

《计算模型导引》习题

Yang

2022

目录

1	递归函数	2
1.1	2
1.2	2
1.3	2
1.4	3
1.5	3
1.6	4
1.7	4
1.8	6
1.9	6
1.10	7
1.11	7
1.12	8
1.13	8
1.14	8
1.15	9
1.16	9
1.17	9

1.18	10
1.19	10
1.20	11
1.21	11
1.22	11
1.23	11
3 λ-演算	12
3.1	12
3.2	12
3.3	12
3.4	12
3.5	12
3.6	13
3.7	13
3.8	13
3.9	13
3.10	13
3.11	14
3.12	15
3.13	15
3.14	15
3.16	16
3.17	16
3.18	16
3.19	17
3.20	17
3.21	17

3.22	18
3.23	18
3.24	19
3.25	20
5 图灵机	20
5.1	20
5.2	20
5.3	22
5.4	23
5.5	23
5.6	24
5.7	24
5.8	24
5.9	25
5.10	25
5.11	25
5.12	26
5.13	26
5.14	27
5.15	27
5.16	28
5.17	28
5.18	29
5.19	29
5.20	29

1 递归函数

1.1 证明：对于固定的 k ，一元数论函数 $x + k \in \mathcal{BF}$

Proof. 借助恒等函数 P_1^1 与后继函数 S ，对任意 k ，可组合构造 $f_k(x) = x + k$.

$$f(x) = \begin{cases} P_1^1(x) & k = 0 \\ \underbrace{S \circ \dots \circ S}_{k-1} \circ P_1^1(x) & k > 0 \end{cases}$$

由于 $f_k(x) = x + k$ 可由基本函数 P_1^1 与 S 构造，所以 $x + k \in \mathcal{BF}$. \square

1.2 证明：对于任意 $k \in \mathbb{N}^+$ ， $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ，若 $f \in \mathcal{BF}$ ，则存在 h 使得 $f(\vec{x}) < \|\vec{x}\| + h$

Proof. 对 f 的构造长度 l 进行归纳，当 $l = 0$ 时，我们有 $f \in \mathcal{IF}$ ，此时取 $h = 2$ ，不等式显然恒成立.

对于 $l = n + 1$ 的情况，我们有归纳假设：存在 h_n ，对于构造长度小于等于 n 的函数，使得 $f(\vec{x}) < \|\vec{x}\| + h_n$ 成立. 假设构造序列为 f_1, \dots, f_n, f . 若 $f \in \mathcal{IF}$ ，显然 $f(\vec{x}) < \|\vec{x}\| + h_n + 1$. 若 $f = \text{Comp}_k^m[f_{i_0}, \dots, f_{i_k}]$ ，根据归纳假设有 $f(\vec{x}) < \max\{f_{i_j}(\vec{x})\} + h_n < \|\vec{x}\| + 2h_n$ ，此时我们找到了 $h_{n+1} = 2h_n$. \square

1.3 证明： $f(x, y) = x + y \notin \mathcal{BF}$

Proof. 使用反证法，假设 $f(x, y) = x + y \in \mathcal{BF}$ ，构造 $f'(x) = f(x, x) = 2x$ ，易证 $f' \in \mathcal{BF}$. 根据1.2可知， $\exists h \forall x, f(x) = 2x < x + h$ ，该式显然不成立，反证成立. \square

1.4 证明: $f(x, y) = x \div y \notin \mathcal{BF}$

Proof. 只需说明 $\text{pred}(x) = x \div 1 \notin \mathcal{BF}$ 即可, 因为 pred 可由 $x \div y$ 与基本函数构造出.

假设 $\text{pred} \in \mathcal{BF}$, 在其最短构造序列上做分解证明. 首先, $\text{pred} \notin \{S, Z, P\}$, 于是可设该构造序列为 f_0, \dots, f_n .

首先说明该序列中不存在对 P 的使用: 若存在, 设为 $f_i(\vec{x}) = P_i(g_1(\vec{x}), \dots, g_k(\vec{x}))$, 我们可以直接找到一个更简单的构造 $f_i(\vec{x}) = g_i(\vec{x})$ 使得序列更短.

此时思考如何使用 $\{S, Z\}$ 构造 pred . 由于两者都是一元函数, 其组合可写为 $F_0 \circ \dots \circ F_m (F_i \in \{S, Z\})$, 易知 $Z \circ \dots = Z$, 根据最短序列的假设, $\text{pred} = S^c \circ Z$ 或 $\text{pred} = S^c$, 其中 c 为某个常数.

1. $S^c \circ Z = c$, 为常数, 不满足需求.
2. $S^c = +_c$, 只会增加, 不会递减, 不满足需求.

综上, $x \div y \notin \mathcal{BF}$

□

1.5 说明 $\text{pg}(x, y) = 2^x(2y + 1) \div 1$ 为配对函数

Proof. 令 $K(z) = \text{ep}_0(z + 1), L(z) = \frac{1}{2} \cdot (\frac{z+1}{2^{\text{ep}_0(z+1)}} \div 1)$. 我们注意到, $2^x(2y + 1) > 0$ 恒成立, 所以计算 pg 时 \div 可理解为 $-$; 2^x 为偶数, $2y + 1$ 为奇数.

$K(\text{pg}(x, y))$ 取 $2^x(2y + 1)$ 的 2 的指数, 即 x .

$L(\text{pg}(x, y)) = \frac{2y}{2} = y$.

$\text{pg}(K(z), L(z)) = 2^{\text{ep}_0(z+1)} \cdot (2 \cdot \frac{1}{2} \cdot (\frac{z+1}{2^{\text{ep}_0(z+1)}} \div 1) + 1) - 1 = z + 1 - 1 = z$

注: $2_i^{\text{ep}}(n)$ 必然被 n 整除. 为了使配对函数组满足双射, 需要避免计算出现不确定性, 如取整.

□

1.6 设 $f: \mathbb{N} \rightarrow \mathbb{N}$, 证明: f 可以作为配对函数的左函数当且仅当

$$\forall i \in \mathbb{N}, |\{z \in \mathbb{N} : f(z) = i\}| = \aleph_0$$

Proof. 设 $Z_{x=i} = \{z \in \mathbb{N} : f(z) = i\}$. 若存在配对函数, 设为 $pg(x, y)$, 右函数设为 $g(z)$.

\Rightarrow 根据可数选择公理, 只需证明 $\forall i, Z_{x=i}$ 是无限的. 假设对于某个 i , $Z_{x=i}$ 有限, 取 $Y_{x=i} = \{j | g(z) = j \wedge z \in Z_{x=i}\}$, 可知 $Y_{x=i}$ 也是有限的. 取任意 $y \in \mathbb{N} - Y_{x=i}$, 根据配对函数定义, $f(pg(i, y)) = i$, 这意味着 $pg(i, y) = z \in Z_{x=i}$, 这意味着 $y \in Y_{x=i}$, 矛盾.

\Leftarrow 此时, 对于任意的 i , $Z_{x=i}$ 可以与 \mathbb{N} 建立一个双射 $h_{x=i}: \mathbb{N} \rightarrow Z_{x=i}$, 其逆为 $h_{x=i}^{-1}$. 此时定义 g 如下:

$$g(z) = \begin{cases} h_{x=i}^{-1}(z) & z \in Z_{x=i}, \\ 0 & \text{otherwise.} \end{cases}$$

令 $pg(x, y) = h_{x=x}(y)$, 显然, $\forall x, y. f(pg(x, y)) = x \wedge g(pg(x, y)) = y$, 满足配对函数定义.

