

# 《计算模型导引》第五章参考答案\*

2019 年 6 月 18 日

## 习题 5.1

主要思想是先后抹除  $\bar{x}$  最后移到  $\bar{y}$  的头. 机器如下:

	0	1	解释
1	0R2	0R1	抹除第一个 $\bar{x}$ 并指向 $\bar{y}$ 头
2	0R3	1R2	移动到第二个 $\bar{x}$ 头
3	0L4	0R3	抹除第二个 $\bar{x}$
4	0L4	1L5	越过 $\bar{y}$ 的最后一个 1
5	0R6	1L5	移到 $\bar{y}$ 头

## 习题 5.2

主要思想是每次抹掉输入的一个 1 对应两个输出各增加一个 1. 即设计机器  $M$  实现如下功能:

$$\begin{aligned}
 M|1 : 0 \dots 0 \underset{\uparrow}{1} 0 1^k 0 1^k 0 \dots &\rightarrow u : 0 \dots 0 \underset{\uparrow}{1}^{k+1} 0 1^{k+1} 0 \dots \\
 M|1 : 0 \dots 0 \underset{\uparrow}{1}^{l+1} 0 1^k 0 1^k 0 \dots &\rightarrow v : 0 \dots 0 \underset{\uparrow}{1}^l 0 1^{k+1} 0 1^{k+1} 0 \dots
 \end{aligned}$$

这里  $k \geq 0$ ,  $u$  为停机状态. 这样  $M[v := 1]$  既可实现循环.

具体实现如下:

---

\*Ver. 0.2. 原答案由宋方敏教授给出手稿, 然后由丁超录入修订补充. 此文档来源为[https://github.com/sleepycoke/NJU\\_Com\\_Models](https://github.com/sleepycoke/NJU_Com_Models)

	0	1	解释
1		0R2	输入移除 1
2	0R3	1R12	判断输入是否还有 1, 移到尾部再跳过后面的 0
3	1R4	1R3	输入为 0, 那么走第一个分支 3-8. 将第一个 $1^k$ 后的 0 改为 1
4		0R5	将第二个 $1^k$ 的第一个 1 改为 0
5	1R6	1R5	将第二个 $1^k$ 后的 0 改为 1
6	1L7		再添个 1
7	0L8	1L7	跳过 $1^{k+1}$
8	0R9	1L8	移到输入头, 停机
12	0R13	1R12	输入不为 0, 那么走第二个分支 12-19. 跳过第一个输入
13	1R14	1R13	将第一个 $1^k$ 后的 0 改为 1
14		0R15	将第二个 $1^k$ 的第一个 1 改为 0
15	1R16	1R15	将第二个 $1^k$ 后的 0 改为 1
16	1L17		再添个 1
17	0L18	1L17	跳过 $1^{k+1}$
18	0R19	1L18	再跳过 $1^{k+1}$
19	0R1	1L19	再移到输入头, 循环

### 习题 5.3

大家可能在某些参考资料上看到了如下的乘法图灵机  $M_1$ , 但它并不适用于我们对输入的定义. 实际上,  $M_1$  接收的输入为  $01^x01^y0\cdots$  而不是我们定义的  $01^{x+1}01^{y+1}\cdots$ .

	0	1	解释
1	0R10	0R2	判断 $x$ 是否为 0, 不为 0 抹除一个 1 进入 2-8 作一次累加
2	0R3	1R2	移到第一串 1 的尾部
3	0L8	0R4	3-7 作一次累加, 作过 $y$ 次后转到 8. 进入 4 之前抹掉 1 留作标记给 7 找回用
4	0R5	1R4	找到本串 1 的尾部
5	1L6	1R5	找到 $1^j$ 的尾部, +1
6	0L7	1L6	找到 $1^{j+1}$ 的头
7	1R3	1L7	找到 $1^i$ 的头, 补回一个 1
8	0L9	1L8	一次累加结束
9	0R1	1L9	找到第一串 1 的头
10	0R11	0R10	抹除第一串 1

我们可以先证明

$$M_1|1:0\cdots01^x01^y0\cdots \rightarrow u:0\cdots01^{x*y}0\cdots$$

$\uparrow$ 
 $\uparrow$

而理解  $M_1$  和关键在于理解 3-7 状态完成了以下操作:

$$M_1|3:0\cdots011^i01^j\cdots \rightarrow 3:11^i01^{j+1}0\cdots$$

$\uparrow$ 
 $\uparrow$

从而  $M_1$  和通过 3-8 状态完成了以下操作:

$$M_1|3 : \underset{\uparrow}{1^i}01^j \dots \rightarrow 8 : \underset{\uparrow}{1^i}01^{j+i}0 \dots$$

即实现了一次累加. 这样总体上每抹除第一个输入的一个 1 就, 对应一次加法, 整体实现了乘法.

