
   $$P(r,1) \leq P(r,2) \leq P(r,e) \leq ...$$
   における近似の極限.
2. 階乗のプログラム（プログラム構造）：
   $$\mathrm{fact}(x) = \textbf{if}\ x = 0\ \textbf{then}\ 1\ \textbf{else}\ x * \mathrm{fact}(x-1)$$
   の再帰呼出しを展開すると，無限の長さのプログラムが得られる.
   $$\begin{aligned}\mathrm{fact}_\infty(x) = \textbf{if}\ x = 0\ &\textbf{then}\ 1\\ &\textbf{else}\ x * (\textbf{if}\ x-1=0\ \textbf{then}\ 1\\ &\qquad\qquad \textbf{else}\ (x-1) * (\textbf{if}\ x-2=0\ \textbf{then}\ 1\\ &\qquad\qquad\qquad\qquad \textbf{else}\ x * (...\end{aligned}$$
   ここで，値が未定義であることを意味する **undef** を用いると，展開を途中で止めたプログラム $\mathrm{fact}_1, \mathrm{fact}_2, \mathrm{fact}_3, ...$ を考えることができる.
   $$\begin{aligned}\mathrm{fact}_1(x) &= \textbf{if}\ x = 0\ \textbf{then}\ 1\ \textbf{else}\ \textbf{undef}\\ \mathrm{fact}_2(x) &= \textbf{if}\ x = 0\ \textbf{then}\ 1\\ &\quad\ \textbf{else}\ x * (\textbf{if}\ x-1=0\ \textbf{then}\ 1\ \textbf{else}\ \textbf{undef})\\ \mathrm{fact}_3(x) &= \textbf{if}\ x - 0\ \textbf{then}\ 1\\ &\quad\ \textbf{else}\ n * (\textbf{if}\ x-1=0\ \textbf{then}\ 1\\ &\qquad\qquad \textbf{else}\ (x-1) * (\textbf{if}\ x-2=0\ \textbf{then}\ 1\ \textbf{else}\ \textbf{undef}))\end{aligned}$$
   すなわち，fact_n は，次の部分関数を表している.
   $$\mathrm{fact}_n(x) = \begin{cases} x! & (0 \leq x < n)\\ \textbf{undef} & (n \leq x)\end{cases}$$
   fact_n は，fact_∞ の近似であり，fact_{n+1} は，fact_n よりよい近似になっている. fact_∞ は，fact_n の極限.

- 計算機は，有限の対象しか扱えないが，有限の表現の極限を含めることによって，無限の対象を扱うことができる.

$$\Downarrow$$

このような近似の概念をもつデータ領域の**数学的構造**は?

4.1.1 プログラムのデータ領域

定義 4.1. 集合 D 上の二項関係 $\sqsubseteq$ で，次の性質を満たすものを，D 上の**半順序**（*partial order*）と呼ぶ.

1. $a \sqsubseteq a$（反射律）
2. $a \sqsubseteq b$ かつ $b \sqsubseteq a$ ならば $a = b$ （反対称律）
3. $a \sqsubseteq b$ かつ $b \sqsubseteq c$ ならば $a \sqsubseteq c$ （推移律）

半順序が定義されている集合を，**半順序集合** （*paritally ordered set*） と呼ぶ. ∎

- 近似の概念は，半順序で表す.
  1. $a \sqsubseteq b$： a は b の近似．b は a より精度が高い（情報が多い).
  2. 最小限： まったく情報を含まない．どんな要素よりも精度が低い（情報が少ない）近似の要素

定義 4.2. 半順序集合 D 上の**最小元**（*lest element* あるいは *bottom*）とは，次の条件を満たす元 $\bot \in D$ のことである.

$$\forall a \in D. \bot \sqsubseteq a$$

注 ： すべての半順序集合が最小元をもつとは限らない．しかし，半順序集合 D が，最小元をもてば，*1* つである．この最小元は，$\bot_D$ あるいは単に，$\bot$ と書く.

半順序集合 D 上の**最大元**（*greatest element* あるいは *top*）とは，次の条件満たす元 $\top \in D$ のことである.

$$\forall a \in D. a \sqsubseteq \top$$

∎

次に近似の極限の概念を導入する.

