吉林大学

可计算性与计算复杂性

Calculability & Complexity

第二章 可计算函数

1、原语言

本书所有变元为非负整数

五条基本指令：

① $X=X+1$
② $X=X-1$ （X=0，结果仍为 0）
③ TO A IF $X\neq 0$
④ TO A
⑤ $Y=X$

约定：

① 输入变元：$X_0,X_1,\cdots\cdots$
② 临时变元：$Z_0,Z_1,\cdots\cdots$
③ 输出变元：Y
④ 初始时，一切变元为 0（输入变元除外）
⑤ 转向无定义标号或执行了最后一条指令时，停机

宏指令：

$Y=X_1+X_2$

```
     Y= X1
[B]  TO A IF X2≠0
     TO E
[A]  X2=X2-1
     Y=Y＋1
     TO B
```

$Y=X_1\cdot X_2$

```
[B]  TO A IF X2≠0
     TO E
[A]  X2=X2-1
     Y=Y＋X1
     TO B
```

2、可计算函数

部分可计算函数：若有一程序 P，使得 $\varphi_P(X_1,X_2,\ldots,X_n)=f(X_1,X_2,\ldots,X_n)$，则称 $f(X_1,X_2,\ldots,X_n)$为部分可计算函数。

这里"="表示：或者两边都无定义，或者两边都有定义并且值相同。

可计算函数：若 $f(X_1,X_2,\ldots,X_n)$是部分可计算的而且是全函数，则称 $f(X_1,X_2,\ldots,X_n)$为可计算函数。

两个常用函数

$$X_1 \ominus X_2 = \begin{cases} X_1-X_2, & X_1\geqslant X_2 \\ \text{无定义}, & X_1<X_2 \end{cases}$$

$$X_1 \dot{-} X_2 = \begin{cases} X_1-X_2, & X_1\geqslant X_2 \\ 0, & X_1<X_2 \end{cases}$$

习 题

用原语言证明下列函数是可计算函数

（显然均为全函数，故只需证明存在 n 元程序 P，使得 $\varphi_P(X_1,X_2,\ldots,X_n)=f(X_1,X_2,\ldots,X_n)$即可）

（1） $f(X_1,X_2)=X_1^{X_2}$

```
     Y=Y+1
[B]  TO A IF X2≠0
     TO E
[A]  X2=X2-1
     Y=Y・X1     //宏指令
     TO B
```

（2）$f(X_1,X_2)=\min\{X_1,X_2\}$

```
[C]  TO A IF X1≠0
     TO E
[A]  TO B IF X2≠0
     TO E
[B]  X1=X1-1
     X2=X2-1
     Y=Y+1
```

（3）$f(X)=\alpha(X \dot{-} 3)$

X=X-1

X=X-1

X=X-1

TO E IF X≠0

Y=Y+1

（4）$f(X)=X_1 \dot{-} X_2$

[C] TO A IF $X_2 \neq 0$

$Y=X_1$

TO E

[A] TO B IF $X_1 \neq 0$

$Y=X_1$

TO E

[B] $X_1=X_1-1$

$X_2=X_2-1$

TO C

（5）$f(X)=[\sqrt{X}]$（[ ]为向下取整）

// $t^2 \leqslant X < (t+1)^2$

[B] $Z_1=Z_0$

$Z_1=Z_1 \cdot Z_1$ //宏指令

$Z_1=Z_1-1$

$Z_1=X \dot{-} Z_1$ //宏指令

TO A IF $Z_1 \neq 0$

$Y=Z_0-1$

TO E

[A] $Z_0=Z_0+1$

TO B

（6）$f(X)=[\log_2(X+1)]$

// $2^t \leqslant X < 2^{t+1}$

X=X+1

[B] $Z_1=Z_0$

$Z_1=2^{Z_1}$ //宏指令

$Z_1=Z_1-1$

$Z_1=X \dot{-} Z_1$ //宏指令

TO A IF $Z_1 \neq 0$

$Y=Z_0-1$

TO E

[A] $Z_0=Z_0+1$

TO B

第三章 递归函数

1、算子

复合算子：设一个函数 $y=f(z_1,z_2,\ldots,z_m)$，和一组函数

$z_1=g_1(x_1,x_2,\ldots,x_n)$ $z_2=g_2(x_1,x_2,\ldots,x_n)$ … $z_m=g_m(x_1,x_2,\ldots,x_n)$

