

理论计算机基础

Little Wolf

2024 年 10 月 11 日

目录

1	\mathcal{S} 程序和可计算函数	2
2	原始递归函数	9
3	通用程序	12

1 \mathcal{S} 程序和可计算函数

题目. P_9 1.3.6 对程序:

\mathcal{P}_2 :

```

IF  $X \neq 0$  GOTO  $A$ 
 $Y \leftarrow Y + 1$ 
 $Z \leftarrow Z + 1$ 
IF  $Z \neq 0$  GOTO  $E$ 
 $[A]X \leftarrow X - 1$ 
 $[B]IF \ X \neq 0 \text{ GOTO } B$ 
```

给出它从输入变量 X 分别等于0, 1, 5的初始状态开始的计算.

解答. (1) 思路: $X = 0$, 不跳转到 $[A]$, 之后 $Y = 1, Z = 1$, 此时 $Z \neq 0$, 跳转到 $[E]$, 结束程序. 程序结束时, 输出变量 $Y = 1$.

计算:

- (1, $\{X = 0, Y = 0, Z = 0\}$)
- (2, $\{X = 0, Y = 0, Z = 0\}$)
- (3, $\{X = 0, Y = 1, Z = 0\}$)
- (4, $\{X = 0, Y = 1, Z = 1\}$)
- (7, $\{X = 0, Y = 1, Z = 1\}$) (终点快相)

(2) 思路: $X = 1$, 跳转到 $[A]$, 执行 $X \leftarrow X - 1$ 后, $X = 0$, 不执行 $[B]$, 结束程序. 程序结束时, 输出变量 $Y = 0$.

计算:

- (1, $\{X = 1, Y = 0, Z = 0\}$)
- (5, $\{X = 1, Y = 0, Z = 0\}$)
- (6, $\{X = 0, Y = 0, Z = 0\}$)
- (7, $\{X = 0, Y = 0, Z = 0\}$)

(3) 思路: $X = 5$, 跳转到 $[A]$, 执行 $X \leftarrow X - 1$ 后, $X = 4$, 执行 $[B]$, 进入死循环. 程序结束时, 输出变量 $Y = 0$.

计算:

- (1, $\{X = 5, Y = 0, Z = 0\}$)
- (5, $\{X = 5, Y = 0, Z = 0\}$)
- (6, $\{X = 4, Y = 0, Z = 0\}$)
- (6, $\{X = 4, Y = 0, Z = 0\}$)
- ... (死循环)

□

题目. P_9 1.3.7 对程序

\mathcal{P}_3 :

```

 $X_1 \leftarrow X_1 + 1$ 
 $X_1 \leftarrow X_1 + 1$ 
 $[A]X_1 \leftarrow X_1 - 1$ 
  IF  $X_1 \neq 0$  GOTO  $C$ 
 $[B]Z \leftarrow Z + 1$ 
  IF  $Z \neq 0$  GOTO  $B$ 
 $[C]X_1 \leftarrow X_1 - 1$ 
  IF  $X_1 \neq 0$  GOTO  $A$ 
  IF  $X_2 \neq 0$  GOTO  $D$ 
 $Y \leftarrow Y + 1$ 
 $[D]Y \leftarrow Y$ 

```

设输入变量的初始状态的值如下:

(1) $X_1 = 2, X_2 = 0$

(1) $X_1 = 4, X_2 = 3$

(1) $X_1 = 1, X_2 = 4$

写出计算

解答. (1) 分析: 执行了 $[A]$ 后, $X_1 = 3$, 跳转到 C , 之后 $X_1 = 2$, 跳转回 $[A]$, $X_1 = 1$, 再跳转到 $[C]$, $X_1 = 0$, 而 $X_2 = 0$, 执行 $Y \leftarrow Y + 1$, 之后进入空指令 $[D]$. 最后输出变量 $Y = 1$.

