合成型上の関数

胡 振江
東京大学 計数工学科
2006 年度

内容

- 組型とその上の関数
  - 基本関数 (構成子＋分離関数)
  - 有理数上の計算
- 関数型とその上の関数
  - 関数合成
  - 逆関数
  - 正格関数・非正格関数
- リスト型とその上の関数
  - リスト上の再帰関数
  - リスト上の高階関数

対型

型 (T_1, T_2) は第 1 要素が T_1 型の値で第 2 要素が T_2 型の値であるような値の対で構成される型である.

$$
\begin{array}{lcl}
(17.3, 3) & :: & (\mathsf{Float}, \mathsf{Int}) \\
(3, 6) & :: & (\mathsf{Int}, \mathsf{Int}) \\
(\mathsf{True}, (+)) & :: & (\mathsf{Bool}, \mathsf{Int} \to \mathsf{Int} \to \mathsf{Int})
\end{array}
$$

対型上の基本関数

- 構成子 $(,)$

$$\begin{array}{lcl} (,) & :: & a \to b \to (a,b) \\ (,)\ x\ y & = & (x,y) \end{array}$$

- 分離関数 fst, snd:

$$\begin{array}{lcl} \mathsf{fst} & :: & (a,b) \to a \\ \mathsf{fst}\ (x,y) & = & x \end{array}$$

$$\begin{array}{lcl} \mathsf{snd} & :: & (a,b) \to b \\ \mathsf{snd}\ (x,y) & = & y \end{array}$$

組型

型 $(T_1, \ldots, T_n)$ は第 1 要素が T_1 型の値で第 n 要素が T_n 型の値であるような値の組で構成される型である.

$$(17.3, 3, \mathsf{True}) \quad :: \quad (\mathsf{Float}, \mathsf{Int}, \mathsf{Bool})$$
$$(3, 6, 7, 8, 9) \quad :: \quad (\mathsf{Int}, \mathsf{Int}, \mathsf{Int}, \mathsf{Int}, \mathsf{Int})$$

対型上の基本関数

- n 組を作る構成子 $(,\ldots,)$

$$(,\ldots,) :: a_1 \to \cdots \to a_n \to (a_1,\ldots,a_n)$$
$$(,\ldots,)\ x_1\ \cdots\ x_n = (x_1,\ldots,x_n)$$

- 分離関数 sel_i^n:

$$\mathsf{sel}_i^n :: (a_1,\ldots,a_i,\ldots,a_n) \to a_i$$
$$\mathsf{sel}_i^n\ (x_1,\ldots,x_i,\ldots,x_n) = x_i$$

組型上の関数の定義

2 次方程式の根を求める問題

- 2 次方程式 $ax^2 + bx + c = 0$ を係数の組 (a, b, c) で表現する.
- 二つの根を対 (r_1, r_2) で表現する.

$$
\begin{aligned}
&\mathsf{roots} \;::\; (\mathsf{Float}, \mathsf{Float}, \mathsf{Float}) \to (\mathsf{Float}, \mathsf{Float}) \\
&\mathsf{roots}\ (a, b, c) \mid d \geq 0 \;=\; (r_1, r_2) \\
&\qquad \textbf{where} \\
&\qquad\quad r_1 = (-b + r)/(2 * a) \\
&\qquad\quad r_2 = (-b - r)/(2 * a) \\
&\qquad\quad r = \mathsf{sqrt}\ d \\
&\qquad\quad d = b^2 - 4 * a * c
\end{aligned}
$$

有理数上の演算の定義

- 有理数 x/y を整数の対 (x, y) で表現する.
- 有理数の正規化：norm $(8, 6) \Rightarrow (4, 3)$

```
norm :: (Int, Int) → (Int, Int)
norm (x, y) | y ≠ 0 = (div u d, div v d)
    where
        u = (sign y) * x
        v = abs y
        d = gcd (abs u) v
```

- 有理数の四則演算子の定義

$$\begin{aligned}
&\mathsf{radd} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \\
&\mathsf{radd}\ (x, y)\ (u, v) \ = \ \mathsf{norm}\ (x * v + u * y, y * v)
\end{aligned}$$

$$\begin{aligned}
&\mathsf{rsub} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \\
&\mathsf{rsub}\ (x, y)\ (u, v) \ = \ \mathsf{norm}\ (x * v - u * y, y * v)
\end{aligned}$$

$$\begin{aligned}
&\mathsf{rmul} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \\
&\mathsf{rmul}\ (x, y)\ (u, v) \ = \ \mathsf{norm}\ (x * u, y * v)
\end{aligned}$$

$$\begin{aligned}
&\mathsf{rdiv} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \\
&\mathsf{rdiv}\ (x, y)\ (u, v) \ = \ \mathsf{norm}\ (x * v, y * u)
\end{aligned}$$

