函　数

School of Computer
Wuhan University

本章内容

1. 函数、复合函数
   - 函数的定义
   - 复合函数
   - 单射、满射和双射
2. 函数的递归定义
   - 自然数集合上的递归函数
   - Euclid算法和尾递归
   - List集合上的递归函数
   - Ackermann函数
   - 高阶函数

Alonzo Church (1903 – 1995)

Outline

1 函数、复合函数
- 函数的定义
- 复合函数
- 单射、满射和双射

2 函数的递归定义

Example

Example

- 命题公式$G(P_1, P_2, \ldots, P_n)$的语义解释$I(G)$：
$$\{0,1\}^n \to \{0,1\},$$
$$\langle x_1, x_2, \ldots, x_n \mapsto G(x_1, x_2, \ldots, x_n)$$
实际上是$\{0,1\}^n \times \{0,1\}$上的一个特殊的关系；
- 设$\mathcal{P}$是所有命题公式的集合，$\mathcal{B}$是所有布尔函数的集合，所有命题公式的语义是上述两者的特殊关系$\mathscr{I}$:
$$\mathcal{P} \to \mathcal{B}$$
$$G \mapsto I(G)$$

Example

Example

- 程序执行过程中的状态$\mathcal{S}$可以看成是存储空间$\mathcal{M}$到其存储的数值$\mathbb{Z}$上的函数：
$$\mathcal{M} \to \mathbb{Z},$$
$$m \mapsto \mathcal{S}(s)$$
- 程序的运行环境 $\mathcal{E}$是名称集合$Name$到存储空间$\mathcal{M}$的函数：
$$Name \to \mathcal{M},$$
$$x \mapsto \mathcal{E}(x)$$

函数的定义

Definition (函数, function(map, mapping))

设 f 是集合X到 Y上的关系($f \subseteq X \times Y$), f 是函数, iff, f 满足下述两个条件:

1. 完全性: $\forall x \in X, \exists y \in Y$, such that $\langle x, y\rangle \in f$;
2. 多对一: if $\langle x, y\rangle \in f \wedge \langle x, z\rangle \in f$, then $y = z$;

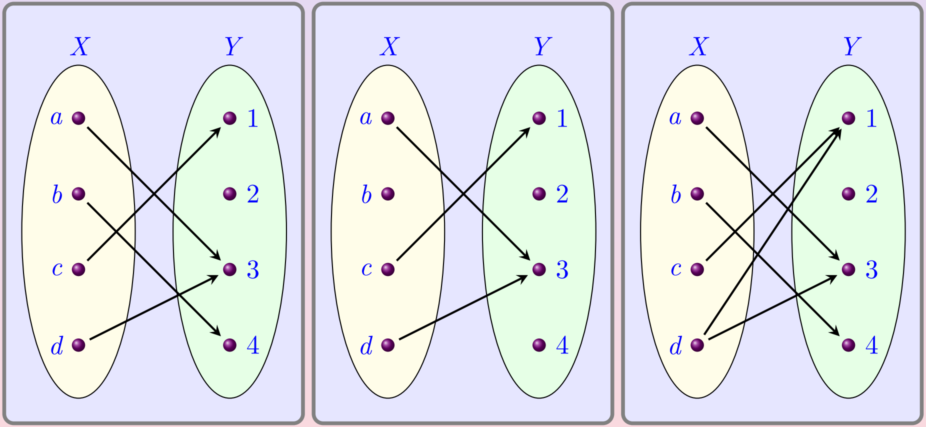
集合X和Y分别称为函数 f 的定义域(domain)和陪域(codomain). 与 $x \in X$有关系 f 的$y \in Y$ 记为: $f(x)$, 即$y = f(x)$.