□

1.7 证明: 所有的初等函数, 都可以由本原函数与复合和算子 $\prod_{i=n}^m [\cdot]$ 生成, 其中,

$$\prod_{i=n}^m [f(i)] = \begin{cases} f(n) \cdot f(n+1) \cdots f(m) & m \geq n, \\ 1 & m < n \end{cases}$$

Proof. 乘法不能直接退化为加法. 我们尝试放大 $\prod_{i=n}^m [\cdot]$ 的分支计算能力, 构造以下函数. 其中, 0 可理解为布尔运算的 true, 1 理解为 false, N 可以理解为 \neg .

$$\text{pow}(x, k) = \prod_{i=1}^k x$$

$$N(x) = \prod_{i=1}^x Z(i)$$

$$\text{leq}(x, y) = \prod_{i=x}^y Z(i)$$

$$\text{geq}(x, y) = \prod_{i=y}^x Z(i)$$

$$\text{and}(x, y) = \text{pow}(x, N(y)) \quad (x, y \in \{0, 1\})$$

$$\text{eq}(x, y) = \text{and}(\text{leq}(x, y), \text{geq}(x, y))$$

利用 eq 可以构造求解某范围内函数所有零点的积的函数 h （若没有零点，返回 1）。若我们准确知道函数 f 具有唯一零点，那么 h 便可以准确求得该零点。

$$h(n)[f(x)] = \prod_{i=0}^n i^{N(\text{eq}(f(i), 0))}$$

令 $f(i) = 2^i - n$ ，取 $\log(x) = \prod_{i=0}^n i^{N(\text{eq}(2^i - n, 0))}$ ，由于 $\log(x)$ 若存在解，该解一定在 $[0, x]$ 间，所以该定义可以准确求解 2^k 得对数。现在可利用该函数将乘法退化为加法。

$$\sum_{i=n}^m f(i, \vec{x}) = \log\left(\prod_{i=n}^m 2^{f(i, \vec{x})}\right)$$

于是可以构造其他基本初等函数（注意到 $\sum_{i=m}^n [\cdot] = 0$ if $m > n$ ）：

$$x \times y = \sum_{i=1}^x y$$

$$x + y = \log(2^x \times 2^y)$$

$$x \dot{-} y = \left(\sum_{i=x+1}^y 1 + \sum_{i=y+1}^x 1 \right)$$

□

1.8 设

$$M(x) = \begin{cases} M(M(x+11)) & x \leq 100, \\ x-10 & x > 100. \end{cases}$$

试证明：

$$M(x) = \begin{cases} 91 & x \leq 100, \\ x-10 & x > 100. \end{cases}$$

Proof. 分类情况讨论，首先可知 $M(101) = 91$.

$$90 \leq x \leq 100 \quad M(x) = M(x+1) = \cdots = M(101) = 91.$$

$$0 \leq x \leq 90 \quad \exists k, n. 91 \leq x + 11k \leq 101 \wedge M(x) = M^n \circ M(x + 11k) = M^n(91) = 91 \quad (\text{注：易证 } \forall n. M^n(91) = 91.) \quad \square$$

1.9 证明： $\min x \leq n.[f(x, \vec{y})] = n - \max x \leq n.[f(n-x, \vec{y})]$.

Proof. 若 $f(x, \vec{y})$ 关于 x 在范围内不存在零点，等式 $n = n - 0$ 显然成立.

若 $f(x, \vec{y})$ 存在零点，我们设最小零点为 m ，最大零点为 M . 可知对于任意 $a < m$, $f(a, \vec{y}) \neq 0$. 我们尝试说明 $x = n - m$ 是 $f(n-x, \vec{y})$ 的最大零点.

1. 由于 $f(n - (n - m), \vec{y}) = f(m, y) = 0$, $x = n - m$ 是零点.
2. 假设 $x = (n - m)$ 不是最大零点, 那么 $\exists k > 0. x' = n - m + k \wedge f(n - x', \vec{y}) = 0$. 化简得 $f(n - x', \vec{y}) = f(n - n + m - k, \vec{y}) = f(m - k, \vec{y}) = 0$.
 - (a) 若 $m = 0$, 则与 $n - m$ 为不是最大零点矛盾
 - (b) 若 $m > 0$, 则 $\exists m' = m - k, m' < m \wedge f(m', \vec{y}) = 0$, 与 m 为最小零点矛盾.

综上, 得证. 对称形式用相似方法亦可证.

□

1.10 证明: \mathcal{EF} 对有界 max-算子封闭

Proof.

$$\sum_{i=0}^n [N(\prod_{j=0}^i [N^2(f(n-j, \vec{y}))])] = \begin{cases} \max x \leq n.[f(x, \vec{y})] + 1 & \exists x. f(x, \vec{y}) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

所以,

$$\max x \leq n.[f(x, \vec{y})] = \sum_{i=0}^n [N(\prod_{j=0}^i [N^2(f(n-j, \vec{y}))])] \div 1$$

于是对于任意 $f \in \mathcal{EF}$, $\max x \leq n.[f(x, \vec{y})] \in \mathcal{EF}$, 所以 \mathcal{EF} 对该算子封闭.

□

1.11 Euler 函数 $\varphi: \mathbb{N} \rightarrow \mathbb{N}$ 定义为

$$\varphi(n) = |\{x \mid 0 < x \leq n \wedge \gcd(x, n) = 1\}|,$$

证明 $\varphi \in \mathcal{EF}$.

Proof.

$$\varphi(n) = \sum_{m=1}^n \prod_{j=0}^n (N(\text{ep}(j, n) = 0 \vee \text{ep}(j, m) = 0))$$

□

1.12 令 $h(x)$ 计算 x 的最大素因子下标, 证明 $h \in \mathcal{EF}$.

Proof.

$$h(x) = \max n \leq x. [\text{ep}(n, x) \neq 0].$$

□

1.13 对于斐波那契函数 f , 证明 (1) $f \in \mathcal{PRF}$ (2) $f \in \mathcal{EF}$.

Proof. 寻找原始递归的构造时, 需要借助配对函数, 使返回值可以包含多个值, 用以传递前两层的结果。

我们令 $\{\text{pg}, K, L\}$ 为 \mathcal{PRF} 的一个配对函数, 构造 F :

$$F(0) = \text{pg}(1, 0)$$

$$F(x+1) = \text{pg}(K(F(x)) + L(F(x)), K(F(x)))$$

此时, $f(x) = K(F(x)), f(x \div 1) = L(F(x))$. 因而 $f \in \mathcal{PRF}$.

了解到斐波那契递归对应的原始问题: $f(x)$ 计算了长度为 $x-1$ 的不包含连续 1 的二进制串数量. (两个子问题: 串首位为 0 或首位为 10).

该问题可以用初等函数以遍历形式表达, 以说明 $f \in \mathcal{EF}$:

$$f(n) = \sum_{i=0}^{2^{n-1}-1} N \left[\sum_{i=0}^{n-2} \text{neq} \left(\frac{\text{rs}(i, 2^j)}{2^{j-1}} \right) \text{neq} \left(\frac{\text{rs}(i, 2^{j+1})}{2^j} \right) \right]$$

该函数对范围内满足检查的自然数进行计数: 检查每个自然数的每相邻两位不存在同时等于 1 的情况. □

1.14 证明 $Q(x, y, z, v) \equiv p(\langle x, y, z \rangle) \mid v$ 是初等函数.