- 有理数の比較演算子の定義

$$\mathsf{requals} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to \mathsf{Bool}$$
$$\mathsf{requals}\ (x, y)\ (u, v)\ =\ x * v == y * u$$

$$\mathsf{rless} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to \mathsf{Bool}$$
$$\mathsf{rless}\ (x, y)\ (u, v)\ =\ x * v < y * u$$

$$\mathsf{rgreater} :: (\mathsf{Int}, \mathsf{Int}) \to (\mathsf{Int}, \mathsf{Int}) \to \mathsf{Bool}$$
$$\mathsf{rgreater}\ (x, y)\ (u, v)\ =\ x * v > y * u$$

- 分数を表示する関数の定義

$$\begin{aligned}&\mathsf{showrat} :: (\mathsf{Int}, \mathsf{Int}) \to \mathsf{String}\\&\mathsf{showrat}\ (x, y)\ =\ \mathbf{let}\ (u, v) = \mathsf{norm}\ (x, y)\\&\qquad\qquad\qquad\qquad\mathbf{in\ if}\ v == 1\ \mathbf{then}\ \mathsf{show}\ u\ \mathbf{else}\ \mathsf{show}\ u +\!\!+ \text{"/"} +\!\!+ \mathsf{show}\ v\end{aligned}$$

関数型と関数上の関数

関数型 $(\rightarrow)$ はすべての関数の集まりである.

$$+,\ -,\ *,\ /,\ \mathsf{square},\ (\wedge),\ \mathsf{ord},\ \mathsf{until},\ldots$$

関数はあらゆる型の値を引数にとりうるし，あらゆる種類の値を結果として返すことができる.

高階関数: 引数として関数をとる、あるいは結果として関数を返す関数.

例：微分演算子

$$\frac{d}{dx} :: \text{関数} \rightarrow \text{導関数}$$

関数合成

- $(\circ)$: 二つの関数を合成する演算子.

$$\begin{array}{lcl} (\circ) & :: & (\beta \to \gamma) \to (\alpha \to \beta) \to (\alpha \to \gamma) \\ (f \circ g)\ x & = & f\ (g\ x) \end{array}$$

- 関数合成は結合性をもつ演算子.

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

関数の equality

$$f = g \quad \textbf{iff} \quad \forall x.f\ x = g\ x.$$

逆関数

単射関数 $f :: A \to B$ に対して，A の任意の値 x に対して，

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

となる g を f の逆関数といい，一般的に f^{-1} と表す.

例：関数

$$\begin{array}{lcl} f & :: & \mathsf{Int} \to (\mathsf{Int}, \mathsf{Int}) \\ f\ x & = & (\mathsf{sign}\ x, \mathsf{abs}\ x) \end{array}$$

は単射であり，次の逆関数をもつ.

$$\begin{array}{lcl} f^{-1} & :: & (\mathsf{Int}, \mathsf{int}) \to \mathsf{Int} \\ f^{-1}\ (s, a) & = & s * a \end{array}$$

正格関数と非正格関数

- 正格関数
  - 定義：$f\ \bot = \bot$ であるような関数 f を正格関数 (strict function) という.
  - 例：$square(1/0) = \bot$
- 非正格関数
  - 定義：正格でない関数
  - 例：次の定義について考えよう.

$$\begin{array}{lcl} \mathsf{three} & :: & \mathsf{Int} \to \mathsf{Int} \\ \mathsf{three}\ x & = & 3 \end{array}$$

このときに，three $(1/0) = 3$ である.

リスト

リスト型 $[\alpha]$ は線形に順序のついた，型が α である値の集まりである．

$[1,2,3]$	::	$[\mathsf{Int}]$
$['h','e','l','l','o']$	::	$[\mathsf{Char}]$
$[[1,2],[3]]$	::	$[[\mathsf{Int}]]$
$[(+),(-)]$	::	$[\mathsf{Int} \to \mathsf{Int} \to \mathsf{Int}]$
$[\,]$	::	$[a]$
$[1,\text{"fine day"}]$		リストではない！

数学の多くの分野で集合が重要であるのと同様に，関数プログラミングにおいてはリストが重要な役割を果たすのである．

リストの構成的定義

リスト型 $[\alpha]$ の集合を次のように構成的に定義する.