Remark

不满足上述两个条件的任何一个, 将不能构成函数:

1. 非完全的: $\exists x \in X \wedge \forall y \in Y, \langle x, y\rangle \notin f$,
   表示函数在有些点没有定义. 将满足条件②不满足条件①的关系称为部分函数(partial function);
2. 一对多: if $\exists x \in X \wedge y, z \in Y \wedge y \neq z \wedge \langle x, y\rangle \in f \wedge \langle x, z\rangle \in f$.
   表示函数在有些点可能对应多值. 将满足条件①不满足条件②的关系称为多值函数(multivalued function).

图例



(a)函数 (b)部分函数 (c)多值函数

相关定义

Definition (函数的集合)

- 记$Y^X \triangleq \{f \mid f : X \to Y\}$为所有的从$X$到$Y$的函数的集合;
- 若X和Y为有限集合,则$|Y^X| = |Y|^{|X|}$;
- $|Y|^{|X|} < 2^{|X| \times |Y|}$, $\because Y^X \subsetneq \mathcal{P}(X \times Y)$.

Definition (函数相等)

称两函数 $f, g : X \to Y$ 相等(记为$f = g$), iff, 对应的函数关系相等. 即函数f, g在任一点具有相同的函数值, $\forall x \in X,\ f(x) = g(x)$.

Example

- $\left|\{0, 1\}^{\{0,1\}^n}\right| = 2^{2^n}$;
- $F \Leftrightarrow G$ iff $I(F) = I(G)$;

函数的像和逆像

Definition (像, image; 逆像, inverse image; 值域, range)

设$f: X \to Y$是X到Y上的函数，$A \subseteq X$, $B \subseteq Y$:

- A 在 f 下的像：
  $f(A) \triangleq \{\, y \mid \exists x \in A \wedge y = f(x) \,\} = \{\, f(x) \mid x \in A \,\}$;
- B 在 f 下的逆像：
  $f^{-1}(B) \triangleq \{\, x \mid \exists y \in B \wedge y = f(x) \,\} = \{\, x \mid f(x) \in B \,\}$;
- $f(X)$称为函数 f 的值域(range).

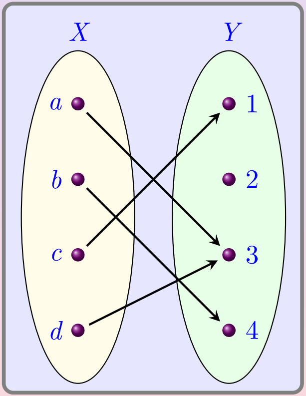
Example

- $f: \mathbb{R} \to \mathbb{R}$, 若 f 是连续函数，则对任意的开区间$]a, b[$, 有$f(]a, b[)$也是开区间；
- 重言式集合 $\mathcal{T} = \mathscr{I}^{-1}(\{\, \mathbb{T} \,\})$.

注意

f^{-1}有两重含义，求逆像和逆函数(反函数)，求逆像的作用对象只能是集合：$\cancel{f^{-1}(a)}$[a是元素]; 只有当 f 的逆函数存在时，f^{-1}才能作用元素.

Example



Example

- $f(\{a, d\}) = \{3\}$;
- $f(\{d\}) = \{3\}$;
- $f^{-1}(\{3\}) = \{a, d\}$;
- $f^{-1}(\{2\}) = \varnothing$.

相关性质

Theorem

设 $f\colon X \to Y$是 X 到 Y 上的函数，$A \subseteq X$，$B \subseteq Y$，则：

1. $A \subseteq f^{-1}(f(A))$;
2. $f(f^{-1}(B)) \subseteq B$.

Proof.

1. $\forall x \in A$，则 $f(x) \in f(A)$，
   $\therefore\ x \in f^{-1}(f(A)) = \{\, x' \mid f(x') \in f(A) \,\}$;
2. $\forall y \in f(f^{-1}(B))$，则 $\exists x \in f(B) \wedge y = f(x)$，
   而$\because x \in f^{-1}(B)$，$\therefore\ f(x) \in B$，即$y \in B$.

$\square$

Remark

直观上，求像：对集合缩小；求逆像：对集合放大.