计算:

- (1, $\{X_1 = 2, X_2 = 0, Y = 0, Z = 0\}$)
- (2, $\{X_1 = 3, X_2 = 0, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 4, X_2 = 0, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 3, X_2 = 0, Y = 0, Z = 0\}$)
- (7, $\{X_1 = 3, X_2 = 0, Y = 0, Z = 0\}$)
- (8, $\{X_1 = 2, X_2 = 0, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 2, X_2 = 0, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 1, X_2 = 0, Y = 0, Z = 0\}$)
- (7, $\{X_1 = 1, X_2 = 0, Y = 0, Z = 0\}$)
- (8, $\{X_1 = 0, X_2 = 0, Y = 0, Z = 0\}$)
- (9, $\{X_1 = 0, X_2 = 0, Y = 0, Z = 0\}$)
- (10, $\{X_1 = 0, X_2 = 0, Y = 0, Z = 0\}$)
- (11, $\{X_1 = 0, X_2 = 0, Y = 1, Z = 0\}$)
- (12, $\{X_1 = 0, X_2 = 0, Y = 1, Z = 0\}$)

(2) 思路: 执行 $[A]$ 之后, $X_1 = 5$, 跳转 $[C]$, 之后 $X_1 = 4$, 跳转回 $[A]$, 在 $[C]$, $[A]$ 间来回跳转, 根据 X_1 的奇偶性, 最后在执行 $[C]$ 的第一步之后 $X_1 = 0$, $X_2 = 3 \neq 0$, 跳转到空指令 D . 最后输出变量 $Y = 0$.

计算:

- (1, $\{X_1 = 4, X_2 = 3, Y = 0, Z = 0\}$)
- (2, $\{X_1 = 5, X_2 = 3, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 6, X_2 = 3, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 5, X_2 = 3, Y = 0, Z = 0\}$)
- (7, $\{X_1 = 5, X_2 = 3, Y = 0, Z = 0\}$)
- (8, $\{X_1 = 4, X_2 = 3, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 4, X_2 = 3, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 3, X_2 = 3, Y = 0, Z = 0\}$)
- (7, $\{X_1 = 3, X_2 = 3, Y = 0, Z = 0\}$)
- (8, $\{X_1 = 2, X_2 = 3, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 2, X_2 = 3, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 1, X_2 = 3, Y = 0, Z = 0\}$)
- (7, $\{X_1 = 1, X_2 = 3, Y = 0, Z = 0\}$)
- (8, $\{X_1 = 0, X_2 = 3, Y = 0, Z = 0\}$)
- (9, $\{X_1 = 0, X_2 = 3, Y = 0, Z = 0\}$)
- (11, $\{X_1 = 0, X_2 = 3, Y = 0, Z = 0\}$)
- (12, $\{X_1 = 0, X_2 = 3, Y = 0, Z = 0\}$)

(3) 思路: 执行[A]之后, $X_1 = 2$, 跳转[C], 之后 $X_1 = 1$, 跳转回[A], $X_1 = 0$, 执行[B], $Z = 1$, 在[B]中进入死循环. 最后输出变量 $Y = 0$.

计算:

- (1, $\{X_1 = 1, X_2 = 4, Y = 0, Z = 0\}$)
- (2, $\{X_1 = 2, X_2 = 4, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 3, X_2 = 4, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 2, X_2 = 4, Y = 0, Z = 0\}$)
- (7, $\{X_1 = 2, X_2 = 4, Y = 0, Z = 0\}$)
- (8, $\{X_1 = 1, X_2 = 4, Y = 0, Z = 0\}$)
- (3, $\{X_1 = 1, X_2 = 4, Y = 0, Z = 0\}$)
- (4, $\{X_1 = 0, X_2 = 4, Y = 0, Z = 0\}$)
- (5, $\{X_1 = 0, X_2 = 4, Y = 0, Z = 0\}$)
- (6, $\{X_1 = 0, X_2 = 4, Y = 0, Z = 1\}$)
- (5, $\{X_1 = 0, X_2 = 4, Y = 0, Z = 1\}$)
- (6, $\{X_1 = 0, X_2 = 4, Y = 0, Z = 2\}$)
- ... (进入死循环)

□

题目. P_{12} 1.1 写出计算下述函数的 \mathcal{S} 程序(允许使用宏指令):

- (1) $f(x) = \lfloor x/2 \rfloor$ (向下取整)
- (2) x 偶数, $f(x) = 1$; x 奇数, $f(x)$ 无定义.

解答. (1) 思路: 除以2可以用一直减2表示.

\mathcal{P}_1 :

```

    Z ← Z + 1
    X ← X + 1  (+1的目的是为了保证2的输出是1, 以此类推)
[A] X ← X - 1
    X ← X - 1
    IF X ≠ 0 GOTO B
    IF Z ≠ 0 GOTO E
[B] Y ← Y + 1
    IF Y ≠ 0 GOTO A

```

使用宏指令的版本:

\mathcal{P}_1^* :

```

    X ← X + 1
[A] X ← X - 2
    IF X ≠ 0 GOTO B
    GOTO E
[B] Y ← Y + 1
    GOTO A

```

(2) 思路: 对输入的 X , 循环减两次1, 但每次都检查 X 是否是0, 来判断奇偶性, 为了兼容0, 首先加上1. 简单来说, 就是看减去的是奇数个还是偶数个1来进行出口的分类.