Proof. 由于 $p \in \mathcal{EF}$, 且 $\langle x, y, z \rangle = 2^x \cdot 3^y \cdot 5^z \in \mathcal{EF}$,

$x \mid y = \text{eq}(\text{rs}(y, x), 0) \in \mathcal{EF}$. 所以 $Q \in \mathcal{EF}$. □

1.15 证明 $f \in \mathcal{PRF}$, f 定义为

$$f(0) = 1$$

$$f(1) = 4$$

$$f(2) = 6$$

$$f(x+3) = f(x) + (f(x+1))^2 + (f(x+2))^3$$

Proof. 与1.13类似, 如下定义 F :

$$F(0) = \langle 1, 4, 6 \rangle$$

$$F(x+1) = \langle \text{ep}_1(F(x)), \text{ep}_2(F(x)), \text{ep}_0(F(x)) + \text{ep}_1^2(F(x)) + \text{ep}_2^3(F(x)) \rangle$$

$$\text{于是 } f(x) = \text{ep}_0(F(x)).$$

□

1.16 设 $f(n) = n^{\cdot \cdot \cdot n}$, 证明 $f \in \mathcal{PRF} - \mathcal{EF}$.

Proof. 首先证 $f \in \mathcal{PRF}$. 构造递归函数 g , $g(n, 0) = 0, g(n, x+1) = n^{g(n, x)}$, 显然 $f(n) = g(n, n)$, 由于 $g \in \mathcal{PRF}$, 所以 $f \in \mathcal{PRF}$.

然后反证 $f \notin \mathcal{EF}$. 若 $f \in \mathcal{EF}$, 我们能找到 k , 使得对于任意 n , 控制函数 $G(k, n) > f(n)$ 恒成立. 但显然, $f(k+2) = (k+2)^{\cdot \cdot \cdot k+2} > G(k, k+2) = 2^{\cdot \cdot \cdot k+2}$ (幂级的长度和每一个幂级的数字, 前者都更大). 所以 $f \notin \mathcal{EF}$.

□

1.17 设 $g \in \mathcal{PRF}$, 证明 $f \in \mathcal{PRF}$

$$f(x, 0) = g(x)$$

$$f(x, y+1) = f(f(\dots f(f(x, y), y-1), \dots), 0)$$

Proof. 易见, $f(x, y) = g^{2^{(y-1)}}(x)$. 此时可以构造原始递归式计算 $g^n(x)$.

$$G(x, 0) = g(0)$$

$$G(x, y + 1) = g(G(x, y))$$

显然 $f(x, y) \in \mathcal{PRF}$.

□

1.18 若 $f, g \in \mathbb{N} \rightarrow \mathbb{N}$ 只在有限作用域的函数值不同,

证明 $f \in \mathcal{GRF} \iff g \in \mathcal{GRF}$.

Proof. 设这个作用域为 $S = \{s_0, s_1, \dots, s_k\}$, 根据题意, 有 $\forall x \in \mathbb{N} - S, f(x) = g(x)$.

此时可基于 f 在 \mathcal{GRF} 构造 $G = g$, 它在 $x \in S$ 时取 $G(x) = g(x)$, 在 $x \in \mathbb{N} - S$ 时取 $G(x) = f(x)$.

$$G(x) = \sum_{i=0}^k g(s_i) \cdot N(\text{eq}(s_i, x)) + N\left(\sum_{i=0}^k N(\text{eq}(s_i, x))\right) f(x).$$

对于前半表达式, 由于 S 有限, 它属于 \mathcal{GRF} . 后者保持 $f \in \mathcal{GRF}$. 所以 $F \in \mathcal{GRF}$. 对称证明类似.

□

1.19 证明 $\left\lfloor \left(\frac{\sqrt{5}+1}{2}\right)n \right\rfloor \in \mathcal{EF}$.

Proof.

$$f(n) = \max_z z \leq n. [(2z - n)^2 - 5n^2]$$

显然 $f \in \mathcal{EF}$.

□

1.20 证明 $\text{Ack}(4, n) \in \mathcal{PRF} - \mathcal{RF}$.

Proof. $f(n) = \text{Ack}(4, n)$, 我们可以为 f 写递归式:

$$f(0) = \text{Ack}(4, 0) = 13$$

$$f(n+1) = \text{Ack}(4, n+1) = \text{Ack}(3, f(n)) = 2^{f(n)+3} \dot{-} 3$$

所以 $f \in \mathcal{PRF}$.

假设 $f \in \mathcal{EF}$, 则存在 k 使得 $f(n) < G(k, n)$. 但 $f(n) = \underbrace{2^{\cdot^{\cdot^{\cdot^2}}}}_{n+3} - 3$, 增长率显然 f 一定比控制函数更快, 进而 $f \notin \mathcal{EF}$. □

1.21 设 $f: \mathbb{N} \rightarrow \mathbb{N}$ **是双射函数, 证明** $f \in \mathcal{GRF} \Leftrightarrow f^{-1} \in \mathcal{GRF}$.

Proof. 双射意味着 $\forall x_1, x_2, y, f(x_1) = y \wedge f(x_2) = y \Rightarrow x_1 = x_2$. 也即, $f^{-1}(y) = \mu x. [f(x) \dot{-} y]$.

显然命题成立. □

1.22 设 $p(x)$ **为整系数多项式, 令** $f(a)$ **定义为** $p(x) - a$ **对于** x **的最小非负整数根, 证明** $f \in \mathcal{RF}$.

Proof.

$$f(a) = \mu x. [p(x) \dot{-} a]$$

显然 $f \in \mathcal{RF}$. □

1.23 设

$$f(x, y) = \begin{cases} x/y & y \neq 0 \wedge y \mid x, \\ \perp & \text{otherwise.} \end{cases}$$

证明 $f \in \mathcal{RF}$.

Proof. $f(x, y) = \mu z. [(zy \dot{-} x)(N(y))]$. □

3 λ -演算

3.1 证明括号引理：对于任何 $M \in \Lambda$ ， M 的左右括号个数相同.

Proof. 采用结构归纳：

1. 对于 $x \in \nabla$ ，显然左右括号数相同.
2. 对于 $(\Lambda_1 \Lambda_2)$ ，显然新增左右括号数相同，根据归纳假设， Λ_1 与 Λ_2 左右括号数相同，所以该情况满足.
3. 对于 $(\lambda \nabla \Lambda)$ ，道理相同.

□

3.2 试求 $SSSS$ 的 β -nf.

Proof. 草纸上演算得： $\lambda ab.ab(\lambda c.bc(abc))$

□

3.3 证明： $(\lambda x.xxx)(\lambda x.xxx)$ 没有 β -nf.

Proof. 对于 $n > 1$ ，易证 $(\lambda x.xxx)^n \rightarrow_\beta (\lambda x.xxx)^{n+1}$. $(\lambda x.xxx)^{n+1}$ 永远含有一个可规约子项，最左侧的 $(\lambda x.xxx)^2$. 所以 $(\lambda x.xxx)^2$ 没有 β -nf. □

3.4 设 $F \in \Lambda$ 呈形 $\lambda x.M$ ，证明：(1) $\lambda z.Fz =_\beta F$ ；(2) $\lambda z.yz \neq_\beta y$.

Proof. 1. $\lambda z.(\lambda x.M)z =_\beta \lambda z.M[x := z] =_\alpha \lambda x.M$

2. 显然 $\lambda z.yz \neq y$ ，根据合流性， $\lambda z.yz \neq_\beta y$

□

3.5 证明二元不动点定理：对于任意 $F, G \in \Lambda$ ，存在 $X, Y \in \Lambda$ ，满足 $FX Y = X, GXY = Y$.