最后为了符合我们的输入输出规范, 实现机器  $M_2, M_3$  分别实现:

$$M_2|1 : \underset{\uparrow}{0}1^{x+1}01^{y+1}0 \dots \rightarrow u : \underset{\uparrow}{0}01^x01^y0 \dots$$

$$M_3|1 : \underset{\uparrow}{0}1^x00 \dots \rightarrow v : \underset{\uparrow}{0}1^{x+1}0 \dots$$

这两个机器很容易实现不赘述.

从而乘法机器为

$$M_2 \Rightarrow M_1 \Rightarrow M_3$$

## 习题 5.4

参照第 133 页定理 5.14, 我们需要作一个预处理, 添上一个输入  $\bar{1}$ . 具体可以由如下机器  $M_3$  实现:

	0	1
1	0R2	1R1
2		1R3
3	1L4	
4		1L5
5	0L6	
6	0R7	1L6

如书中第 134 页构造机器  $M_1$ :

	0	1
1		0R2
2	0R4	1R3
3	0R4	1R3

令  $M_2$  为  $M_1 \Rightarrow \boxed{\text{double}} \Rightarrow \boxed{\text{compress}} \Rightarrow \boxed{\text{shiftl}}$ , 从而

若  $x = 0$ , 则  $M_2|1 : \underset{\uparrow}{0}\bar{x}0\bar{y}0 \dots \rightarrow v : \underset{\uparrow}{0}00\bar{y}0 \dots$

若  $x > 0$ , 则  $M_2|1 : \underset{\uparrow}{0}\bar{x}0\bar{y}0 \dots \rightarrow w : \underset{\uparrow}{0}0\bar{x} - 10g(\bar{y})0 \dots$ , 其中  $w$  为  $M_2$  的输出时的状态.

令  $\boxed{f} = M_2[w := 1]$ ,  $M_3 \Rightarrow \boxed{f}$  即为所求.

## 习题 5.5

- 若  $x = 0$ :

$$1 : 010000 \dots \quad (1R2)$$

$$2 : 010000 \dots \quad (0L3)$$

$$3 : 010000 \dots \quad (1L3)$$

$$3 : 010000 \dots \quad (0L3)$$

$$3 : 010000 \dots \quad (\text{halt})$$

- 若  $x > 0$ :

$$1 : 011^x 0000 \dots \quad (1R2)$$

$$2 : 011^x 0000 \dots \quad (0R1)$$

$$1 : 0101^{x-1} 0000 \dots \quad (1R2)$$

$$2 : 01011^{x-2} 0000 \dots \quad (0R1)$$

$$\dots \quad (1R2, 0R1)$$

$$3 : \overbrace{0101 \dots 01}^{\lfloor x/2 \rfloor + 1 \text{ 个 } 01} 0000 \dots \quad (1L3)$$

$$\dots \quad (0L3, 1L3)$$

$$3 : \overbrace{0101 \dots 01}^{\lfloor x/2 \rfloor + 1 \text{ 个 } 01} \dots 010000 \dots \quad (0L3)$$

$$3 : \overbrace{0101 \dots 01}^{\lfloor x/2 \rfloor + 1 \text{ 个 } 01} \dots 010000 \dots \quad (\text{halt})$$

总之就是擦除了第偶数个 1, 然后指向第一个 0 左边 (越界).

## 习题 5.6

易见  $M_2 | 1 : 0\bar{x}0\bar{y}0 \dots \rightarrow 1 : 000\bar{y}0 \dots$

从而  $M_2 | 1 : 01^n 01^m 01^k 0\bar{y}0 \dots \rightarrow 1 : 0^{n+m+k+3} 00 \dots (\text{停})$

又易见  $M_2 | 1 : 000 \dots \rightarrow 7 : 011110 \dots$

于是  $M_2 | 1 : 01^n 01^m 01^k 0\bar{y}0 \dots \rightarrow 7 : 0^{m+n+k+4} 11110 \dots (\text{停})$

## 习题 5.7

由于

$$y = \lfloor \sqrt{x} \rfloor \Leftrightarrow y \leq \sqrt{x} < y+1 \Leftrightarrow y^2 \leq x < (y+1)^2$$

所以

$$f(x) = \mu y. x < (y+1)^2 = \mu y. N^2((sx) \div (sy)^2)$$

首先平方机器 `sqrt` 可由 `copy1`  $\Rightarrow$  `shiftl`  $\Rightarrow$  `multi` 构造, 其中 `multi` 为习题 5.3 的乘法机器. 然后减法机器也在习题 5.11 中给出, 于是  $f$  涉及到的机器都已经构造出, 那么 `f` 是可构造的. 具体构造过程可对照教材实现.