常用的函数

Description

1. $\mathbb{1}_X: X \to X,\ x \mapsto x$. 恒等函数;
2. $b: X \to Y,\ x \mapsto b$. 常数函数;
3. $s: \mathbb{N} \to \mathbb{N},\ n \mapsto n+1$. 后继函数;
4. $f: X_1 \times X_2 \times \cdots \times X_n \to Y$. n元函数;
5. $p_i: X_1 \times X_2 \times \cdots \times X_n \to X_i, \langle x_1, x_2, \ldots, x_n \rangle \mapsto x_i$. 投影函数;
6. $X \to \mathcal{P}(X \times Y),\ x \mapsto \{x\} \times Y$. 截痕函数.

函数的合成

Definition (合成函数(复合), Composite function)

设 $f\colon\ X\to Y$ 是 X 到 Y 的函数, $g\colon\ Y\to Z$ 是 Y 到 Z 的函数, 关系 f 和 g 的合成关系 $f\circ g$ 也是函数, 记为: $g\circ f\colon\ X\to Z,\ x\mapsto g(f(x))$, 称为 f 和 g 的合成函数. (注意: 习惯上函数合成的写法与关系合成相反.)

函数的合成.

- 合成交换图

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "g\circ f"'] & Y \arrow[d, "g"] \\ & Z \end{tikzcd}$$

Proof.

1. $\because$ f 是函数,
   $\therefore$ $\forall x\in X,\ \exists! y\in Y, \langle x,y\rangle\in f,\ \ y=f(x)$;
2. $\because$ g 是函数,
   $\therefore$ $\forall y\in Y,\ \exists! z\in Z, \langle y,z\rangle\in g,\ \ z=g(y)$;
3. 所以, 对合成关系 $f\circ g\subseteq X\times Z$,
   $\forall x\in X,\ \exists! z\in Z,\ \ z=g(y)=g(f(x))$,
   即合成关系 $f\circ g$ 是 X 到 Z 的函数.
   记为 $g\circ f\colon X\to Z,\ \ x\mapsto g(f(x))$,
   即 $z=g(y)=g(f(x))=g\circ f(x)$.

相关性质

Theorem

设$f: X \to Y,\ g: Y \to Z,\ h: Z \to W$:

- $\mathbb{1}_Y \circ f = f \circ \mathbb{1}_X = f$;
- $h \circ (g \circ f) = (h \circ g) \circ f$, (结合律).

Notation(合成运算的幂)

- 设 $f: X \to X,\ n \in \mathbb{N}$,
  1. $f^0 = \mathbb{1}_X$;
  2. $f^{n+1} = f \circ f^n$;
- 和关系合成一样: $\forall m, n \in \mathbb{N},\ f^m \circ f^n = f^{m+n};\ (f^m)^n = f^{mn}$.

单射、满射和双射

Definition

设$f:\ X \to Y$:

- 若 $f(X) = Y$, 称 f 为满射(onto);
- 若 $\forall x, x' \in X,\ f(x) = f(x') \to x = x'$ (或者 $\forall x, x' \in X,\ x \neq x' \to f(x) \neq f(x')$), 称 f 为单射(one to one);
- 若f 既是单射又是满射, 则称 f 为双射(bijection).

Example

设$|X| = m,\ |Y| = n,\ m, n \in \mathbb{N}$:

- $\mathbb{1}_X$是双射;
- f 是单射, 则, $|f(X)| = |X| = m$, 因此, $|X| \leq |Y|$;
- f 是满射, 则, $|Y| = |f(X)| \leq |X|$, 因此, $|Y| \leq |X|$;
- f 是双射, 则, $|X| = |Y|$.

Examples

Definition (置换, Permutation)

有限集合上的双射称为置换. 通常表示为集合$\{1, 2, \ldots, n\}$上的双射.