称 $y=f(z_1,z_2,\ldots,z_m)=f(g_1(x_1,x_2,\ldots,x_n), g_2(x_1,x_2,\ldots,x_n),\ldots, g_m(x_1,x_2,\ldots,x_n))$

是复合算子作用于函数 f 和 $g_1,g_2,\ldots,g_m$ 的结果。

递归算子：设 $m(x_1,x_2,\ldots,x_n)$ 和 $\phi(x_1,x_2,\ldots,x_{n+2})$ 是全函数，定义

$$\begin{cases} h(x_1,x_2,\ldots,x_n,0)=m(x_1,x_2,\ldots,x_n) \\ h(x_1,x_2,\ldots,x_n,t+1)=\phi(x_1,x_2,\ldots,x_n,h(x_1,x_2,\ldots,x_n,t),t) \end{cases}$$

称 h 是递归算子作用于函数 m 和 ϕ 的结果。

（h 是全函数）

取极小算子：设 $f(x_1,x_2,\ldots,x_n,z)$ 是全函数，定义 $h(x_1,x_2,\ldots,x_n)=\min\{z|f(x_1,x_2,\ldots,x_n,z)=0\}$

称 h 是取极小算子 z 作用于函数 f 的结果。

（h 不一定是全函数，若 f 为正则函数，则 h 为全函数）

2、原始递归函数

原始递归函数：由初始函数 $S(x)=x+1$、$n(x)=0$、$U_i^n(x_1,x_2,\ldots,x_n)=x_i$ $(1\leqslant i\leqslant n)$ 出发，

只用复合和递归算子得到的函数称为原始递归函数。

（它们都是全函数）

原始递归函数

$add(x,y)$	$mul(x,y)$
$fac(x)=x!$	$exp(x,y)=x^y$
$\lvert x-y\rvert$	$sub(x,y)=x\dot{-}y$
$p(x)$ $\begin{cases} p(0)=0 \\ p(x+1)=x \end{cases}$	$\alpha(x)=\begin{cases} 1 & ,x=0 \\ 0 & ,x\neq 0 \end{cases}$
$d(x,y)=\begin{cases} 0 & ,x=y \\ 1 & ,x\neq y \end{cases}$	

3、原始递归谓词

特征函数：设谓词 $P(x_1,x_2,\ldots,x_n)$，定义函数 $\delta(x_1,x_2,\ldots,x_n)=\begin{cases} 0 & ，当 P(x_1,x_2,\ldots,x_n) \\ 1 & ，当\sim P(x_1,x_2,\ldots,x_n) \end{cases}$

称 δ 为谓词 P 的特征函数。

$$\delta(x_1,x_2,\ldots,x_n)=\begin{cases} 0 & ，当(x_1,x_2,\ldots,x_n)\in S \\ 1 & ，当(x_1,x_2,\ldots,x_n)\notin S \end{cases}$$

，称 δ 为集合 S 的特征函数。

原始递归谓词（集合）：谓词 P（集合 S）的特征函数是原始递归函数。

可计算谓词（集合）：谓词 P（集合 S）的特征函数是可计算的。

谓词的受囿全称量词：$(\forall t)_{\le y} P(t, x_1, x_2, ..., x_n) \Leftrightarrow \prod_{t=0}^{y} P(t, x_1, x_2, ..., x_n) = 1$

谓词的受囿存在量词：$(\exists t)_{\le y} P(t, x_1, x_2, ..., x_n) \Leftrightarrow \sum_{t=0}^{y} P(t, x_1, x_2, ..., x_n) \ne 0$

定理：$f(x_1,x_2,...,x_n,y)$是原始递归函数，则 $\sum_{t=0/1}^{y} f(x_1, x_2, ..., t)$ 和 $\prod_{t=0/1}^{y} f(x_1, x_2, ..., t)$ 是原始递归函数。

定理：P、Q 是原始递归谓词，则～P 、$P \vee Q$ 、$P \wedge Q$ 是原始递归谓词。

定理：R、S 是原始递归集合，则 $\overline{R}$ 、$R \cup S$ 、$R \cap S$ 是原始递归集合。

定理：P 是原始递归谓词，g 和 h 是原始递归函数，则

$$f(x_1,x_2,...,x_n)=\begin{cases} g(x_1,x_2,...,x_n) & \text{，当 } P(x_1,x_2,...,x_n) \\ h(x_1,x_2,...,x_n) & \text{，当～} P(x_1,x_2,...,x_n) \end{cases}$$