定義 4.3. D を半順序集合，X を D の部分集合とすると，元 $d \in D$ について，

$$\forall x \in X. x \sqsubseteq d$$

のとき，d は X の**上界**（*upper bound*）と呼び，$X \sqsubseteq d$ と書く．また，d が X の上界のうち最小の元であるとき，d を X の**上限**（*supremum*）あるいは**最小上界**（*least upper bound*） と呼ぶ．すなわち，X の上限は，次の *2* つの条件を満たす元 $d \in D$ である.

$$\begin{array}{l} X \sqsubseteq d \\ \forall a \in D. X \sqsubseteq a \text{ ならば } d \sqsubseteq a \end{array}$$

上界，上限の対の概念として，下界，下限を定義する．元 $d \in D$ について，

$$\forall x \in X. d \sqsubseteq x$$

のとき，d は X の下界（*lower bound*）と呼び，$d \sqsubseteq X$ と書く．また，d が X の下界のうち最大の元であるとき，d を X の**下限**（*infimum*） あるいは最大下界（*greatest lower bound*）と呼ぶ. ∎

- 半順序集合 D の部分集合 X は，常に上限をもつとは限らないが，存在すれば唯一である．その元を $\sqcup X$ で表す．同様に，X に下限が存在すれば唯一であり，$\sqcap X$ で表す．また，有限個の元に対して，次のような記法も用いる.

$$a \sqcup b = \sqcup\{a, b\} \ a \sqcap b = \sqcap\{a, b\}$$

- 上限が近似の極限を表している例：
  1. 実数全体 $\mathbf{R}$ は，$\leq$ に関して半順序．$\mathbf{R}$ の部分集合 P を，

$$P = \{3, 3.1, 3.14, 3.141, 3.1415, ...\}$$

と定義すると，P の上限が存在して，

$$\sqcup P = 3.1415... = \pi$$

である．

2. 階乗を求めるプログラムについて，自然数全体の集合 $\mathbf{N}$ から $\mathbf{N}$ への部分関数の間に，

$$f \sqsubseteq g \Leftrightarrow \forall x \in \mathbf{N}.f(x) \text{ が定義されていれば } g(x) \text{ も定義され } f(x) = g(x)$$

のように半順序を定義すると，

$$P = \{\mathrm{fact}_1, \mathrm{fact}_2, \mathrm{fact}_3, ...\}$$

の上限は，fact_∞ である．

注　：半順序集合 D のすべての部分集合 X が常に上限をもつとは限らない．

定義 4.4. 半順序集合 D のすべての部分集合 $X \subseteq D$ について上限 $\sqcup X \in D$ が存在するとき，D を**完備束**（*complete lattice*）と呼ぶ． ∎

- $X = \emptyset$ のとき，$\sqcup X$ は，D の最小元．
- $X = D$ のとき，$\sqcup X$ は，最大元．

$\Downarrow$

- 完備束は，常に，最小限と最大元をもつ． ∎

4.1.2　CPO

プログラムで扱うデータ領域として，完備束の条件 “すべての部分集合が上限をもつ” は厳しすぎる．

定義 4.5. （**ω 鎖**）
半順序集合 D の元の列

$$a_0 \sqsubseteq a_1 \sqsubseteq a_2 \sqsubseteq ...$$

を ω **鎖**（*ω-chain*）と呼ぶ．列 $< a_0, a_1, a_2, ... >$ は自然数の集合と *1* 対 *1* に対応し，$i \leq j$ ならば $a_i \sqsubseteq a_j$．

- 円の面積を求めるプログラム $P(r,1) \leq P(r,2) \leq ...$ や，再帰呼出しの無限展開 $fact_1, fact_2, ...$ は，線形に並んだ値の極限を考えた．$\Rightarrow$ すべての ω 鎖が上限をもつ半順序集合をデータ領域と考えてもよい．
ここでは，**さらに条件を緩めたもの**を採用する． ∎

定義 4.6. （**有向集合**）
半順序集合 D の**空でない**部分集合 X で，

$$\forall a \in X \forall b \in X \exists c \in X.a \sqsubseteq c \text{ かつ } b \sqsubseteq c$$

が成り立つとき，X は**有向集合**（*directed set*）と呼ぶ．

- 全体として，一定の方向を向いている列（w 鎖のように一列に並んでいる必要はない）．ω 鎖は有向集合の一例．

例　： 有向集合は，頂点●と辺によって表現されることが多い（**ハッセ図式**，*Hasse diagram*）． ∎

定義 4.7. （**cpo**）
次の *2* つの条件を満たす半順序集合 D を**完備半順序集合**（*complete partially ordered set*，**cpo**）と呼ぶ．

1. D は最小元をもつ．
2. D の任意の有向部分集合 X について，X の上限 $\sqcup X \in D$ が存在する．

プログラムが扱うデータ領域は *cpo* である.

例 1　：任意の集合 S に対して，S の部分集合全体の集合 $\mathcal{P}(S) = \{A|A \subseteq S\}$ は，集合の包含関係 $\subseteq$ に関して *cpo* となる.

例 2　：集合 S から T への部分関数全体を $[S \to T]$ と表す．部分関数間の半順序を

$$f \sqsubseteq g \Leftrightarrow \forall x \in S. f(x) \text{ が定義されていれば } g(x) \text{ も定義され } f(x) = g(x)$$

と定義すると，$[S \to T]$ は *cpo*.