Example

- 记$P_n = \{p \mid p$ 是 $\{1, 2, \ldots, n\}$上的置换.$\}$, 则$|P_n| = n!$;
- 设$f: X \to Y$, $|X| = m$, $|Y| = n$:
  - 设$m \leq n$, 则X 到 Y 上的单射共有$C_n^m m!$个;
  - 设$m \geq n$, 则X 到 Y 上的满射个数等于 m 个元素的集合共有多少个 n 分区的个数. (即, m 个不同的小球放入 n 个不同盒子, 且不允许有空盒的放球方案数.)

相关性质(1/2)

Theorem

设$f: X \to Y,\ g: Y \to Z$:

1. 若 $f,\ g$ 是单射,则 $g \circ f$ 也是单射;
2. 若 $f,\ g$ 是满射,则 $g \circ f$ 也是满射;
3. 若 $f,\ g$ 是双射,则 $g \circ f$ 也是双射.

①的证明.

1. 设$x, x' \in X$, 若$g \circ f(x) = g \circ f(x')$, 即 $g(f(x)) = g(f(x'))$,
2. $\because\ g$ 是单射, $\therefore\ f(x) = f(x')$,
3. 又 f 是单射, $\therefore\ x = x'$, 则$g \circ f$也是单射.

□

相关性质(2/2)

Theorem

设$f: X \to Y,\ g: Y \to Z$:

1. 若 $g \circ f$ 是单射,则 f 是单射;
2. 若 $g \circ f$ 是满射,则 g 是满射;
3. 若 $g \circ f$ 是双射,则 f 是单射且 g 是满射.

Proof.

1. $\forall x, x' \in X$, 若$f(x) = f(x')$, $\because g$ 是函数, $\therefore\ g(f(x)) = g(f(x'))$, 即 $g \circ f(x) = g \circ f(x')$, 又$\because\ g \circ f$ 是单射, $\therefore\ x = x'$, 故 f 是单射;
2. $\forall z \in Z$, $\because\ g \circ f$ 是满射, $\therefore\ \exists x \in X$, 使得$z = g \circ f(x)$, 即 $z = g(f(x))$, 则存在$y = f(x) \in Y$, 使得 $z = g(y)$, 故 g 是满射;
3. 由①,②可直接得出.

$\square$

单射、满射的充要条件

Theorem

设 $f: X \to Y$:

1. f 是单射, iff, $\exists g: Y \to X \wedge g \circ f = \mathbb{1}_X$, 称 g 为 f 的*左逆函数*;
2. f 是满射, iff, $\exists g: Y \to X \wedge f \circ g = \mathbb{1}_Y$, 称 g 为 f 的*右逆函数*.

①的证明.

1. $\Longleftarrow$ 设$f(x) = f(x')$, 则 $g(f(x)) = g(f(x'))$, $\therefore$ $g \circ f(x) = g \circ f(x')$, 即 $\mathbb{1}_X(x) = \mathbb{1}_X(x')$, $x = x'$, 所以 f 是单射;
2. $\Longrightarrow$ 构造$g: Y \to X$: 任取 $x_0 \in X$, 定义

$$g(y) = \begin{cases} x & \text{if} \quad y \in f(X) \wedge f(x) = y; \\ x_0 & \text{if} \quad y \notin f(X). \end{cases}$$

$\because$ f 是单射, 若$y \in f(X)$, $\exists! x \in X$, $y = f(x)$, $\therefore$ g 是well-defined;
且$\forall x \in X$, $g \circ f(x) = g(f(x)) = g(y) = x$, 故$g \circ f = \mathbb{1}_X$.

$\square$

注意:从构造过程可知,函数的左、右逆元不唯一.

双射的充要条件

Theorem

设 $f: X \to Y$, f 是双射, iff, $\exists! g: Y \to X \wedge g \circ f = \mathbb{1}_X \wedge f \circ g = \mathbb{1}_Y$, 称 g 为 f 的逆函数(反函数), 记为 $g = f^{-1}$. f 为可逆的.

Proof.