是原始递归函数。

定理：$P(x_1,x_2,...,x_n)$是原始递归谓词，$h_1,h_2,\ldots,h_n$ 是原始递归函数，则 $P(h_1,h_2,...,h_n)$是原始递归谓词。

定理：f 和 g 是原始递归函数，则 f=g 是原始递归谓词。

定理：若 $P(t,x_1,x_2,...,x_n)$是原始递归谓词，则

$(\forall t)_{\le y} P(t,x_1,x_2,...,x_n)$ 和 $(\exists t)_{\le y} P(t,x_1,x_2,...,x_n)$ 是原始递归谓词。

4、受囿取极小

受囿取极小：设 $P(t,x_1,x_2,...,x_n)$是一个谓词，定义函数

$$\min_{t \le y} P(t, x_1, x_2, ..., x_n) = \begin{cases} \min\{t \mid P(t,x_1,x_2,...,x_n)=1\} & \text{，若 } (\exists t)_{\le y} P(t,x_1,x_2,...,x_n) \\ 0 & \text{，否则} \end{cases}$$

称 $\min\limits_{t \le y}$ 为受囿取极小。

定理：若 $P(t,x_1,x_2,...,x_n)$是原始递归谓词，则函数 $f(y,x_1,x_2,...,x_n)=\min\limits_{t \le y} P(t, x_1, x_2, ..., x_n)$ 是原始递归函数。

原始递归谓词

x=y	x＞y	x≤y
y|x	GN(x)　在 x 因式分解中没有 0 指数	Prim(x)　x 是素数

原始递归函数

[x/y]	P_i　第 i 个素数
R(x,y)　x 除以 y 的余数	t(x)　x 因式分解中非 0 指数的个数

$(x)_i$ x 因式分解中 P_i 的指数	Lt(x) x 因式分解中第 Lt(x)个以后指数均为 0
$[a_1,a_2,\ldots,a_n]=P_1^{a_1}\bullet P_2^{a_2}\bullet\ldots\bullet P_n^{a_n}$ 歌德尔数 x*y $x=[a_1,a_2,\cdots,a_n]$ $y=[b_1,b_2,\cdots,b_m]$ $x*y=[a_1,a_2,\cdots,a_n,b1,b_1,b_2,\cdots,b_m]$	#(a, x) x 因式分解中指数为 a 的有几个
配对函数$\langle x,y\rangle=\frac{1}{2}(x+y)(x+y+1)+y$ ， l(z) ，r(z) $l(\langle x,y\rangle)=x$ $r(\langle x,y\rangle)=y$ $\langle l(z),r(z)\rangle=z$ $l(z),r(z)\leqslant z$	

5、递归与可计算性

部分递归函数：由初始函数 S(x)=x+1，n(x)=0，$\bigcup_i^n(x_1,x_2,\ldots,x_n)=x_i$ （1≤i≤n）出发，用复合、递归和取极小算子得到的函数称为部分递归函数。

递归函数：由初始函数 S(x)=x+1，n(x)=0，$\bigcup_i^n(x_1,x_2,\ldots,x_n)=x_i$ （1≤i≤n）出发，用复合、递归和对正则函数取极小算子得到的函数称为递归函数。