Proof. 设解向量 $A = [X, Y]$,

等式组等价于等式 $(\lambda z.[F(z)_1^2(z)_2^2, G(z)_1^2(z)_2^2])A = A$. 等式组的解等价于求该等式的解. 由一元不动点定理可知该等式存在不动点 A ，所以等式组也存在解 X, Y . □

3.6 证明：对任何 $M, N \in \Lambda^\circ$ ，方程 $xN = Mx$ 对于 x 有解。

Proof. 令 x 呈形 $\lambda a.T$. 原式化为 $T = M(\lambda a.T) = (\lambda t.M(\lambda a.t))T$. 根据一元不动点定理，存在不动点 $T = \Theta(\lambda t.M(\lambda a.t))$. 进而 x 也有解 $\lambda a.\Theta(\lambda t.M(\lambda a.t))$. \square

3.7 证明：对任何 P, Q ，若 $P \rightarrow_\beta Q$ ，则存在 $n \geq 0$ 以及 $P_0, \dots, P_n \in \Lambda$ ，满足 (1) $P \equiv P_0$; (2) $Q \equiv P_n$; (3) 对于任何 $i < n$ ， $P_i \rightarrow_\beta P_{i+1}$.

Proof. 对 \rightarrow_β 做结构归纳：

1. 若 $P \equiv Q$ ，显然具有单步规约序列（序列长度为 1， $n = 0$ ）.
2. 若 $P \rightarrow_\beta R \wedge R \rightarrow_\beta Q$ ，根据归纳假设， $P \rightarrow_\beta R$ 具有 $n = k_1$ 的单步规约序列， $R \rightarrow_\beta Q$ 具有 $n = k_2$ 的单步规约序列. 将两个序列合并，得到 $n = k_1 + k_2 - 1$ 的单步规约序列.

\square

3.8 证明：对任何 P, Q ，若 $P \rightarrow_\beta Q$ ，则 $\lambda z.P \rightarrow_\beta \lambda z.Q$.

Proof. 根据合拍性，该命题是显然的. \square

3.9 证明：对任何 $P, Q \in \Lambda$ ，若 $P =_\beta Q$ ，则存在 $n \in \mathbb{N}$ 以及 $P_0, \dots, P_n \in \Lambda$ ，满足 (1) $P \equiv P_0$; (2) $Q \equiv P_n$; (3) 对任何 $i < n$ ， $P_i \rightarrow_\beta P_{i+1}$ 或 $P_{i+1} \rightarrow_\beta P_i$.

Proof. 根据定理 3.20，有 $T \in \Lambda$ ， $P \rightarrow_\beta T \wedge Q \rightarrow_\beta T$. 根据 3.7，可构造序列 $P \rightarrow_\beta \dots \rightarrow_\beta T \leftarrow_\beta \dots \leftarrow_\beta Q$. \square

3.10 证明定理 3.12：对于任意 $M, N \in \Lambda$ ，

$$M =_\beta N \Leftrightarrow \lambda\beta \vdash M = N$$

Proof.

\Rightarrow 已知 $M =_{\beta} N$, 证 $\lambda\beta \vdash M = N$.

1. 对于 $M \rightarrow_{\beta} N$ 条件:

(a) $M \rightarrow_{\beta} N$ 对应于规则 β .

(b) 自反性对应于公理 ρ .

(c) 传递性对应于规则 τ .

2. 对称性对应于规则 σ .

3. 合拍性对应于规则 μ, ν, ξ .

\Leftarrow 已知 $\lambda\beta \vdash M = N$, 证 $M =_{\beta} N$. 规则可以显然地对应到关系中.

1. 公理 α 对应于自反闭包.

2. 公理 β 对应于关系 *beta*.

3. 规则 σ 对应于对称闭包.

4. 规则 τ 对应于传递闭包.

5. 规则 μ, ν, ξ 分别对应于合拍关系的一个条件.

□

3.11 证明定理 3.13: 对于任意 $M, N \in \Lambda$,

$$M =_{\beta, \eta} N \Leftrightarrow \lambda\beta\eta \vdash M = N$$

Proof. 只需要在3.10基础上说明公理 η 与 η 关系的对应即可. 而这个对应是显然的.

□

3.12 证明若 $M =_\beta N$ ，则存在 T 使 $M \rightarrow_\beta T$ 且 $N \rightarrow_\beta T$.

Proof. 已知

$$(M, N) \in \bigcup_{k=0} (\rightarrow_\beta \cup \leftarrow_\beta)^k.$$

我们对 k 做归纳, $k = 0$ 时 ($M \equiv N$) 命题显然成立.

$k = n + 1$ 时, 我们有 P 满足 $(P, N) \in (\rightarrow_\beta \cup \leftarrow_\beta)^n$, $M \rightarrow_\beta P$ 或 $M \leftarrow_\beta P$. 根据归纳假设, 存在 T_0 使得 $P \rightarrow_\beta T_0 \wedge N \rightarrow_\beta T_0$.

1. $M \rightarrow_\beta P$. 根据传递性, T_0 也可作为 M 与 N 的 β -规约汇点.
2. $M \leftarrow_\beta P$. 根据合流性, P 作为源点, M, T_0 作为分支点, 可以找到 T_1 满足 $M \rightarrow_\beta T_1 \wedge T_0 \rightarrow_\beta T_1$. 根据传递性, T_1 可以作为 M 与 N 的 β -规约汇点.

□

3.13 证明若在系统 $\lambda\beta$ 中加入下述公理 (A) $\lambda xy.x = \lambda xy.y$, 则对任何 $M, N \in \Lambda$, $\lambda\beta + (A) \vdash M = N$.

Proof. 根据合拍规则,

$$\begin{aligned} & \lambda\beta + (A) \vdash \lambda xy.x = \lambda xy.y \\ \Rightarrow & \lambda\beta + (A) \vdash (\lambda xy.x)MN = (\lambda xy.y)MN \\ \Rightarrow & \lambda\beta + (A) \vdash M = N \end{aligned}$$

□

3.14 证明命题 3.14: 设 R 是 Λ 上的二元关系, $M \in \text{NF}_R$, 则 (1) 不存在 $N \in \Lambda$ 使得 $M \rightarrow_R N$; (2) $M \rightarrow_R N \Rightarrow M \equiv N$.

Proof.

1. 根据 R 范式定义, M 不存在 R 可约子项, 所以 M 必然无法进行一步规约.

2. 若 $M \neq N$, 则必然存在一个长度大于 1 的 R 规约序列, 这意味着 M 必然可以进行一步规约, 与 (1) 矛盾.

□

3.16 试找出 $A \in \Lambda^\circ$ 使 A λ -定义函数 $f(m, n) = m + n$.

Proof.

$$\begin{aligned}\lceil m + n \rceil &= \lambda f x. f^m(f^n x) \\ &= \lambda f x. ((\lambda x. f^m x)(f^n x)) \\ &= \lambda f x. ((\lceil m \rceil f)(\lceil n \rceil f x))\end{aligned}$$

取 $A = \lambda m n f x. ((m f)(n f x))$.

□

3.17 试找出 $F \in \Lambda^\circ$ 使 F λ -定义函数 $f(m) = 3m$.

Proof.

$$\begin{aligned}\lceil 3m \rceil &= \lambda f x. f^{3m} x \\ &= \lambda f x. ((\lambda f x. f^m x)(\lambda x. f^3 x)) \\ &= \lambda f x. (\lceil m \rceil (\lceil 3 \rceil f))\end{aligned}$$

取 $A = \lambda m f x. (m(\lceil 3 \rceil f))$.

□

3.18 令 $D \equiv \lambda x y z. z(Ky)x$, 证明: 对于任意的 $X, Y \in \Lambda$,

$$DXY \lceil 0 \rceil = X,$$

$$DXY \lceil n + 1 \rceil = Y.$$

这里 $K \equiv \lambda x y. x$, $\lceil n \rceil \equiv \lambda f x. f^n x$.

Proof. 对于一般的 m , $DXY \lceil m \rceil = \lceil m \rceil (\lambda y. Y)X$.

1. $m = 0$ 时, $(\lambda f x. x)(\lambda y. Y)x = X$.
2. $m > 0$ 时 (即 $\exists n, m = n + 1$), $(\lambda f x. f^m x)(\lambda y. Y)x = (\lambda y. Y)^m x = Y$.