1. $\Longrightarrow$
   $\because$ f 是双射, 由前述定理, $\exists g, g' : Y \to X$, $g \circ f = \mathbb{1}_X \wedge f \circ g' = \mathbb{1}_Y$;
   $\therefore$
   $$\begin{aligned} g &= g \circ \mathbb{1}_Y \\ &= g \circ (f \circ g') \\ &= (g \circ f) \circ g' \\ &= \mathbb{1}_X \circ g' \\ &= g' \end{aligned}$$
   $\therefore$ $g = g'$, 即 g 存在并且唯一;
2. $\Longleftarrow$
   由前述定理可知, f 既是单射也是满射, 故 f 是双射.

□

逆函数的性质

Theorem

设 $f: X \to Y,\ g: Y \to Z,\ f,\ g$ 是双射, 则

1. $(f^{-1})^{-1} = f$;
2. $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

②的证明.

1. 由逆函数和合成定义, $f^{-1} \circ g^{-1}:\ Z \to X$;
2. 由合成的结合律,

$$(f^{-1} \circ g^{-1}) \circ (g \circ f) = f^{-1} \circ (g^{-1} \circ g) \circ f = \mathbb{1}_X,$$

$$(g \circ f) \circ (f^{-1} \circ g^{-1}) = g \circ (f \circ f^{-1}) \circ g^{-1} = \mathbb{1}_Y,$$

$$\therefore\ (g \circ f)^{-1} = f^{-1} \circ g^{-1}.$$

□

Examples

Example

设 $f: \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$, $x \mapsto x^2$; $g: \mathbb{R}^+ \cup \{0\} \to \mathbb{R}, x \mapsto \sqrt{x}$ ($\sqrt{x}$ 是 x 的非负平方根), 求 $f \circ g$ 和 $g \circ f$;

1. 函数 g 的值域和 f 的定义域为相同集合,根据合成定义,
   $\forall x \in \mathbb{R}^+ \cup \{0\}$, $(f \circ g)(x) = f(g(x)) = f(\sqrt{x}) = (\sqrt{x})^2 = x$;
   $\therefore\ f \circ g: \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+ \cup \{0\}$, $x \mapsto x$.
2. 函数 f 的值域和 g 的定义域为相同集合,根据合成定义,
   $\forall x \in \mathbb{R}$, $(g \circ f)(x) = g(f(x)) = g(x^2) = (\sqrt{x^2})$;
   $\therefore\ g \circ f: \mathbb{R} \to \mathbb{R}$, $(g \circ f)(x) = \begin{cases} x & \text{if} \quad x \geq 0 \\ -x & \text{if} \quad x < 0 \end{cases}$.

Example

设 $f: \mathbb{R}^+ \to \mathbb{R}$, $f(x) = \ln(x)$, 则 $f^{-1}: \mathbb{R} \to \mathbb{R}^+$, $f^{-1}(x) = e^x$, 则:

- $e^{\ln x} = x \wedge \ln(e^x) = x$;
- $(e^{\ln x})' = x' \implies e^{\ln x} \cdot (\ln x)' = 1 \implies x \cdot (\ln x)' = 1 \implies (\ln x)' = 1/x$.

相关性质(1/2)

Theorem

设 $f: X \to Y$. f 是单射, iff, $\forall A \subseteq X,\ f^{-1}(f(A)) = A$.

Proof.

- $\Longrightarrow$
  1. $A \subseteq f^{-1}(f(A))$✓ 需证 $f^{-1}(f(A)) \subseteq A$?
  2. $\forall x \in f^{-1}(f(A)),\ f(x) \in f(A),\ \therefore\ \exists x' \in A \wedge f(x) = f(x')$;
  3. 而 f 是单射, $\therefore x = x' \in A,\ \therefore\ f^{-1}(f(A)) \subseteq A$.
- $\Longleftarrow$
  1. 设 $f(x) = f(x')$, 需证 $x = x'$;
  2. 令 $A = \{x\}$, 则 $f^{-1}(f(A)) = f^{-1}(\underline{f(\{x\})}) = f^{-1}(\underline{\{f(x)\}}) = A = \{x\}$;
  3. $f(x') = f(x) \in \{f(x)\}$, 则 $x' \in f^{-1}(\{f(x)\}) = \{x\}$;
  4. $\therefore\ x = x'$, 故 f 是单射.