\mathcal{P}_2^* :

```

    X ← X + 1
[A] X ← X - 1
    IF X = 0 GOTO B
    X ← X - 1
    IF X ≠ 0 GOTO A
    GOTO C
[B] Y ← Y + 1
    GOTO E
[C] Z ← Z + 1
    IF Z ≠ 0 GOTO C

```

如果不允许判断 $X = 0$, 可以这么写:

\mathcal{P}_2^* :

```

    X ← X + 1
[A] X ← X - 1
    IF X ≠ 0 GOTO B
    GOTO C  (偶数出口)

```

```

[B]  $X \leftarrow X - 1$ 
    IF  $X \neq 0$  GOTO A
    GOTO D (奇数出口)
[C]  $Y \leftarrow Y + 1$ 
    GOTO E
[D]  $Z \leftarrow Z + 1$ 
    IF  $Z \neq 0$  GOTO D (死循环)

```

□

题目的注记. 可供使用的宏指令:

- GOTO A
- $V \leftarrow V'$
- 判断 $X = 0$ 和跳转

题目. P_{12} 1.2 给出下列程序 \mathcal{P} 计算的函数 $\psi_{\mathcal{P}}^{(1)}(x)$:

```

(1) [A]  $X \leftarrow X + 1$ 
       $X \leftarrow X - 1$ 
      IF  $X \neq 0$  GOTO A
(2) [A]  $X \leftarrow X - 1$ 
      IF  $X = 0$  GOTO A
       $X \leftarrow X - 1$ 
      IF  $X \neq 0$  GOTO A
(3) 空程序

```

解答. (1) $\psi_{\mathcal{P}_1}^{(1)}(x) = \begin{cases} \uparrow \text{ (未定义)} & \text{if } x \in \mathbb{N}^*, \\ 0 & \text{if } x = 0. \end{cases}$

(2) $\psi_{\mathcal{P}_1}^{(1)}(x) = \begin{cases} 0, & x \text{ 是正偶数} \\ \uparrow, & x = 0 \text{ 或 } x \text{ 是奇数} \end{cases}$

(3) $\psi_{\mathcal{P}_1}^{(1)}(x) = 0, \forall x \in \mathbb{N}$

□

题目. P_{12} 1.3 证明下面的函数是部分可计算的:

(1) $x_1 + x_2$; (2) $x_1 - x_2$; (3) $x_1 x_2$; (4) 空函数

解答. 要证明一个函数是部分可计算的, 实际上就是可以用 S 函数把它写出来, "部分" 指的是可以在某些点上没有定义

(1) 思路: x_2 一直减1, x_1 一直加1, 直到减到零

```

 $Y \leftarrow X_1$ 
 $Z \leftarrow X_2$ 
[A] IF  $Z \neq 0$  GOTO B
    GOTO E

```

```

[B]  $Z \leftarrow Z - 1$ 
     $Y \leftarrow Y + 1$ 
    GOTO A

```

(2) 思路: 如果 $x_1 < x_2$, 那么无定义, 因此看谁先减到0.

```

 $Z_1 \leftarrow X_1$ 
 $Z_2 \leftarrow X_2$ 
[A] IF  $Z_1 \neq 0$  GOTO B
    GOTO  $D_1$ 
[B] IF  $Z_2 \neq 0$  GOTO C
    GOTO  $D_2$ 
[C]  $Z_1 \leftarrow Z_1 - 1$ 
     $Z_2 \leftarrow Z_2 - 1$ 
    GOTO A
[ $D_1$ ]  $Y \leftarrow Z_2$ 
    GOTO E
[ $D_2$ ]  $Y \leftarrow Z_1$ 
    GOTO E

```

(3) 思路: 如果有0, 返回0; 如果都不是0, 一直减 x_2 , 同时对 x_1 做加法(已经证明是部分可计算函数)

```

 $Z_1 \leftarrow X_1$ 
 $Z_2 \leftarrow X_2$ 
[A] IF  $Z_1 \neq 0$  GOTO B
    GOTO E
[B] IF  $Z_2 \neq 0$  GOTO C
    GOTO E
[C]  $Z_2 \leftarrow Z_2 - 1$ 
     $Z_1 \leftarrow Z_1 + Z_1$ 
    GOTO B