例 3　：f を S から T への部分関数として，直積

$$S \times T = \{< a, b > | a \in S \text{ かつ } b \in T\}$$

の部分集合

$$\{< x, f(x) > | x \in S \text{ かつ } f(x) \text{ が定義されている }\}$$

を f の**グラフ**と呼ぶ．部分関数 f とそのグラフを同一視すると，$f \subseteq g$ と $f \sqsubseteq g$ は同じ．このとき，最小元は，空集合 $\emptyset \in S \times T$ であり，$[S \to T]$ の有向部分集合 F の上限は，$\cup F$.

例 4　：集合 S に要素 $\perp$ を加えた集合 $S_\perp$ は，

$$a \sqsubseteq b \Leftrightarrow a = \perp \text{ あるいは } a = b$$

と定義した半順序について *cpo*．この *cpo* は，**平坦 cpo**（*flat cpo*）と呼ぶ.

平坦 cpo1　：$\perp$ を未定義として，$f \in [S \to T]$ は次の全関数 $\hat{f} : S \to T_\perp$ で表せる.

$$\hat{f}(x) = \begin{cases} f(x) & (f(x) \text{ が定義}) \\ \perp & (f(x) \text{ が未定義}) \end{cases}$$

平坦 cpo2　：$\mathbf{N}_\perp$，$\mathbf{B}_\perp$（$B = \{true, false\}$）

例 5　：実数 $a, b \in \mathbf{R}$ について，

$$[a, b] = \{x \in \mathbf{R} | a \le x \le b\}$$

の閉区間を定義する．閉区間に $\mathbf{R}$ 自身を加えた集合

$$I_\mathbf{R} = \{[a, b] | a \le b\} \cup \{\mathbf{R}\}$$

は，包含関係 $\subseteq$ に対して *cpo*.

- $I_\mathbf{R}$ の部分集合 $I^*_\mathbf{R}$ を

$$I^*_\mathbf{R} = \{[a, b] | a \le b \text{ で } a \text{ と } b \text{ は有理数 }\}$$

  と定義すると，任意の $[a, b] \in I_\mathbf{R}$ について，

$$[a, b] = \sqcup \{[c, d] \in I^*_\mathbf{R} | [c, d] \sqsubseteq [a, b]\}$$

  が成り立つ．$I_\mathbf{R}$ の各要素は，$I^*_\mathbf{R}$ のある集合の上限.

- $a = b$ とおくと，

$$[a, a] = \sqcup \{[c, d] \in I^*_\mathbf{R} | c \le a \le d\}$$

  各実数は，有理数の区間の集合の上限.　∎

- cpo の条件を弱めて，“すべての ω 鎖が上限をもつ” とした場合 $\Rightarrow$ 2 つの違いは，濃度の問題.

命題 4.1. 半順序 D について次の *2* つの条件は同値.

1. 任意の可算な有向集合 $X \subseteq D$ について，X は上限をもつ.
2. 任意の ω 鎖は上限をもつ.

証明：

(1) $\Rightarrow$ *(2)*： w 鎖は可算な有向集合.

(2) $\Rightarrow$ *(1)*： X の元の w 鎖 $A = a_0, a_1, a_2, ..., a_n, ...$ を n に関する帰納法で定義する．$a_0 = x_0$ として，a_n が定義されるとすると，X は有向集合なので，a_n と x_{n+1} の上界 $x \in X$ （$a_n \sqsubseteq x$ かつ $x_{n+1} \sqsubseteq x$）が存在する．そのうちの *1* つを a_{n+1} と定義すると，A は ω 鎖．*(2)* から A の上限が存在すると過程すると，任意 $x_n \in X$ について $x_n \sqsubseteq a_n$ なので，$\sqcup A$ は X の上界．また，$A \subseteq X$ なので，$\sqcup A$ は X の上界のうち最小. ∎

- 上限の計算に役立つ命題 $\Downarrow$

命題 4.2. D を半順序集合，X を D の部分集合，$d \in D$ とすると，次の *2* つの条件は同値

1. $d = \sqcup X$ （X の上限が存在し，d に等しい）.
2. $\forall a \in D. d \sqsubseteq a \Leftrightarrow X \sqsubseteq a$

証明：

(1)$\Rightarrow$*(2)* ： $d = \sqcup X$ とすると，$X \sqsubseteq a$ ならば $d \sqsubseteq a$．また，d は X の上界なので，$d \sqsubseteq a$ ならば $X \sqsubseteq a$.

(2)$\Rightarrow$*(1)* ：*(2)* で，$a = d$ とおくと $X \sqsubseteq d$ （d は X の上界）．また，$X \sqsubseteq a$ ならば $d \sqsubseteq a$ なので，d は，X の最小上界.