1. 空リスト $[\,]$ は $[\alpha]$ の要素である. $[\,] \in [\alpha]$.
2. $x \in [\alpha]$ ならば, $a \in \alpha$ に対して, $a : x \in [\alpha]$ である.
3. これら以外に $[\alpha]$ の要素ではない.

今後, 簡便な表記法として

$$a_0 : (a_1 : (\cdots (a_{n-1} : [\,])))$$

を

$$[a_0, a_1, \ldots, a_{n-1}]$$

のように書くこととする.

リスト上の基本関数

- リスト構成子

  $[\,]$: 空リストを作る構成子

  $$[\,] :: [\alpha]$$

  $(:)$: リストの先頭に要素を付け加える構成子

  $$(:) :: \alpha \to [\alpha] \to [\alpha]$$

- リストを分離する関数

$$
\begin{array}{lcl}
\mathsf{head} & :: & [\alpha] \to \alpha \\
\mathsf{head}\ (a : x) & = & a
\end{array}
$$

$$
\begin{array}{lcl}
\mathsf{tail} & :: & [\alpha] \to [\alpha] \\
\mathsf{tail}\ (a : x) & = & x
\end{array}
$$

- 空リストと判断する関数

$$
\begin{array}{lcl}
\mathsf{null} & :: & [\alpha] \to \mathsf{Bool} \\
\mathsf{null}\ [\,] & = & \mathsf{True} \\
\mathsf{null}\ (a : x) & = & \mathsf{False}
\end{array}
$$

リスト上の関数の定義：再帰関数

再帰的なデータを処理する関数は一般的に再帰的関数になる．

- リストの長さを求める関数

$$\begin{aligned} \mathsf{length} &:: [a] \to \mathsf{Int} \\ \mathsf{length}\ [\,] &= 0 \\ \mathsf{length}\ (a : x) &= 1 + \mathsf{length}\ x \end{aligned}$$

例：

$$\mathsf{length}\ [1, 2, 4, 3, 3] \Rightarrow 5$$

- リストの要素の和を求める関数

$$\begin{array}{lcl} \mathsf{sum} & :: & \mathsf{Num}\ a \Rightarrow [a] \to a \\ \mathsf{sum}\ [\,] & = & 0 \\ \mathsf{sum}\ (a : x) & = & a + \mathsf{sum}\ x \end{array}$$

例：

$$\mathsf{sum}\ [1, 2, 3, 4, 5] \Rightarrow 15$$

- 二つリストを連接する関数

$$
\begin{array}{lcl}
(+\!\!+) & :: & [a] \to [a] \to [a] \\
(+\!\!+)\ [\,]\ y & = & y \\
(+\!\!+)\ (a : x)\ y & = & a : ((+\!\!+)\ x\ y)
\end{array}
$$

例：

$$[1,2,3] +\!\!+ [4,5,6,7] \Rightarrow [1,2,3,4,5,6,7]$$

注：中置演算子による分かりやすい記述

$$
\begin{array}{lcl}
(+\!\!+) & :: & [a] \to [a] \to [a] \\
[\,] +\!\!+ y & = & y \\
(a : x) +\!\!+ y & = & a : (x +\!\!+ y)
\end{array}
$$

- リストの綴じ合わせ関数

$$\begin{array}{lcl} \mathsf{zip} & :: & [a] \to [b] \to [(a,b)] \\ \mathsf{zip}\ [\,]\ [\,] & = & [\,] \\ \mathsf{zip}\ (a:x)\ (b:y) & = & (a,b) : \mathsf{zip}\ x\ y \end{array}$$

例:

$$\mathsf{zip}\ [1,2,3,4]\ [11,12,13,14] \Rightarrow [(1,11),(2,12),(3,13),(4,14)]$$

- 先頭部分の取り出し

$$
\begin{array}{lcl}
\mathsf{take} & :: & \mathsf{Int} \to [a] \to [a] \\
\mathsf{take}\ 0\ x & = & [\,] \\
\mathsf{take}\ (n+1)\ [\,] & = & [\,] \\
\mathsf{take}\ (n+1)\ (a : x) & = & a : \mathsf{take}\ n\ x
\end{array}
$$