```

(4) 思路: 空函数就是处处无定义, 直接进入死循环即可.

```

[A]  $Z \leftarrow Z + 1$ 
    IF  $Z \neq 0$  GOTO A

```

□

题目. P_{13} 1.4 证明下述谓词是可计算的

- (1) $x \geq a$, a 是正整数
- (2) $x_1 \leq x_2$
- (3) $x_1 = x_2$

解答. 要证明一个谓词是可计算的, 实际上就是证明这个谓词(判断过程)可以用S语言表示

(1) 思路: 两边一直减1, 看谁先减到0, 又因为是大干等于, 所以可以先验证 Z_2

```

 $Z_1 \leftarrow X$ 
 $Z_2 \leftarrow a$ 
[A]IF  $Z_2 \neq 0$  GOTO B
    GOTO  $D_2$ 
[B]IF  $Z_1 \neq 0$  GOTO C
    GOTO  $D_1$ 
[C] $Z_1 \leftarrow Z_1 - 1$ 
     $Z_2 \leftarrow Z_2 - 1$ 
    GOTO A
[ $D_1$ ]GOTO E
[ $D_2$ ] $Y \leftarrow Y + 1$ 
    GOTO E

```

(2) 思路: 和第一问思路类似, 注意取等条件对判定顺序的影响

```

 $Z_1 \leftarrow X_1$ 
 $Z_2 \leftarrow X_2$ 
[A]IF  $Z_1 \neq 0$  GOTO B
    GOTO  $D_2$ 
[B]IF  $Z_2 \neq 0$  GOTO C
    GOTO  $D_1$ 
[C] $Z_1 \leftarrow Z_1 - 1$ 
     $Z_2 \leftarrow Z_2 - 1$ 
    GOTO A
[ $D_1$ ]GOTO E
[ $D_2$ ] $Y \leftarrow Y + 1$ 
    GOTO E

```

(3) 思路: 可以使用(1)和(2)的判定了

```

 $Z_1 \leftarrow X_1$ 
 $Z_2 \leftarrow X_2$ 
[A]IF  $Z_1 \leq Z_2$  GOTO B
    GOTO E
[B]IF  $Z_2 \leq Z_1$  GOTO C
    GOTO E
[C] $Y \leftarrow Y + 1$ 
    GOTO E

```

□

2 原始递归函数

题目. P_{16} 2.1.5用基本的原始递归函数来表示下面的函数, 从而它们也是原始递归函数

$$(1) E(x) = \begin{cases} 1, & x \text{ 偶数} \\ 0, & x \text{ 奇数} \end{cases}$$

$$(3) \max(x, y) = \begin{cases} x, & x \geq y \\ y, & x < y \end{cases}$$

解答. (1) $E(0) = 1, E(x+1) = \alpha(E(x))$.

(2) $\max(x, y) = x\alpha(y \dot{-} x) + y\alpha(x \dot{-} p(y))$

这是因为, 当 $x \geq y$ 时, $y \dot{-} x = 0$, 因此 $\alpha(y \dot{-} x) = 1$. 但为了防止 $x = y$ 且取非零值 ($x = y = 0$ 不影响) 的时候, 得到 $2x$, 因此要保证 $x = y$ 的时候, y 的系数不能是 1, 因此取 $x \dot{-} p(y)$ \square

题目. P_{22} 2.3.4利用极小化给出下述函数 $f(x)$:

$$(1) f(x) = \begin{cases} \sqrt{x}, & x \text{ 是完全平方数} \\ \uparrow, & \text{否则} \end{cases}$$

(2) $g(x)$ 是全函数, 若存在 t , 使得 $g(t) = x$, 则 $f(x)$ 等于使得 $g(t) = x$ 成立的最小的 t , 否则 $f(x) \uparrow$.

解答. (1) $f(x) = \min_t (t^2 = x)$

(2) $f(x) = \min_t (g(t) = x)$ \square

题目的注记. 应该是对的: 使用极小化定义的函数中, 加入我想要定义的函数不是全函数(即在某些情况下无定义), 那么一定是用无界极小化定义的. 因为有界极小化+原始递归, 定义出的都是全函数

题目. P_{36} 2.1 证明: 仅在有限个点处非零值, 在其余点取零的函数一定是原始递归函数.

解答. 思路: 因为只在有限个地方取非零值, 那么只需要对这有限个点做有限递归

设 $f(x)$ 在 x_1, \dots, x_k 处取非零值 c_1, \dots, c_k , 其余点取零.