## 习题 5.10

参照第 133 页定理 5.14, 我们需要作一个预处理, 添上一个输入  $\bar{0}$ . 具体可以由如下机器  $M_3$  实现:

	0	1
1	0R2	1R1
2	1L3	
3	0L4	
4	0R5	1L4

接下来再如法构造 `f` 既可. 最后结果为  $M_3 \Rightarrow \text{f}$ .

## 习题 5.11

注意到  $0 \div y = 0$ ,  $x \div 0 = x$ ,  $(x+1) \div (y+1) = x \div y$ , 我们可以分情况处理. 机器如下:

	0	1	解释
1		0R2	将 $x$ 减 1
2	0R3	1R4	判断原 $x$ 是否为 0
3	1O15	0R3	原 $x$ 为 0 那么把 $\bar{y}$ 修改为 1 输出
4	0R11	1R4	原 $x$ 不为 0 找到 $\bar{y}$ 的头
5	0L6	1R5	找到 $\bar{y}$ 的尾部
6		0L7	将 $y$ 减 1
7	0L10	1L7	找到 $\bar{x}$ 的尾
10	0R1	1L10	找到 $\bar{x}$ 的头, 迭代
11		1R12	跳过 $\bar{y}$ 的第一个 1
12	0L13	1R5	判断 $y$ 是否为 0
13		0L16	原 $y$ 为 0, 抹掉 $\bar{y}$
14	0R15	1L14	原 $y$ 为 0, 找到 $\bar{x}$ 的头, 输出 $x$
15			停机
16	0L14		前左跳过一个 0

计算过程如下:

- 若  $x = 0$ :

$$\begin{array}{ll}
 1 : 010\overline{y}000\dots & (0R2) \\
 \quad \uparrow & \\
 2 : 000\overline{y}000\dots & (0R3) \\
 \quad \uparrow & \\
 3 : 000\overline{y}0000\dots & (0R3) \\
 \quad \uparrow & \\
 3 : 0000\overline{y-1}0000\dots & (0R3) \\
 \quad \uparrow & \\
 \dots & (0R3) \\
 3 : 000\dots 00000\dots & (1O15) \\
 \quad \uparrow & \\
 15 : 000\dots 10000\dots & (\text{halt}) \\
 \quad \uparrow &
 \end{array}$$

- 若  $x > 0$ ,  $y = 0$ :

$$\begin{array}{ll}
 1 : 011^x01000\dots & (0R2) \\
 \quad \uparrow & \\
 2 : 001^x01000\dots & (1R4) \\
 \quad \uparrow & \\
 \dots & (1R4) \\
 4 : 001^x01000\dots & (0R11) \\
 \quad \uparrow & \\
 11 : 001^x01000\dots & (1R12) \\
 \quad \uparrow & \\
 12 : 001^x01000\dots & (0L13) \\
 \quad \uparrow & \\
 13 : 001^x01000\dots & (0L16) \\
 \quad \uparrow & \\
 16 : 001^x00000\dots & (0L14) \\
 \quad \uparrow & \\
 14 : 001^{x-1}10000\dots & (1L14) \\
 \quad \uparrow & \\
 \dots & (1L14) \\
 14 : 001^x0000\dots & (0R15) \\
 \quad \uparrow & \\
 15 : 001^x0000\dots & (\text{halt}) \\
 \quad \uparrow &
 \end{array}$$

- 若  $x > 0, y > 0$ :

$$\begin{array}{ll}
1 : 011^x 0\bar{y}000 \dots & (0R2) \\
\quad \uparrow & \\
2 : 001^x 0\bar{y}000 \dots & (1R4) \\
\quad \uparrow & \\
\dots & (1R4) \\
4 : 001^x 0\bar{y}000 \dots & (0R11) \\
\quad \uparrow & \\
11 : 001^x 0\bar{y}000 \dots & (1R12) \\
\quad \uparrow & \\
12 : 001^x 01\overline{y-1}000 \dots & (1R5) \\
\quad \uparrow & \\
\dots & (1R5) \\
5 : 001^x 0\bar{y}000 \dots & (0L6) \\
\quad \uparrow & \\
6 : 001^x 01^y 1000 \dots & (0L7) \\
\quad \uparrow & \\
7 : 001^x 01^{y-1} 1000 \dots & (1L7) \\
\quad \uparrow & \\
\dots & (1L7) \\
7 : 001^x 0\bar{y}000 \dots & (0L10) \\
\quad \uparrow & \\
10 : 001^{x-1} 101^y 000 \dots & (1L10) \\
\quad \uparrow & \\
\dots & (1L10) \\
10 : 001^x 01^y 000 \dots & (0R1) \\
\quad \uparrow & \\
1 : 001^x 01^y 000 \dots & (\text{Compute } (x-1) \div (y-1)) \\
\quad \uparrow &
\end{array}$$