例:

$$\mathsf{take}\ 3\ [1,2,3,4] \Rightarrow [1,2,3]$$

- 先頭部分の取り出し

$$
\begin{array}{lcl}
\mathsf{drop} & :: & \mathsf{Int} \to [a] \to [a] \\
\mathsf{drop}\ 0\ x & = & x \\
\mathsf{drop}\ (n+1)\ [\,] & = & [\,] \\
\mathsf{drop}\ (n+1)\ (a : x) & = & \mathsf{drop}\ n\ x
\end{array}
$$

例:

$$\mathsf{drop}\ 3\ [1,2,3,4] \Rightarrow [4]$$

- リストの各要素に関数を適用する関数

$$\begin{array}{lcl} \mathsf{map} & :: & (\alpha \to \beta) \to [\alpha] \to [\beta] \\ \mathsf{map}\ f\ [\,] & = & [\,] \\ \mathsf{map}\ f\ (a : x) & = & f\ a : \mathsf{map}\ f\ x \end{array}$$

例：

$$\mathsf{map\ square}\ [1, 2, 3, 4] \Rightarrow [1, 4, 9, 16]$$

- リストの各要素に述語を適用し述語を満たさない要素を取り除く関数

$$
\begin{array}{lcl}
\mathsf{filter} & :: & (\alpha \to \mathsf{Bool}) \to [\alpha] \to [\alpha] \\
\mathsf{filter}\ p\ [\,] & = & [\,] \\
\mathsf{filter}\ \ (a : x) & = & \textbf{if}\ p\ a\ \textbf{then}\ a : \mathsf{filter}\ p\ x\ \textbf{else}\ \mathsf{filter}\ p\ x
\end{array}
$$

例：

$$\mathsf{filter\ even}\ [1, 2, 3, 4] \Rightarrow [2, 4]$$

- 右側畳み込み関数

$$\begin{array}{lcl} \mathsf{foldr} & :: & (a \to b \to b) \to b \to [a] \to b \\ \mathsf{foldr}\ (\oplus)\ e\ [\,] & = & e \\ \mathsf{foldr}\ (\oplus)\ e\ (a : x) & = & a \oplus \mathsf{foldr}\ (\oplus)\ e\ x \end{array}$$

例：

$$\mathsf{foldr}\ (\oplus)\ e\ [a_1, a_1, a_3] \Rightarrow a_1 \oplus (a_2 \oplus (a_3 \oplus e))$$

foldr の使用例：

$$
\begin{array}{lcl}
\mathsf{sum} & :: & [\mathsf{Int}] \to [\mathsf{Int}] \\
\mathsf{sum} & = & \mathsf{foldr}\ (+)\ 0 \\
 & & \\
\mathsf{produt} & :: & [\mathsf{Int}] \to [\mathsf{Int}] \\
\mathsf{product} & = & \mathsf{foldr}\ (*)\ 1 \\
 & & \\
\mathsf{and} & :: & [\mathsf{Bool}] \to \mathsf{Bool} \\
\mathsf{and} & = & \mathsf{foldr}\ (\wedge)\ \mathsf{True} \\
 & & \\
\mathsf{or} & :: & [\mathsf{Bool}] \to \mathsf{Bool} \\
\mathsf{or} & = & \mathsf{foldr}\ (\vee)\ \mathsf{False}
\end{array}
$$

```
concat    ::   [[a]] → [a]
concat    =    foldr (++) []

reverse   ::   [a] → [a]
reverse   =    foldr postfix [] where postfix a r = r ++ [a]
```

- 左側畳み込み関数

$$\begin{array}{lcl} \mathsf{foldl} & :: & (b \to a \to b) \to b \to [a] \to b \\ \mathsf{foldl}\ (\oplus)\ e\ [\,] & = & e \\ \mathsf{foldl}\ (\oplus)\ e\ (x +\!\!+ [a]) & = & (\mathsf{foldl}\ (\oplus)\ e\ x) \oplus a \end{array}$$

例：

$$\mathsf{foldl}\ (\oplus)\ e\ [a_1, a_1, a_3] \Rightarrow ((e \oplus a_1) \oplus a_2) \oplus a_3$$

練習問題：次の pack 関数を foldl で定義せよ.

$$\mathsf{pack}\ [x_n, x_{n-1}, \ldots, x_0] = x_n * 10^n + x_{n-1} * 10^{n-1} + \cdots + x_0$$

練習問題

- 教科書の 2.4, 2.5, 2.6, 5.1, 5.2, 5.3 を復習すること.