原始递归（全函数）⊂递归（全函数）⊂部分递归（部分函数）

可计算 ⇔ 递归
部分可计算 ⇔ 部分递归

习 题

1、证明下列函数是原始递归函数

（1）$f(x,y)=x^{x^{\cdot^{\cdot^{x}}}}\Big\}$ y 个 x

f(x,y)可递归定义如下：

$$\begin{cases} f(x,0)=1 \\ f(x,y+1)=\exp(x,f(x,y)) \end{cases}$$

∵exp(x,y)为原始递归函数，∴f(x,y)为原始递归函数。

（2）$f(x,y)=[\sqrt[y]{x}]$

设$[\sqrt[y]{x}]=t$，则 $t^y\leqslant x<(t+1)^y$

∴ $[\sqrt[y]{x}]=\min\limits_{t\leq x}(t+1)^y>x$

∵x^y 是原始递归函数，x＞y 是原始递归谓词，∴$(t+1)^y>x$ 是原始递归谓词，∴f(x,y)是原始递归函数。

（3）$f(x)=[\log_2 x]$

设$[\log_2 x]$=t，则 $2^t \leqslant x < 2^{t+1}$

∴ $[\log_2 x] = \min\limits_{t\le x} 2^{t+1} > x$

∴f(x)是原始递归函数。

（4）f(x)表示 x 各位数字之和

设 x 为 n 位数，n=$[\log_2 x]+1$

$$f(x)=\sum_{i=0}^{n-1}[R(x,10^{i+1})-R(x,10^{i})]\cdot 10^{-i}$$

∴f(x)是原始递归函数。

（5）设 a_n=f(n), b_n=g(n)都是递增的原始递归函数，将 a_n，b_n（n＝0，1，2，……）混合在一起，再从小到大排列得到函数 c_n＝φ(n)，证明φ(n)是原始递归函数。

φ(n)递归定义如下：

$$\begin{cases} \phi(0)=\min\{a_0,b_0\} \\ \phi(n+1)=\min\limits_{t\le\max\{a_n,b_n\}}\{((\exists i)_{\le n}(t=a_i)\vee(\exists j)_{\le n}(t=b_j))\wedge(t>\phi(n))\} \end{cases}$$

∵ $\min\{a_0,b_0\}=a_0\cdot\alpha(\mathrm{sub}(a_0,b_0))+b_0\cdot\alpha(\mathrm{sub}(b_0,a_0))\cdot \mathrm{d}(a_0,b_0)$

$\max\{a_n,b_n\}=b_n\cdot\alpha(\mathrm{sub}(a_n,b_n))+a_n\cdot\alpha(\mathrm{sub}(b_n,a_n))\cdot \mathrm{d}(a_n,b_n)$

∴min{a_0,b_0}，max{a_n,b_n}是原始递归函数，∴φ(n)是原始递归函数。

（6）将任意两个素数乘积按从小到大排成一个序列，令这个序列通项即第 n 项为 f(n)，证明 f(n)是原始递归函数。

f(n)可递归定义如下：

$$\begin{cases} f(1)=4 \qquad ;2*2 \\ f(n+1)=\min\limits_{t\le P_n^2}\{(\exists i)_{\le n}(\exists j)_{\le n}(t=P_i\cdot P_j)\wedge t>f(n)\} \end{cases}$$

∴f(n)是原始递归函数。

（7）称三边均为整数的直角三角形的斜边为勾股数，将它们从小到大排列，第 n 个勾股数记作 R(n)，证明 R(n)是原始递归函数。

R(n)可递归定义如下：

$$\begin{cases} R(1)=5 \\ R(n+1)=\min\limits_{t\le R^2(n)}\{(\exists i)_{\le R^2(n)}(\exists j)_{\le R^2(n)}(i^2+j^2=t^2)\wedge(t>R(n))\} \end{cases}$$

∴R(n)是原始递归函数。

2、计算 3*2＝？

3=$2^0\times 3^1$＝[0,1]，2＝2^1＝[1]，3*2＝[0,1,1]＝$2^0\times 3^1\times 5^1$＝15

3、用原语言程序（限用五条基本指令）计算谓词“$x_1=x_2$”的特征函数。

$$\begin{cases} 0 \quad , x_1=x_2 \\ \end{cases}$$

δ (x_1,x_2)=

1 ， $x_1 \neq x_2$

```
[C]  TO A IF X1≠0
     TO D IF X2≠0
     TO E
[D]  Y=Y+1
     TO E
[A]  TO B IF X2≠0
     TO D
[B]  X1=X1-1
     X2=X2-1
     TO C
```

4、用原语言程序证明每个原始递归函数都是可计算函数。

A、证明初始函数是可计算的

S（X）=X+1

```
X=X+1
```

n(X)=0

```
X=X+1
Y=X                 Y=0
```

$U_i^n(x_1,x_2,\ldots,x_n) = x_i$

$x_1 = x_1 + 1$

$x_2 = x_2 + 1$

……

$x_{i-1} = x_{i-1} + 1$

$Y = x_i$

$x_{i+1} = x_{i+1}$

……

$x_n = x_n + 1$

第四章 Post-Turing 程序和 Turing 机

1、P-T 程序

双向无穷带 符号：B 和 1

指令：

$\boxed{\text{RIGHT}}$ $\boxed{\text{LEFT}}$ $\boxed{\text{WRITE 1}}$ $\boxed{\text{WRITE B}}$ $\boxed{\text{TO A IF B}}$ $\boxed{\text{TO A IF 1}}$ （不改变指针位置）