## 习题 5.12

欲证 *Even* Turing-可计算, 只需构造机器  $E$  满足:

$$\begin{array}{l}
E|1 : 01^{2x+1}0 \dots \rightarrow u : 0 \dots 010 \dots \\
\quad \uparrow \qquad \qquad \qquad \uparrow \\
E|1 : 01^{2x}0 \dots \rightarrow u : 0 \dots 0110 \dots \\
\quad \uparrow \qquad \qquad \qquad \uparrow
\end{array}$$

其中  $E(1, u)$  无定义.  $E$  可如下构造:

	0	1
1	1L3	0R2
2	1O4	0R4
3	1O3	

$E(1, 1)$  的  $E(1, 2)$  形成循环将成对的 1 改为 0. 若有奇数个 1, 最后会执行  $M(0, 2)$  停机, 输出  $\bar{0}$ . 若有偶数个 1, 最后会执行  $M(0, 1), M(0, 3)$  停机, 输出  $\bar{1}$ .

## 习题 5.17

首先 0,1 显然都不是一个合法的机器编码, 假定输入至少为 2.

此题的关键在于给定机器  $M$  的编码  $\#M$  如何确定其行数. 解法为:

设  $M$  有  $k+1$  行, 那么显然有  $k < \#M$ . 从而设  $k = \max z \leq \#M, [p_z | \#M]$ ,

一旦  $k$  确定, 那么对于各行的解码过程都是可计算的. 一旦出现不合法的行则解码失败, 全部成功解码则说明输入是一个合法的机器编码, 属于  $S$ .

反过来由于编码和解码是一对逆过程, 那么解码失败的输入一定不属于  $S$ .

如果我们承认 CT, 那么我们已经给出了一个能行的方法判定  $S$ , 从而  $S$  是 Turing-可计算的.

若持有更保守的观点, 那可以通过证明  $S$  为递归集来证明其 Turing-可计算. 过程繁杂从略.

## 习题 5.18

由 Taylor 公式

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

(1) 令  $f(n) = \lfloor e \cdot n! \rfloor$ , 我们往证  $f \in \mathcal{EF}$ :

$f(0) = f(1) = 2$ , 当  $n \geq 2$

$$f(n) = \left( \sum_{i=0}^n \frac{1}{i!} \right) n! + \frac{e^\theta}{n+1}$$

注意到当  $n \geq 2$  时  $0 < \frac{e^\theta}{n+1} < 1$ , 从而

$$f(n) = \sum_{i=0}^n \binom{n}{i} (n-i)!$$

于是  $f \in \mathcal{EF}$ .

(2) 令  $h(n) = \lfloor e \cdot n \rfloor$  则  $h(0) = 0$ ,  $h(n) = \left\lfloor \frac{e \cdot n!}{(n-1)!} \right\rfloor = \left\lfloor \frac{f(n)}{(n-1)!} \right\rfloor (*)$ ,  $n \geq 1$ .

从而  $h \in \mathcal{EF}$ .

注: 下面给出 (\*) 的证明:

设

$$e \cdot n! = k_1(n-1)! + r_1$$

$$f(n) = k_2(n-1)! + r_2$$

其中  $k_1, k_2 \in \mathbb{N}$ ,  $r_1, r_2 \in [0, (n-1)!]$

由于  $f(n)$  为整数, 我们有  $r_2 \in [0, (n-1)! - 1]$ . 那么

$$r \triangleq e \cdot n! - f(n) = (k_1 - k_2)(n-1)! + (r_1 - r_2) \in [0, 1)$$

又  $r_1 - r_2 \in [1 - (n-1)!, (n-1)!]$ , 从而  $k_1 = k_2$ .

(3)

$$g(1) = 2$$

$$g(n) = h(10^n) \div h(10^{n-1}) \cdot 10, \quad n > 1$$

从而  $g \in \mathcal{EF}$ , 于是  $g$  可计算。



注意到余项的上界  $\frac{e}{(n+1)!}$  严格单调递减且收敛速度超过指数函数, 此题也可以设计数值算法来求  $e$  的前  $n$  位有效数字, 进而通过 CT 说明  $g$  可计算.