□

3.19 设 $\text{Exp} \equiv \lambda xy. yz$,

证明对于任意 $n \in \mathbb{N}, m \in \mathbb{N}^*$, $\text{Exp}^\top n^\top m^\top =^\top n^{m^\top}$

Proof.

$$\begin{aligned} \top m^\top n^\top &= (\lambda f x. f^m x)(\lambda f x. f^n x) \\ &= (\lambda x. (\lambda f x. f^n x)^m x) \\ &= (\lambda x. (\lambda f x. f^{(n^m)}) x) \\ &=^\top n^{m^\top} \end{aligned}$$

□

3.20 构造 $F \in \Lambda^\circ$ 使得对于任意 $n \in \mathbb{N}$, $F^\top n^\top =_\beta^\top 2^{n^\top}$.

Proof. 根据3.19, 可取 $F = \lambda n f x. n^\top 2^\top x$.

□

3.21 设 $f, g : \mathbb{N} \rightarrow \mathbb{N}, f = \text{Itw}[g]$, 即

$$f(0) = 0,$$

$$f(n+1) = g(f(n)),$$

且 $G \in \Lambda^\circ$ λ -定义了函数 g . 试求 $F \in \Lambda^\circ$ 使得 F λ -定义函数 f .

Proof. 需要 F 满足:

$$F^\top 0^\top =^\top 0^\top,$$

$$F^\top n + 1^\top = G(F^\top n^\top).$$

等价于不动点方程:

$$\begin{aligned} F &= \lambda n. D n^\top 0^\top (G(F(\text{pred } n))) \\ &= (\lambda z n. D n^\top 0^\top (G(z(\text{pred } n)))) F \end{aligned}$$

根据不动点定理, 可取 $F \equiv \Theta(\lambda z n. D n^\top 0^\top (G(z(\text{pred } n))))$.

□

3.22 证明引理 3.39.

Proof.

1. $\forall n \in \mathbb{N}. \text{var}(n) = \sharp(v^{(n)}) = [0, n]$ 显然是递归函数.
2. $\forall M, N \in \Lambda. \text{app}(\sharp M, \sharp N) = \sharp(MN) = [1, [\sharp M, \sharp N]]$ 显然是递归函数.
3. $\forall x \in \nabla, M \in \Lambda. \text{abs}(\sharp x, \sharp M) = \sharp(\lambda x. M) = [2, [\sharp x, \sharp M]]$ 显然是递归函数.
4. 对于 $\sharp^\Gamma n^\neg$, 尝试找到它的递归式:

$$\begin{aligned}
\sharp^\Gamma n + 1^\neg &= \sharp(\lambda f x. f^{n+1} x) \\
&= [2, [\sharp f, \sharp(\lambda x. f^{n+1} x)]] \\
&= [2, [\sharp f, [2, [\sharp x, \sharp f^{n+1} x]]]] \\
&= [2, [\sharp f, [2, [\sharp x, [1, [\sharp f, \sharp f^n x]]]]]] \\
&= [2, [\sharp f, [2, [\sharp x, [1, [\sharp f, (\pi_2)^4(\sharp^\Gamma n^\neg)]]]]]]
\end{aligned}$$

取 $h(z) = [2, [\sharp f, [2, [\sharp x, [1, [\sharp f, (\pi_2)^4(z)]]]]]]$, 则令:

$$\text{num}(0) = \sharp^\Gamma 0^\neg$$

$$\text{num}(n+1) = h(\text{num}(n)).$$

显然 $\text{num}(n) = \sharp^\Gamma n^\neg$ 且 $\text{num} \in \mathcal{PRF}$.

□

3.23 $f(n) = \underbrace{n^{\dots n}}_n$, 试构造 $F \in \Lambda^\circ$ 使 $F^\Gamma n^\neg = {}^\neg f(n)^\neg$ 对 $n \in \mathbb{N}^+$ 成立.

Proof.

$$F^\Gamma n^\neg = {}^\neg \underbrace{n^{\dots n}}_n^\neg = \underbrace{{}^\neg n^\neg \dots {}^\neg n^\neg}_n \quad n > 0$$

令 $CO_n = \lambda x. \underbrace{x \dots x}_n$, 注意 $\sharp x$ 为某常数 c , 现尝试说明 $f(n) = \sharp CO_n \in \mathcal{PRF}$:

$$\begin{aligned}
f(0) &= 0 \\
f(n+1) &= [2, [\sharp x, \underbrace{\sharp x \dots x}_{n+1}]] \\
&= [2, [\sharp x, [\sharp x, \underbrace{\sharp x \dots x}_n]]] \\
&= [2, [\sharp x, [\sharp x, [(\pi_2)^2(f(n))]]]]
\end{aligned}$$

显然 $f \in \mathcal{PRF}$. 根据递归函数的 λ -可定义性, 有 $F' \ulcorner n \urcorner = \ulcorner CO_n \urcorner$. 利用枚举子, 有 $E(F' \ulcorner n \urcorner) \ulcorner n \urcorner = CO_n \ulcorner n \urcorner$, 取 $F = \lambda n. E(F'n)n$ 即可.

□

3.24 构造 $H \in \Lambda^\circ$, 使得对于任意 $n \in \mathbb{N}, x_1, \dots, x_n \in \Lambda$, 有

$$H \ulcorner n \urcorner x_1 \dots x_n =_\beta [x_1, \dots, x_n].$$

Proof. 1. $l(n) = \sharp L_n \in \mathcal{GRF}$. 如下, $h \in \mathcal{PRF}$, 所以 $l \in \mathcal{PRF}$.

$$l(n) = [2, [\sharp 3, \sharp z x_1 \dots x_n]] = [2, [1, h(n)]]$$

$$h(n) = \sharp z x_1 \dots x_n$$

$$h(1) = \sharp z x_1 = [1, [1, \sharp x_1]] = [1, [1, [0, 1]]]$$

$$h(n+1) = [2, [1, h(n), \sharp x_{n+1}]] = [2, [h(n), [0, n+1]]]$$

2. $g(n) = \sharp M_n = \lambda x_1 \dots x_n. [x_1, \dots, x_n] \in \mathcal{PRF}$.

$$g(n) = [2, [\sharp x_1, [2, \dots [2, [\sharp x_n, l(n)] \dots]]]]$$

$$\text{let } f(i, y) = [2, [[0, i, y]]] \in \mathcal{PRF}$$

$$g(n) = f(1, f(2, \dots f(n-1, f(n, l(n)) \dots))) \in \mathcal{PRF}.$$

3. 有 G λ -定义 g . 取 $H \equiv \lambda z. E(Gz)$, 得

$$H \ulcorner n \urcorner x_1 \dots x_n =_\beta \lambda z. E(Gz) x_1 \dots x_n =_\beta [x_1, \dots, x_n].$$

□

3.25 证明：存在 $H_2 \in \Lambda^\circ$ ，使得对于任意 $F \in \Lambda$ ，有

$$H_2 \ulcorner n \urcorner =_\beta F \ulcorner H_2 \urcorner F \ulcorner \urcorner.$$

Proof. 即求第二不动点组合子.

令 $W \equiv \lambda xy. Ey \text{ (App (App } x \text{ (Num } x)) \text{ (Num } y))}$, $\Theta_2 = W \ulcorner W \urcorner$.
其中 E 为枚举子.

$$\begin{aligned} \Theta_2 \ulcorner F \urcorner &= W \ulcorner W \urcorner \ulcorner F \urcorner \\ &= E \ulcorner F \urcorner \text{ (App (App } \ulcorner W \urcorner \text{ (Num } \ulcorner W \urcorner)) \text{ (Num } \ulcorner F \urcorner))} \\ &= F \text{ App } \ulcorner W \urcorner \ulcorner W \urcorner \ulcorner \ulcorner F \urcorner \urcorner \\ &= F \ulcorner \Theta_2 \urcorner \ulcorner F \urcorner \end{aligned}$$

□

5 图灵机

5.1 构造机器计算函数 $f(x, y, z) = y$.