定义:

$$\chi_{x_i}(x) = \begin{cases} 1, & x = x_i \\ 0, & x \neq x_i \end{cases} = (x = x_i)$$

是原始递归的谓词, 那么定义:

$$f(x) = \sum_{i=1}^k c_i \cdot \chi_{x_i}(x)$$

(有限求和, 常数乘法原始递归) 因此也是原始递归的. \square

题目的注记. 也可以写成:

$$f(x) = \sum_{i=1}^k c_i \alpha(x - x_i)$$

题目. P_{36} 2.3 设 $f(0) = 1, f(1) = 1, f(2) = 2^2, f(3) = 3^{3^3} = 3^{27}$ 等. 一般地, $f(n)$ 等于高度为 n 的一叠 n , 都是指数, 试证明 f 是原始递归的.

解答. 自己没有想出来,看了答案才想出来的,一直纠结于单变量的情况,却没有想到构建多变量的函数,之后给参数赋值来变回单变量

设 $h(n, m)$ 是 $m + 1$ 个 n 的指数堆叠, 那么 $h(n, 0) = n, h(n, t + 1) = n^{h(n, t)}$, 所以 $h(n, m)$ 原始递归, 因此 $h(n, n - 1) = f(n)$ 原始递归.

虽然最后一步更自然的说法应该是取 $n = m + 1$, 那么 $h(n, m) = h(m + 1, m) = h(n, n - 1) = f(n)$. 因为上面的 n 作为参数, 我是证明了 $h(n, m)$ 这个二元函数关于 m 是原始递归的. \square

题目. P_{36} 2.5 设 $\sigma(0) = 0$; 当 $x \neq 0$ 时, $\sigma(x)$ 是 x 的所有因子的和. 证明 $\sigma(x)$ 是原始递归函数.

解答. 写成原始递归的形式:

$$\sigma(x) = \sum_{i=1}^x \min_{i \leq t \leq n} (t|x) = \sum_{i=1}^x \min_{t \leq n} ((t|x) \wedge (i \leq t)), \quad x \neq 0$$

谓词 $(t|x)$ 和 $(i \leq t)$ 是原始递归的, 原始递归谓词的”且” \wedge 是原始递归的, 有界极小化 $\min_{t \leq n} ((t|x) \wedge (i \leq t))$ 是原始递归的, 因此 $\sigma(x)$ 也是原始递归的 \square

题目. P_{36} 2.7 设 $\phi(x)$ 是小于等于 x 且与 x 互素的正整数的个数, 证明Euler函数 $\phi(x)$ 是原始递归函数.

解答. 先考虑判断 x, y 是否互素的量词 $P(x, y)$ 是原始递归的.

$$P(x, y) = (\forall)_{t \leq x} ((t = 1) \vee \neg((t|x) \wedge (t|y))) = (\forall)_{t \leq x} ((t = 1) \vee \neg(t|x) \vee \neg(t|y))$$

因此:

$$\phi(x) = \sum_{t=1}^x P(x, t)$$

\square

题目的注记. 奇怪的是, 书上的答案是这么写的:

$$P(x, y) = (x > 0 \vee y > 0) \wedge (\forall)_{t \leq x} ((t = 1) \vee \neg(t|x) \vee \neg(t|y))$$

即 x, y 不可以同时为0, 这是必要的吗?

题目. P_{24} 2.4.3 验证:

$$(x)_i = \min_{t \leq x} \{\neg(p_i^{t+1} | x)\}$$

对于 $(0)_i$ 和 $(x)_0$ 成立, 其中 x 和 i 是任意的自然数.

解答. 实际含义: 第 i 个素数 p_i 作为 x 的因子的次数

$$(0)_i = \min_{t \leq 0} \{\neg(p_i^{t+1} | 0)\} = \neg(p_i | 0) = \neg 1 = 0$$

$$(x)_0 = \min_{t \leq x} \{\neg(0^{t+1} | x)\} = \min_{t \leq x} \{\neg(0 | x)\} = \min_{t \leq x} \{1\} = 0$$

解释: $(0|x) = (\exists)_{t \leq x} (t \cdot 0 = x) = 0$ \square

题目. P_{37} 2.11 设 $R(x, t)$ 是原始递归谓词, 定义有界极大化:

$$g(x, y) = \max_{t \leq y} R(x, t)$$

当存在 $t \leq x$ 使得 $R(x, t)$ 为真时, $g(x, y)$ 等于这样的 t 的最大值; 当不存在这样的 t 的时候, $g(x, y) = 0$. 证明 $g(x, y)$ 原始递归.