数 x 在带上的表示：$\overline{x}$ （$\overline{0}=1$，$\overline{1}=11$，$\overline{3}=1111$）

$f(x_1,x_2,...,x_n)$的初值在带上的表示：$\overline{x}_1B\overline{x}_2B...B\overline{x}_n$ （2，0，3）＝111B1B1111

结果：带上 1 的个数减 1

$$f(x)=\begin{cases} x \dot{-} 2 & (x>1) \\ x+2 & (x\leqslant 1) \end{cases}$$

f(x)=x-1

```
    RIGHT
    RIGHT
    TO A IF 1
    WRITE 1
    RIGHT
    WRITE 1
    TO E IF 1
[A] LEFT
    WRITE B
    LEFT
    WRITE B   //消去最左边两个 1
```

```
    RIGHT
    TO E IF B
[A] RIGHT
    TO A IF 1
    LEFT
    WRITE B   //消去最后一个 1
```

P-T 部分可计算：若有一个 P-T 程序计算函数 f，则称 f 是 P-T 部分可计算的。

P-T 可计算：若函数 f 是 P-T 部分可计算的而且是全函数，则称 f 是 P-T 可计算的。

广义 P-T 机

字母表：$S_0,S_1,...,S_k$ 约定 $S_0=B,S_1=1$

指令：

$\boxed{\text{RIGHT}}$ $\boxed{\text{LEFT}}$ $\boxed{\text{WRITE } S_i}$ $i=0,1,...,k$

$\boxed{\text{TO A IF } S_i}$ （不改变指针位置）

(部分)可计算
⇔ P-T(部分)可计算
⇔ 广义 P-T(部分)可计算

字母表编码：$S_0\to1$，$S_1\to2$，…，$S_k\to k+1$

带上符号串编码：$S_2S_3S_0S_1S_2\to[3,4,1,2,3]=2^3\cdot3^4\cdot5^1\cdot7^2\cdot11^3$

TAPE(x)=[2,2,...,2]=$(\prod_{i=1}^{x+1}P_i)^2$，$\overline{x}$ 的歌德尔数编码。

$\text{TAPE}^n(x_1,x_2,...,x_n)=\text{TAPE}(x_1)*2*\text{TAPE}(x_2)*2*...*2*\text{TAPE}(x_n)$，$\overline{x}_1B\overline{x}_2B...B\overline{x}_n$ 的歌德尔数编码。