Proof. 取 M 为 $\boxed{\text{erase}} \Rightarrow \boxed{\text{copy}_2} \Rightarrow \boxed{\text{erase}} \Rightarrow \boxed{\text{erase}}$. □

5.2 构造机器 $\boxed{\text{copy}_1}$ **使得** $\boxed{\text{copy}_1} \mid 01^x 0 \dots \rightarrow 01^x 01^x 0 \dots$.

Proof. 构造机器 $\boxed{\text{cbit}}$ 使得对于 $m, n, k > 0$ ，有 $\boxed{\text{cbit}} \mid 01^m 0^n 01^k \dots \rightarrow u : 01^{\overset{\uparrow}{m+1}} 0^{n-1} 01^{k+1} \dots$ ；对于 $m, k > 0, n = 0$ ，有 $\boxed{\text{cbit}} \mid 01^m 01^k \dots \rightarrow v : 01^{\overset{\uparrow}{m}} 01^k \dots$ 其状态转移表设计如下.

	0	1	解释
1	1R2	1R1	$1 : 01^m 0^n 01^k \rightsquigarrow$ \uparrow $1 : 01^m 00^n 1^k \dots \rightsquigarrow$ \uparrow $2 : 01^{m+1} 0^n 1^k \dots$
2	0O3	1L8	$2 : 01^{m+1} 0^n 1^k \dots \rightsquigarrow$ \uparrow $3 : 01^{m+1} 0^n 1^k \dots$ \uparrow <p>或</p> $2 : 01^{m+1} 1^k \dots \rightsquigarrow$ \uparrow $8 : 01^m 11^k \dots$
3	0R3	1R4	$3 : 01^{m+1} 0^n 1^k \dots \rightsquigarrow$ \uparrow $3 : 01^{m+1} 0^n 1^k \dots \rightsquigarrow$ \uparrow $4 : 01^{m+1} 0^n 11^{k-1} \dots$
4	1L5	1R4	$4 : 01^{m+1} 0^n 11^{k-1} \dots \rightsquigarrow$ \uparrow $4 : 01^{m+1} 0^n 1^k 0 \dots \rightsquigarrow$ \uparrow $5 : 01^{m+1} 0^n 1^{k-1} 11 \dots$
5	0L6	1L5	$5 : 01^{m+1} 0^n 1^{k-1} 11 \dots \rightsquigarrow$ \uparrow $5 : 01^{m+1} 0^{n-1} 01^{k+1} \dots \rightsquigarrow$ \uparrow $6 : 01^{m+1} 0^{n-2} 001^{k+1} \dots$
6	0L6	1L7	$6 : 01^{m+1} 0^{n-2} 001^{k+1} \dots \rightsquigarrow$ \uparrow $6 : 01^m 10^n 1^{k+1} \dots \rightsquigarrow$ \uparrow $7 : 01^{m-1} 110^n 1^{k+1} \dots$
7	0Ru	1L7	$7 : 01^{m-1} 110^n 1^{k+1} \dots \rightsquigarrow$ \uparrow $7 : 01^{m+1} 0^n 1^{k+1} \dots \rightsquigarrow$ \uparrow $u : 01^{m+1} 0^n 1^{k+1} \dots$
8		0L9	$8 : 01^m 11^k \dots \rightsquigarrow$ \uparrow $9 : 01^{m-1} 101^k \dots$ \uparrow
9	0Rv	1L9	$9 : 01^{m-1} 101^k \dots \rightsquigarrow$ \uparrow $9 : 01^m 01^k \dots \rightsquigarrow$ \uparrow $u : 01^m 01^k \dots$

构造机器 M_1 使得 $M_1 \mid 01^m \dots \rightarrow 010^{m-1}01\dots$

	0	1	解释
1	0R2	1R1	$1 : 01^m \dots \rightsquigarrow$ \uparrow $1 : 01^m 0 \dots \rightsquigarrow$ \uparrow $2 : 01^m 0 \dots$ \uparrow
2	0R3		$2 : 01^m 0 \dots \rightsquigarrow$ \uparrow $3 : 01^m 00 \dots$ \uparrow
3	1L4		$3 : 01^m 00 \dots \rightsquigarrow$ \uparrow $4 : 01^m 01 \dots$ \uparrow
4	0L5		$4 : 01^m 01 \dots \rightsquigarrow$ \uparrow $5 : 01^{m-1} 101 \dots$ \uparrow
5	0R6	0L5	$5 : 01^{m-1} 101 \dots \rightsquigarrow$ \uparrow $5 : 00^m 01 \dots \rightsquigarrow$ \uparrow $6 : 000^{m-1} 01 \dots$ \uparrow
6	1Ou		$6 : 000^{m-1} 01 \dots \rightsquigarrow$ \uparrow $u : 010^{m-1} 01 \dots$ \uparrow

则, 机器 $\boxed{\text{copy}_1} = M_1 \Rightarrow \text{repeat } \boxed{\text{cbit}}$.

□

5.3 构造机器计算函数 $f(x, y) = x * y$.

Proof.

构造机器 M_{pre} 使得, $M_{pre} \mid 01^{x+1}01^{y+1} \rightarrow 001^{x+1}01^{y+1}01^{y+1} \dots$

构造机器 M_1 使得,

$m > 1$ 时, 有 $M_1 \mid \dots 01^{m+1}0^r 1^{kn+1}01^{n+1} \dots \rightarrow u : \dots 01^m 0^{r'} 1^{kn+n+1}01^{n+1} \dots$,

$m = 1$ 时, 有 $M_1 \mid \dots 010^r 1^{kn+1}01^{n+1} \dots \rightarrow v : \dots 01^{kn+1}0 \dots$

那么, $M = M_{pre} \Rightarrow \text{repeat } M_1$ 则计算了函数 f .

$M_{pre} = \boxed{\text{shiftr}} \Rightarrow \boxed{\text{copy}_1} \Rightarrow \boxed{\text{shiftl}}$.

$M_1 = M_{check} \Rightarrow M' \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{\text{copy}_1} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{\text{add}} \Rightarrow \boxed{\text{shiftr}}$, 其中 $M'(1,1) = 0Ru$; $M_{check}(1,1) = 1R2$, $M_{check}(2,1) = 1Lu$, $M_{check}(2,0) = 0L3$, $M_{check}(3,1) = 0R4$, $M_{check}(4,0) = 0R4$, $M_{check}(4,1) = 1R5$, $M_{check}(5,1) = 1R5$, $M_{check}(5,0) = 0R6$, $M_{check}(6,0) = 0R6$, $M_{check}(6,1) = 0R7$, $M_{check}(7,1) = 0R7$, $M_{check}(7,0) = 0L8$, $M_{check}(8,0) = 0L8$, $M_{check}(8,1) = 1L9$, $M_{check}(9,1) = 1L9$, $M_{check}(9,0) = 0Rv$.

(注: M_1 用于递减迭代次数; M_{check} 用于单次循环开始前判断后续迭代次数是否为 0, 是则进行结果的整理并返回) . \square

5.4 构造机器计算函数 $f(x) = 2^x$.

Proof.