解答. 第一反应是, 已经知道了有界极小化原始递归, 可以尝试用有界极小化来表示有界极大化.

那么可以这么想: 使得 $R(x, t)$ 成立的最大的 t 就是: 使得 $k = t$ 到 y 的 $R(x, k)$ 只有 $R(x, t)$ 成立的最小的 t .

$$g(x, y) = \min_{t \leq y} \left\{ R(x, t) \wedge \left(\left((t < y) \wedge \alpha \left(\sum_{k=t+1}^y R(x, k) \right) \right) \vee (t = y) \right) \right\}$$

当不存在这样的 t 的时候, $R(x, t)$ 恒为 0, 因此 $g(x, y) = 0$, 满足题设

□

题目的注记. 书上给的两种解法也很有意思:

(1) 方法一: 和我的思路是一样的, 即首先 $R(x, t)$ 要成立, 且要么 $s \leq t$, 要么 $R(x, s)$ 不成立. 但是他比我聪明的地方在于对 \forall 的量词和 \vee 的使用

$$g(x, y) = \min_{t \leq y} \{ R(x, t) \wedge (\forall s)_{s \leq y} [(s \leq t) \vee \neg R(x, s)] \}$$

(2) 方法二: 倒着开始用有界极小化, 找到了再减回去得到正着数的下标

$$g(x, y) = \begin{cases} y - \min_{t \leq x} R(x, y - t), & (\exists z)_{z \leq y} R(x, z) \\ 0, & \text{否则} \end{cases}$$

题目. P_{37} 2.13

(1) Cantor 编码 $\pi(x, y)$ 的定义如下所示

(2) 若 $\pi(x, y) = z$, 则 $\sigma_1(z) = x, \sigma_2(z) = y$

(3) $\sigma(z) = \sigma_1(z) + \sigma_2(z)$

证明 $\pi(x, y), \sigma_1(z), \sigma_2(z), \sigma(z)$ 都是原始递归的.

解答. (1) $\pi(x, y)$ 是 x 行 y 列, 首先, 第 0 行, 第 y 列是 $\frac{y(y+1)}{2}$, 所以 (x, y) 元素是:

$$\frac{y(y+1)}{2} + (y+2) + (y+3) + \cdots + (x+y+1) = \frac{(x+y)(x+y+1)}{2} + x$$

因此是原始递归函数

(2) 答案是这么写的, 但是还有点疑惑

$$\sigma_1(z) = \min_{x \leq z} [(\exists y)_{y \leq z} \pi(x, y) = z]$$

$$\sigma_2(z) = \min_{y \leq z} [(\exists x)_{x \leq z} \pi(x, y) = z]$$

(3) $\sigma_1(z), \sigma_2(z)$ 原始递归, 所以加和原始递归

□

题目. P_{38} 2.21 证明下述字函数是原始递归的:

(1) 长度函数 $|w|$

$|w|$ 等于 w 的长度, 即 w 中的字符个数;

(2) 连接函数 $\text{CONCAT}^{(m)}(w_1, w_2, \cdots, w_m)$, 其中 m 是正整数,

$$\text{CONCAT}^{(m)}(w_1, w_2, \cdots, w_m) = w_1 w_2 \cdots w_m$$

(3) 字尾函数 $\text{RTEND}(w)$

当 $w = \varepsilon$ 时, $\text{RTEND}(w) = \varepsilon$; 当 $w \neq \varepsilon$ 时, $\text{RTEND}(w)$ 等于 w 最右端的字符;

(4) 字头函数 $\text{LTEND}(w)$

当 $w = \varepsilon$ 时, $\text{LTEND}(w) = \varepsilon$; 当 $w \neq \varepsilon$ 时, $\text{LTEND}(w)$ 等于 w 最左端的字符;

(5) 截尾函数 w^-

当 $w = \varepsilon$ 时, $w^- = \varepsilon$; 当 $w \neq \varepsilon$ 时, w^- 等于 w 删去最右端的字符后的子串;

(6) 截头函数 w^-

当 $w = \varepsilon$ 时, $-w = \varepsilon$; 当 $w \neq \varepsilon$ 时, $-w$ 等于 w 删去最左端的字符后的子串。

解答. 要证明字函数是原始递归函数, 需要证明这个字函数对应的数论函数是原始递归的; 或者是由其他的原始递归的字函数构造得到的

指定字母表 $A = \{a_1, \dots, a_n\}$

(1) 从直观上说, 给定一个字符串 w , 获取 w 的长度是不需要知道 w 的字符所属的字母表的, 但是, 这里为了分析这个过程的**可计算性**, 需要指定一个字母表.