$\square$

相关性质(2/2)

Theorem

设 $f: X \to Y$. f 是满射, iff, $\forall B \subseteq Y,\ f(f^{-1}(B)) = B$.

Proof.

- $\Longrightarrow$
  1. $f(f^{-1}(B)) \subseteq B$✓ 需证 $B \subseteq f(f^{-1}(B))$?
  2. $\forall y \in B$, $\because$ f 是满射, $\therefore x \in X,\ f(x) = y \in B$;
  3. $\therefore\ x \in f^{-1}(B)$, 即 $f(x) \in f(f^{-1}(B))$;
  4. 即 $y \in f(f^{-1}(B))$, $\therefore\ B \subseteq f(f^{-1}(B))$.
- $\Longleftarrow$ $f(X) = Y$?
  1. 令 $B = Y$, 则 $Y = f(f^{-1}(Y))$;
  2. 又 $f^{-1}(Y) \subseteq X$, $\therefore\ f(f^{-1}(Y)) \subseteq f(X)$;
  3. $\therefore\ Y = f(f^{-1}(Y)) \subseteq f(X) \subseteq Y$,
  4. $\therefore\ Y = f(X)$, 故 f 是满射.

$\square$

Outline

1 函数、复合函数

2 函数的递归定义
- 自然数集合上的递归函数
- Euclid算法和尾递归
- List集合上的递归函数
- Ackermann函数
- 高阶函数

自然数集上的递归函数

Example

$f:\ \mathbb{N} \to \mathbb{N}$ 递归定义如下，求 f 的解析式，

1. $f(0) = 3$;
2. $f(n+1) = 2f(n) + 3$.

Solution.

$$
\begin{aligned}
f(n) &= 2 * f(n-1) + 3 \\
&= 2 * (2 * f(n-2) + 3) + 3 \\
&= 2 * (2 * (2 * f(n-3) + 3) + 3) + 3 \\
&= 2^3 * f(n-3) + 2^2 * 3 + 2 * 3 + 3 \\
&\cdots\cdots \\
&= 2^n * f(0) + 2^{n-1} * 3 + \cdots + 2^2 * 3 + 2 * 3 + 3 \\
&= 3 * (2^n + 2^{n-1} + \cdots + 2^2 + 2 + 1) \\
&= 3 * (2^{n+1} - 1)
\end{aligned}
$$

□

对应的程序

Recursion

```
int f(int n)
{
    if (n < 0) error();
    if (n == 0) return 3;
    return 2 * f(n-1) + 3;
}
```

For-loops

```
int f(int n)
{
    int result = 3, i;
    if (n < 0) error();
    for (i = 1; i<= n; i++)
      result = 2 * result + 3;
    return result;
}
```

Example

Example

Fibonacci序列递归定义如下,求其解析式:

1. $f(0) = 0,\ f_1 = 1$;
2. $f_n = f_{n-1} + f_{n-2}$.

Solution.

1. 求特征方程 $r^2 - r - 1 = 0$ 的根 $r_1,\ r_2$;
2. 序列的通项 $f_n = \alpha_1 r_1^n + \alpha_2 r_2^n$;
3. 根据初始条件①求出待定系数 α_1 和 α_2;
4. $$\therefore\ f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

$\square$

对应的程序

Recursion

```
int f(int n)
{
  if (n == 0) return 0;
  if (n == 1) return 1;
  return f(n-2) + f(n-1);
}
```

For-loops

```
int f(int n)
{
   int x = 0;    /* f(n-2) */
   int y = 1;    /* f(n-1) */
   if (n == 0) return 0;
   if (n < 0) error();
   for (i = 1, i <= n-1; i++) {
     int z = x + y;
     x = y;
     y = z;
     }
   return y;
}
```

Euclid算法

Example (Greatest Common Divisor)

$\gcd: \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ 递归定义如下：

1. $\gcd(0,n)=n$;
2. $\gcd(m,n)=gcd((n \mod m),m)$.