构造机器 M_{pre} 使得, $M_{pre} \mid 01^m 0 \cdots \rightarrow u : 0 \overset{\uparrow}{1} 1^m 0110 \cdots$,

构造机器 M_1 使得,

$x > 0$ 时, $M_1 \mid \cdots 01^{x+1} 01^y \cdots \rightarrow u : \cdots 0 \overset{\uparrow}{1} 1^x 01^{2y} \cdots$,

$x = 0$ 时, $M_1 \mid \cdots 0101^y \cdots \rightarrow v : \cdots 0 \overset{\uparrow}{1} 1^y 0 \cdots$,

取 M 为 $M' \Rightarrow M_0 \Rightarrow M_{pre} \Rightarrow \text{repeat } M_1$, 得到所求的机器. 其中 $M'(1,1) = 0Ru$, $M_0(0,0) = 1Ov$. (M' 计算 $x-1$, 即“翻倍次数”; M_0 用于处理 $x=0$ 的情况) .

M_{pre} 是简单的, 略去列表.

$M_1 = M_{check} \Rightarrow M' \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{\text{copy}_1} \Rightarrow \boxed{\text{add}} \Rightarrow \boxed{\text{shiftr}}$. 其中, $M_{check}(1,1) = 1R2$, $M_{check}(2,1) = 1Lu$, $M_{check}(2,0) = 0L3$, $M_{check}(3,1) = 0R4$, $M_{check}(4,0) = 0R4$, $M_{check}(4,1) = 1Ov$. (M_{check} 仍然承担对循环变量的判断, 并在确定循环终止时整理数据与带状态) . \square

5.5 求机器的输出.

Proof. 擦除偶数位的 1, 并由于指针抵达最左侧而运行终止. \square

5.6 求机器的输出.

Proof. 最终机器状态为: $7: 0^{n+m+k+3}11110\dots$, 即 $f(x, y, z) = 3$. \square

5.7 构造机器计算函数 $f(x) = \lfloor \sqrt{x} \rfloor$.

Proof. $f(x) = \mu y. [S(x) \div \text{sq}(S(y))]$, $\boxed{\text{sq}} = \boxed{\text{copy}_1} \Rightarrow \boxed{\text{mul}}$. 类似书中定理 5.16 的构造, 我们取 $M = M_{pre} \Rightarrow \text{repeat } M_1$.

其中, $M_{pre} \mid 01^x 0 \dots \rightarrow 01^x 0 \underset{\uparrow}{1} 01^x 010 \dots$ 如此构造: $M_{pre} = M' \Rightarrow \boxed{\text{shiftrl}} \Rightarrow \boxed{\text{copy}_2}^2$. $M'(1, 1) = 1R1, M'(1, 0) = 0R2, M'(2, 0) = 1Ou$.

设 $g(x, y) = S(x) - \text{sq}(S(y))$, $M_1 \mid 01^{x+1} 0 \underset{\uparrow}{1} 01^{y+1} 01^{x+1} 01^{y+1} \dots \rightarrow 01^{x+1} 01^{y+1} 0 \underset{\uparrow}{1} 01^{g(x,y)+1} \dots$, 取 $M_1 = \boxed{\text{copy}_2} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{S} \Rightarrow \boxed{\text{shiftrl}}^2 \Rightarrow \boxed{\text{erase}} \Rightarrow \boxed{\text{copy}_2} \Rightarrow \boxed{S} \Rightarrow \boxed{\text{sq}} \Rightarrow \boxed{\text{shiftrl}} \Rightarrow \boxed{\text{sub}} \Rightarrow \boxed{\text{compress}}$.

如下构造: $\boxed{f} = M_{pre} \Rightarrow \text{repeat } (M_1 \Rightarrow M_{check})$. M_{check} 对结果是否为 0 进行检查, 如不为 0, 则递增 y 进行下一轮计算, 若为 0, 则整理结果, 结束运行.

$$M_{check} = M'[u_1 := e_1][u_2 := e_2] \cup (M_{ok} + e_1) \cup (M_{cont} + e_2),$$

$$M'(1, 1) = 1R2, M'(2, 0) = 0Lu_1, M'(2, 1) = 0Lu_2,$$

$$M_{ok} = \boxed{\text{shiftrl}}^2 \Rightarrow \boxed{\text{proj}_3^2},$$

$$M_{cont} = \boxed{\text{shiftrl}}^2 \Rightarrow \boxed{\text{copy}_3}^2 \Rightarrow \boxed{\text{erase}} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{\text{compress}}. \quad \square$$

5.8 有 $\boxed{f_1}$ 计算 $f_1(x)$, $\boxed{f_2}$ 计算 $f_2(x)$, 试构造 \boxed{f} , 计算 $f(x) = f_1(x) + f_2(x)$.

Proof. $\boxed{f} = \boxed{\text{copy}_1} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{f_1} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftrl}} \Rightarrow \boxed{\text{copy}_2} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{f_2} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftrl}}^2 \Rightarrow \boxed{\text{erase}} \Rightarrow \boxed{\text{add}}$.

(注: 前面实现的 $\boxed{\text{copy}_1}$ 不会移动指针到下一项(复制项); $\boxed{\text{copy}_n}, n > 1$ 按书中实现, 会移动指针到下一项). \square

5.9 设 $f(x) = h(g_1(x), g_2(x), g_3(x))$, 试由 $\boxed{g_1}$, $\boxed{g_1}$, $\boxed{g_1}$ 和 \boxed{h} 构造机器 \boxed{f} .

Proof. $\boxed{f} = \boxed{\text{copy}_1} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{g_1} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{\text{copy}_2} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \boxed{g_1} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftr}}^2 \Rightarrow \boxed{\text{copy}_3} \Rightarrow \boxed{\text{shiftr}}^2 \Rightarrow \boxed{g_1} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftr}}^2 \Rightarrow \boxed{h}$ \square

5.10 设 $f: \mathbb{N} \rightarrow \mathbb{N}$ 定义如下:

$$f(0) = 0,$$

$$f(x+1) = g(f(x)).$$

. 证明, 若 g 为 Turing-可计算, 则 f 为 Turing-可计算.

Proof.

$M'(1, 1) = 1R2, M'(2, 0) = 0Lu_1, M'(2, 1) = 0Lu_2$, 用于条件分支.

$M_{pre}(1, 1) = 1R1, M_{pre}(1, 0) = 0R2, M_{pre}(2, 0) = 1Ou$, 初始化结果 0.

$\boxed{f} = M_{pre} \Rightarrow \boxed{\text{shiftr}} \Rightarrow \text{repeat } M'[u_1 := e_1][u_2 := e_2] \cup (M_{ok} + e_1 \downarrow v) \cup e_2 + (\boxed{\text{shiftr}} \Rightarrow \boxed{g} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftr}})$. 根据 x 应用函数 g 到中间结果. 有两个分支.

$$M_{ok} = \boxed{\text{erase}}.$$

\square

5.11 构造机器计算函数 $f(x, y) = x \dot{-} y$.

Proof. 算法: 不断对 y, x (按序) 各自递减 1, 并在递减的同时检查, 若 y 减至 0 (指长度为 0), 直接归位指针即可. 若 x 减少为 0, 则清零 y , 输出 $\bar{0}$.

$$M = \boxed{\text{shiftr}} \Rightarrow \text{repeat } M_2,$$

$$M_2(1, 1) = 0R2, M(2, 0) = 0Lu_1, M(2, 1) = 1L3, M(3, 0) = 0L3, M(3, 1) = 0L4, M(4, 0) = 0Ru_2, M(4, 1) = 1R5, M(5, 0) = 0R5, M(5, 1) = 1Ou,$$

$$y \text{ 清零: } M_2(u_1, 0) = 0Lu_1, M_2(u_1, 1) = 1L(u_1 + 1), M_2(u_1 + 1, 0) = 0Rv,$$

x 清零: $M_2(u_2, 0) = 0Ru_2, M_2(u_2, 1) = 0R(u_2 + 1), M_2(u_2 + 1, 1) = 0R(u_2 + 1), M_2(u_2 + 1, 0) = 1O(v)$. \square

5.12 证明 $\text{Even} = \{2x : x \in \mathbb{N}\}$ 是 Turing-可计算的.