那么在 n 进制下, 长度为 $k+1$ 的最小字符串是 $a_1 a_1 \dots a_1 = \sum_{t=0}^k n^t$, 那么长度是 (控制 $k \leq x$, 可以是有界极小化, 这里的 x 在比较的时候, 就是数), 根据有界极小化和原始递归谓词定义的函数是原始递归的:

$$\min_{k \leq x} \left\{ \sum_{t=0}^k n^t > x \right\}$$

(2) 连接函数, w_1 向前挪动一格, 值乘上 n , 而向前挪动的格子数可以由第一问的长度函数得到, 不妨设长度函数是 $l(w)$, 那么:

$$\begin{aligned} & \text{CONCAT}^{(m)}(w_1, \dots, w_m) \\ &= w_m + w_{m-1} \cdot n^{l(w_m)} + w_{m-1} \cdot n^{l(w_m)+l(w_{m-1})} + \dots + w_1 \cdot n^{l(w_m)+\dots+l(w_2)} \\ &= \sum_{i=0}^{m-1} w_{m-i} \cdot n^{l(w_m)+\dots+l(w_{m-i+1})} \end{aligned}$$

(3) 首先, 判断 $w = \epsilon$ 当然是原始递归的谓词, 因为这就是 $w = 0$ 的判断, 其次, 获取 w 最右端字符就是 $R^+(w, n)$, 因此是原始递归的

$$\text{RTEND}(w) = \begin{cases} \epsilon, & \text{if } w = \epsilon \\ R^+(w, n), & \text{else} \end{cases}$$

(4) 长度为 $l(w)$ 的字符串的首位是 $n^{l(w)-1}$ 的阶, 因此用 $Q^+(w, n^{l(w)-1})$ 获取最左端的字符 (即 w 用 $n^{l(w)-1}$ 来除之后得到的整数部分), 因此是原始递归的

$$\text{LTEND}(w) = \begin{cases} \epsilon, & \text{if } w = \epsilon \\ Q^+(w, n^{l(w)-1}), & \text{else} \end{cases}$$

当然, 也可以用书上讲过的, 可以直接拿来用的函数, 更加安全, $h(m, n, x)$ 表示字符串 x 在 n 进制表示下第 m 位的字符, 因此可以写成:

$$\text{LTEND}(w) = \begin{cases} \epsilon, & \text{if } w = \epsilon \\ h(l(w) - 1, n, x), & \text{else} \end{cases}$$

□

题目. P_{38} 2.22 设两个字母表 $A = \{s_1, s_2, \dots, s_n\}$ 和 $\tilde{A} = \{s_1, s_2, \dots, s_m\}$, 其中 $n < m$ 。任给一个数 x , 设 $w \in A^*$ 是 x 为底表示 9, 以 5 为底表示 36, 故 $\text{UPCHANGE}_{2,5}(9) = 36$ 。

反之, 任给一个数 x , 设 $w \in \tilde{A}^*$ 是 x 的 m 进制表示, 删去 w 中不属于 A 的符号后得到 $w' \in A^*$, 把以 n 为底 w' 所表示的数记作 $\text{DOWNCHANGE}_{n,m}(x)$ 。例如, $\text{DOWNCHANGE}_{2,5}(36) = 9$ 。又如, $s_1 s_4 s_2 s_3$ 以 5 为底表示 238, 删去 s_4 和 s_3 得到 $s_1 s_2, s_1 s_2$, 以 2 为底表示 4, 故 $\text{DOWNCHANGE}_{2,5}(238) = 4$ 。

试证明: $\text{DOWNCHANGE}_{n,m}(x)$ 和 $\text{UPCHANGE}_{n,m}(x)$ 是原始递归的。

3 通用程序

题目. 计算 $STP(x_1, \dots, x_n, y, 0) = ?$

解答.

$$STP(x_1, \dots, x_n, y, 0) = \begin{cases} 1, & \text{if } y = 0 \\ 0, & \text{否则} \end{cases}$$

□

题目. 3.1 证明 $HALT(0, x)$ 是不可计算的

解答. 反证法, 假设 $HALT(0, x)$ 可计算, 构造:

[A] IF $HALT(0, x)$ GOTO A

这是可计算的, 有程序 P 可以计算, 程序 P 的编号是 k .