Proof.

1. 若 k 是 m, n 的公约数，即 $(k|m)\wedge(k|n)$,
   则 $\exists p,q,\ m=kp\wedge n=kq$; 设 $n \mod m=t$,
   则 $n=rm+t$, $\therefore\ t=n-rm=k(q-rp)$; 即 $k|t$,
   $\therefore\ \gcd(m,n)\le\gcd(t,m)=\gcd(n \mod m,m)$;
2. 若 k 是 t, m 的公约数，$\because\ n=rm+t$, $\therefore\ k|n$,
   $\therefore\ \gcd(t,m)\le\gcd(m,n)$, 即 $\gcd(n \mod m,m)\le\gcd(m,n)$.

□

对应的程序

Tail Recursion

```
  int gcd(int m, int n)
{
    if (m == 0) return n;
    return gcd(n % m, m);
}
```

Ex:

```
gcd(18, 12)
  ↳gcd(12, 18)
    ↳gcd(6, 12)
      ↳gcd(0, 6)
        ↳6
```

While-loops

```
  int gcd(int m, int n)
{
    int tmp;
    while (m != 0) {
    tmp = m;
    m = n % m;
    m = tmp;
  }
  return n;
}
```

List上的递归函数

Example (length)

$\mathrm{length}: \ \Sigma^* \to \mathbb{N}$ 可递归定义如下：

1. $\mathrm{length}(\varepsilon) = 0;$
2. $\mathrm{length}(a \cdot s) = 1 + \mathrm{length}(s).$

Example (length尾递归定义)

$f: \ \mathbb{N} \times \Sigma^* \to \mathbb{N}$ 可递归定义如下：

1. $f(m, \varepsilon) = m;$
2. $f(m, a \cdot s) = f(m+1, s).$

则，$\mathrm{length}(s) = f(0, s).$

对应的程序

Tail Recursion

```
  int f(int m, Σ List s)
{
   if (s == ε) return m;
   return f(m+1,tl(s));
   /* tl(a · s) = s */
}

  int length(Σ List s)
{
   return f(0, s);
}
```

While-loops

```
  int length(Σ List s)
{
   int m = 0;
   while (s != ε) {
   m = m + 1;
   s = tl(s);
   }
   return m;
}

  int length(char *s)
{
   int tmp = 0;
   while (*s++) tmp++;
   return tmp;
}
```

Ackmann函数

Example

Ackermann函数递归定义如下：

$$A(m,n)=\begin{cases} n+1 & \text{if} \quad m=0 \\ A(m-1,1) & \text{if} \quad m>0 \wedge n=0 \\ A(m-1,A(m,n-1)) & \text{if} \quad m>0 \wedge n>0 \end{cases}$$

Description

- 处处有定义;
- m 足够小时,增长缓慢,当$m \geq 4$时,呈指数增长.
  $A(4,2) \doteq 2 \times 10^{19728}$;
- 递归函数,但非原始递归函数(primitive recursive function);
- 不能够用while-loops表达;
- 由于其深度递归性(deep recursion),该函数常用于测试编译器对递归的优化性能.