Proof. 直接构造图灵机 M : $M(1, 1) = 0R2, M(2, 1) = 0R3, M(3, 1) = 0R2, M(2, 0) = 1Ou, M(3, 0) = 1R3, M(4, 0) = 1Ru$. \square

5.13 证明 $S = \{a_1, a_2, \dots, a_k\}$ 是 Turing-可计算的.

Proof. 令 a_1, \dots, a_k 为升序排列, $a_k = n$. 则构造行数为 $n + 8$ 的图灵机.

1. 越过第一个 1: $M(1, 1) = 0R2$,
2. 与每个集合内的数进行比较, 相等则结束 (输出 0), 否则递增继续尝试, $\forall j \leq n, j \in S \Rightarrow (M(1+j, 0) = 0Lu_1) \wedge (M(1+j, 1) = 1R(1+j+1))$,
3. 与非集合内的数进行比较, 相等则结束 (输出 1), 否则递增继续尝试, $\forall j \leq n, j \notin S \Rightarrow (M(1+j, 0) = 0Lu_2) \wedge (M(1+j, 1) = 1R(1+j))$,
4. 整理结果, 输出 0: $M(u_1, 1) = 0L1, M(u_1, 0) = 0R(u_1 + 1), M(u_1 + 1, 1) = 1R(u_1 + 2)$,
5. 整理结果, 输出 1: $M(u_2, 1) = 0L1, M(u_2, 0) = 0R(u_2 + 1), M(u_2 + 1, 1) = 1R(u_2 + 2), M(u_2 + 2, 1) = 1R(u_2 + 3)$.

\square

5.14 设 $f: \mathbb{N} \rightarrow \mathbb{N}$ 是 Turing-可计算的, 构造机器 M 输出 f 的最小零点.

Proof. 设 M 对应的函数为 g , 它求解 f 的最小零点, 即 $g(m) = \mu x. [f(x)]$. 设计算 f 的机器为 \boxed{f} .

思路: 将输入归零, 从零开始不断尝试判断是否为零点. 设有 M_{check} 对结果进行判断, 在符合要求时整理结果、结束执行, 否则整理结果、开启下一次循环. 那么有 $M = \text{repeat } (\boxed{\text{copy}_1} \Rightarrow \boxed{f} \Rightarrow M_{check})$.

M_{check} 定义如下:

1. 当发现结果为 0 时, 将指针移动到最开始. $M(1, 1) = 0R2, M(2, 0) = 0L3, M(3, 0) = 0L3, M(3, 1) = 1L4, M(4, 1) = 1L4, M(4, 0) = 0Rv$.
2. 当发现结果不为 0 时, 清除结果, 将输入递增, 将指针移动到最初. $M(1, 1) = 0R2, M(2, 1) = 0R5, M(5, 1) = 0R5, M(5, 0) = 0L6, M(6, 0) = 0L6, M(6, 1) = 1R7, M(7, 0) = 1L8, M(8, 1) = 1L8, M(8, 0) = 0Ru$.

□

5.15 证明 g 是一般递归函数, 其中

$$g(m) = \begin{cases} \sharp \hat{M}, & \text{若有机机器 } M \text{ 使得 } m = \sharp M, \\ 0 & \text{否则,} \end{cases}$$

Proof. 5.17 有结论可以找到机器 M 判断输入是否为某图灵机的编码, 该判断对应的函数是一般递归的. 常数函数显然是一般递归的, 设为 M_1 . \Rightarrow 操作, 将两个图灵机表格合并, 对第二个表进行递增, 只需要遍历, 显然也是一般递归的, 即计算 $\sharp \hat{M} = \sharp(M_1 \Rightarrow M)$ 一定是一般递归的. 因而 $g(m)$ 一定是一般递归的. □

5.16 证明 $e(m, l) = \langle \#d(a_j, k), \#p(a_j, k), \#s(a_j, k) \rangle$ 为初等函数.

Proof. $e(m, l)$ 寻找机器编码 m 的特定单元格 (j, k) , 其中 $j = (l)_0, k = (l)_1$. 这显然是初等的, 理由如下:

1. 寻找第 j 行: $\text{ep}_j(m)$ 是初等的,
2. 取位置 2,3,4 或者 5,6,7, 显然也是初等.

□

5.17 令 $S = \{\#M : M \text{ 为 Turing-机}\}$, 证明 S 是 Turing-可计算的.

Proof.

已知 $n \in S \Leftrightarrow \exists M, n = \#M$,

设 $k(n) = (\max x \leq n. p_x \mid n) + 1$,

则 M 有 $1 + k(n)$ 行 (写作 $r_0, \dots, r_{k(n)}$), $n = \langle \#r_1, \dots, \#r_{k(n)} \rangle$.

对于 $i \leq k(n)$, 可用 $\text{ep}_i(n)$ 取机器第 i 行编码,

此时构造函数 $\text{codeL}(m, n)$ 对行编码合法性进行检查, 其中 m 为行编码, n 为行数.

$\text{codeL}(m, n) = (\text{ep}_0(m) \leq k(n) \wedge \text{ep}_1(m) \in \{0, 1\} \wedge \text{ep}_2(m) \in \{\#L, \#O, \#R\}$

$\wedge \text{ep}_4(m) \in \{0, 1\} \wedge \text{ep}_5(m) \in \{\#L, \#O, \#R\})$

$\wedge (\text{ep}_0(m) \leq k(n) \wedge \text{ep}_1(m) \in \{0, 1\} \wedge \text{ep}_2(m) \in \{2, 3, 4\}$

$\wedge \text{ep}_4(m) = \text{ep}_5(m) = \text{ep}_6(m) = 4)$

$\wedge (\text{ep}_0(m) \leq k(n) \wedge \text{ep}_1(m) = \text{ep}_2(m) = \text{ep}_3(m) = 2 \wedge \text{ep}_4(m) \in \{0, 1\}$

$\wedge \text{ep}_5(m) \in \{2, 3, 4\})$

可知, $n \in S \Leftrightarrow \forall i \leq k(n), \text{codeL}(\text{ep}_i(n), n)$,

这是个初等谓词, 所以属于 \mathcal{GRF} , 所以 S 可判定.

□

5.18 由 CT 证明函数 $g(n)$ 可计算, 这里 $g(n)$ 为自然常数 e 的十进制展开式的第 n 个数字

Proof. 只需说明 $g(n)$ 在直觉上可计算.

$$e = \sum_{i=0}^m \frac{1}{i!} + \frac{e^\theta}{(m+1)!}, \quad 0 < \theta < 1$$

显然根据 n 不断尝试调整展开长度即可, 直到余项一定小于 $\frac{1}{10^n}$. □

5.19 (1) 什么是停机问题? (2) 什么是可判定问题? (3) 停机问题可判定吗?

Proof.

1. 停机问题指集合 \hat{K} 是否是可判定的.
2. 可判定问题指: 特征函数是 Turing-可计算的问题.
3. 不可判定.

□

5.20 (1) 什么是通用图灵机? (2) 通用图灵机起什么作用?

Proof.

1. 通用图灵机指可以执行任意图灵机 (以其编码作为输入) 的图灵机. 它的存在意味着设计实体计算机时, 不必针对性地为每一种算法单独设计一种计算机, 而是只需要设计通用计算机. 在计算特定算法时, 只需要将其编码输入通用计算机即可. 这也说明了不同算法的基本计算单元是一致的、简单的.
2. 其作用是作为通用计算机的理论模型.

□