那么 $HALT(0, k) \iff P(k) = 0 \iff \neg HALT(0, k)$, 矛盾.

□

题目. 3.3 证明: 不存在可计算函数 $f(x)$, 使得当 $\Phi(x, x) \downarrow$ 时 $f(x) = \Phi(x, x) + 1$.

解答. 反证法, 如果是可计算的, 那么函数 $f(x)$ 有编号 k , 那么 $f(x) = \Phi(x, k)$, 因此有 $f(k) = \Phi(k, k) = \Phi(k, k) + 1$, 矛盾.

□

题目. 设 f 是一个全函数, $B = \{f(n) | n \in N\}$, 证明:

(1) 若 f 是可计算的, 那么 B 是 *r.e.*

(2) 若 f 是可计算的, 严格增加的 ($f(n) < f(n+1), \forall n \in N$), 则 B 是递归的.

解答. (1) 构造一个可计算函数, 使得输入的 x 如果在 B 里面, 就停机(输出什么都行), 否则不停机(无定义)

$$h(x) = \begin{cases} 0, & \text{if } \exists n (f(n) = x) \\ 1, & \text{else} \end{cases}$$

可计算函数:

[A] IF $f(N) = X$ GOTO E

$N \leftarrow N + 1$

GOTO A

(2) 构造一个可计算函数, 使得输入的 x 如果在 B 里面, 就输出 1, 如果不在里面, 就输出 0 又因为函数是严格单调的, 因此我只需要遍历 N , 直到 $f(N)$ 大于 X 或者 $f(N) = X$ 即可.

[A] IF $f(N) = X$ GOTO C

[B] IF $f(N) > X$ GOTO E

$N \leftarrow N + 1$

GOTO A

[C] $Y \leftarrow Y + 1$

GOTO E

□

题目的注记.

- 存在不是可计算的全函数, 例子(这是一个不可计算的谓词): $\text{HALT}(x, x)$
- 存在不是部分可计算的部分函数, 例子: $f(x) = \begin{cases} 1, & \text{if } \neg \text{HALT}(x, x) \\ 0, & \text{if } \text{HALT}(x, x) \end{cases}$

题目. 3.6 设 A, B 是 N 的非空子集, 定义:

$$A \odot B = \{2x | x \in A\} \cup \{2x + 1 | x \in B\}$$

$$A \otimes B = \{ \langle x, y \rangle | x \in A, y \in B \}$$

证明:

- (1) $A \odot B$ 是递归的当且仅当 A 和 B 是递归的
- (2) $A \otimes B$ 是递归的当且仅当 A 和 B 是递归的

题目的注记. 实际上, 上述两个集合的构造方式都是”双射”的编码, 因此只要知道 x 是否属于 $A \odot B$, 那么就一定知道 x 是否属于 A 或者 B , 反之亦然(显然)

解答. (1) 分析谓词: $(x \in A \odot B), (x \in A), (x \in B)$:

$$(x \in A \odot B) \iff ((2|x) \wedge (\lfloor x/2 \rfloor \in A)) \vee (\neg(2|x) \wedge (\lfloor (x-1)/2 \rfloor \in B))$$

(2) 对于配对函数 $z = \langle x, y \rangle$, 有原始递归函数 $l(z), r(z)$.

分析谓词: $(x \in A \otimes B), (x \in A), (x \in B)$:

$$(z \in A \otimes B) \iff ((x = l(z)) \wedge (x \in A)) \vee ((x = r(z)) \wedge (x \in B))$$

□

题目. 3.7 证明集合 $B = \{x \in N | a \in \text{ran} \Phi_x\}$ 是递归可枚举的, a 是常数.

解答. 这里的 $\text{ran} \Phi_x$ 就是值域

注意, 这里的函数 Φ_x 是部分可计算的, 而不是像 3.5 的全函数

我需要构造一个可计算函数来判断 x 是否属于 B 的, 即, 如果属于, 我知道, 但是不知道不属于
需要再添加一个程序来执行对 $\Phi(n_0, x)$ 是否有定义的判定.

```
[A] IF  $\neg \text{STP}(N, X, T)$  GOTO B
    IF  $\Phi(N, X) = a$  GOTO E    需要对输入N停机之后才能有输出来判断是不是a
[B]  $N \leftarrow N + 1$ 
    IF  $N \leq T$  GOTO A
     $T \leftarrow T + 1$ 
     $N \leftarrow$ 
    GOTO A
```

□