A(4,3)的计算

```
A(4, 3) = A(3, A(4, 2))
        = A(3, A(3, A(4, 1)))
        = A(3, A(3, A(3, A(4, 0))))
        = A(3, A(3, A(3, A(3, 1))))
        = A(3, A(3, A(3, A(2, A(3, 0)))))
        = A(3, A(3, A(3, A(2, A(2, 1)))))
        = A(3, A(3, A(3, A(2, A(1, A(2, 0))))))
        = A(3, A(3, A(3, A(2, A(1, A(1, 1))))))
        = A(3, A(3, A(3, A(2, A(1, A(0, A(1, 0)))))))
        = A(3, A(3, A(3, A(2, A(1, A(0, A(0, 1)))))))
        = A(3, A(3, A(3, A(2, A(1, A(0, 2))))))
        = A(3, A(3, A(3, A(2, A(1, 3)))))
        = A(3, A(3, A(3, A(2, A(0, A(1, 2))))))
        = A(3, A(3, A(3, A(2, A(0, A(0, A(1, 1)))))))
        = A(3, A(3, A(3, A(2, A(0, A(0, A(0, A(1, 0))))))))
        = A(3, A(3, A(3, A(2, A(0, A(0, A(0, A(0, 1))))))))
        = A(3, A(3, A(3, A(2, A(0, A(0, A(0, 2)))))))
        = A(3, A(3, A(3, A(2, A(0, A(0, 3))))))
        = A(3, A(3, A(3, A(2, A(0, 4)))))
        = A(3, A(3, A(3, A(2, 5))))
        = ...
        = A(3, A(3, A(3, 13)))
        = ...
        =  A(3, A(3, 65533)) /* A(3, 65533) = 2^(65533+3)-3 */
        = ...
```

对应的程序

Recursion

```
int ack(int m, int n)
{
 if (m == 0) return n+1;
 if (n == 0) return ack(m-1,1);
 return ack(m-1, ack(m, n-1));
}
 /* No linear recursion */
```

Partially while-loops

```
int ack(int m, int int n)
{
  while (m != 0) {
   if (n == 0)
     n = 1;
   else
     n = ack(m, n-1);
   m = m - 1;
   }
  return n+1;
}
```

高阶函数

Example

1. 设 $C(\mathbb{R})$ 是实数上的连续函数的集合，定义函数 g 如下：

$$g:\ C(\mathbb{R}) \to \mathbb{R}^{\mathbb{R}^2},\ f \mapsto \left(\langle x, y\rangle \mapsto \int_x^y f(t)\,dt\right),$$

则 $(g(\sin))(0,1)$ 是 $\sin$ 在 $[0,1]$ 区间上的积分；

2. 定义函数 fold-left : $X^{X\times\Sigma} \to X^{X\times\Sigma^*}$,
$f \mapsto \big(\langle x, a_1 a_2 \cdots a_n \mapsto f(\cdots f(f(x,a_1),a_2)\cdots, a_n)\big)$;
则，设 $\Sigma = \mathbb{Z}$, $X = \mathbb{Z}$,
   - 若 $f(x,y) = x + y$, 则：
     $\text{sum}(a_1 a_2 \cdots a_n) = \big(\text{fole-left}(f)\big)(0, a_1 a_2 \cdots a_n)$;
   - 若 $f(x,y) = 1 + x$, 则：
     $\text{length}(a_1 a_2 \cdots a_n) = \big(\text{fole-left}(f)\big)(0, a_1 a_2 \cdots a_n)$.

小结

Remark

- 函数能够递归定义的首要条件是其定义域必须有一个递归结构;
- 一般是按照定义域集合的递归定义,对该集合中的每个元素进行析构;
- recursion $\neq$ loops;
- tail recursion 能转换为 while-loop;
- recursion: 简单、清晰,但时间开销较大;
- loop: 较为复杂,但效率较高.

本章小节

1. 函数、复合函数
   - 函数的定义
   - 复合函数
   - 单射、满射和双射
2. 函数的递归定义
   - 自然数集合上的递归函数
   - Euclid算法和尾递归
   - List集合上的递归函数
   - Ackermann函数
   - 高阶函数

Reference books

Kenneth H. Rosen.
《离散数学及其应用》(原书第8版).
机械工业出版社.

刘玉珍
《离散数学》.
武汉大学出版社.

王汉飞
《离散数学》讲义.