2、Turing 机

字母表Σ={$S_0,S_1,...,S_k$} 状态集 Q={q_1(初始),q_2,...,q_n}

四元组：$q_i\ S_j\ S_k\ q_l$　　当前状态 q_i，当前指针指向 S_j，把 S_k 写入当前格，转入状态 q_l

　　　　$q_i\ S_j\ L\ q_l$　　当前状态 q_i，当前指针指向 S_j，指针左移一格，转入状态 q_l

　　　　$q_i\ S_j\ R\ q_l$　　当前状态 q_i，当前指针指向 S_j，指针右移一格，转入状态 q_l

$f(x)=2x$

字母表 $\Sigma=\{1,B,a,b\}$

状态集 $Q=\{q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q\}$

x＝3 为例：

q_1 1 a q_2

a	1	1	1	B	B	B	B

a	a	1	1	b	B	B	B

a	a	a	1	b	b	B	B

a	a	a	a	b	b	b	B

q_2 a R q_2

q_2 1 R q_2

q_2 B b q_3

a	1	1	1	b	B	B	B

a	a	1	1	b	b	B	B

a	a	a	1	b	b	b	B

a	a	a	a	b	b	b	b

q_2 b R q_2

q_3 b L q_3

q_3 1 L q_4

q_3 a L q_5

q_4 1 L q_4

q_4 a R q_1

q_5 a L q_5

q_5 B R q_6

q_6 a 1 q_7

1	1	1	1	1	1	1	1

q_6 b 1 q_7

q_6 B L q_8

q_7 1 R q_6

q_8 1 B q

1	1	1	1	1	1	1	B

算法：原来的 1 用 a 表示，复制的 1 用 b 表示
　　　最后统一改为 1 再消去一个 1

q_1：把 1 改写为 a

q_2：改写 a 后，右移至尾，写 b

q_3：写 b 后，左移至遇到 1 或 a

q_4：遇到 1 后，左移至遇到 a

q_5：遇到 a 后，左移至头

q_6：把 a、b 改写为 1

q_7：改写 1 后，右移

q_8：消去一个 1

q：终止

五元组：$q_i\ S_j\ S_k\ L\ q_l$　当前状态 q_i，当前指针指向 S_j，把 S_k 写入当前格，左移指针一格，转入状态 q_l

　　　　$q_i\ S_j\ S_k\ R\ q_l$　当前状态 q_i，当前指针指向 S_j，把 S_k 写入当前格，右移指针一格，转入状态 q_l

Turing 部分可计算：若有一个 Turing 机计算函数 f，则称 f 为 Turing 部分可计算的。

Turing 可计算：若函数 f 是 Turing 部分可计算的而且是全函数，则称 f 为 Turing 可计算的。

相同字母表，四元组 Turing 可计算 ⇔ 五元组 Turing 可计算 ⇔ P-T 可计算

3、P-T 程序编码

指令	编码

RIGHT	1
LEFT	2
WRITE 1	3
WRITE B	4
TO A_i IF 1	2i＋4
TO A_i IF B	2i＋3

指令	标号代码	无标号指令代码	指令代码
[A_2] RIGHT	2	1	<2,1>=7
TO A_1 IF B	0	5	<0,5>=20
TO A_2 IF 1	0	8	<0,8>=44
[A_1] WRITE 1	1	3	<1,3>=13
RIGHT	0	1	<0,1>=2
WRITE 1	0	3	<0,3>=9

程序的编码＝[7,20,44,13,2,9]=$2^7 3^{20} 5^{44} 7^{13} 11^2 13^9$

任意 P-T 程序对应一正整数，任意正整数不能对应 P-T 程序。

4、一些定理

通用程序：用原语言写的一个特殊程序，它有两个输入变元 Z、X。P 是编码为 Z 的 P-T 程序，P 对带上的 $\bar{X}$ 停机，当且仅当通用程序对 Z 和 X 停机，同时计算结果一致。

$\Phi^{(n)}(Z, X_1, X_2, ..., X_n)$：

若 Z 是某 P-T 程序 P 的歌德尔数，并且程序 P 计算部分函数 $g(X_1,X_2,...,X_n)$，则

$$\Phi^{(n)}(Z, X_1, X_2, ..., X_n) = g(X_1,X_2,...,X_n);$$

若 Z 不是任何 P-T 程序的歌德尔数，或者 g 对 $X_1,X_2,...,X_n$ 无定义，则

$\Phi^{(n)}(Z, X_1, X_2, ..., X_n)$ 对 Z, $X_1,X_2,...,X_n$ 无定义。

枚举定理：对每个 n，部分函数 $\Phi^{(n)}(Z, X_1, X_2, ..., X_n)$ 是部分可计算函数。

对任何部分可计算函数 $g(X_1,..,X_n)$，都有 Z 使其等于 $\Phi^{(n)}(Z, X_1, X_2, ..., X_n)$。

计步谓词：$STP^{(n)}(Z, X_1, X_2, ..., X_n) \Leftrightarrow$ PROG(Z)，并且编码为 Z 的 P-T 程序对于输入 $X_1,X_2,...,X_n$ 在≤M 步内停机。 PROG(x)：x 是某 P-T 程序的歌德尔数，取 1；否则取 0。

计步定理：对于任意 n，谓词 $STP^{(n)}(Z, X_1, X_2, ..., X_n)$ 是可计算的。

迭代定理：对于一切 n，有原始递归函数 $S^{(n)}(Z, X_1, X_2, ..., X_n)$ 使得对一切 m，

$$\Phi^{(n+m)}(Z, X_1, X_2, ..., X_n, V_1, ..., V_m) = \Phi^{(m)}(S^{(n)}(Z, X_1, X_2, ..., X_n), V_1, ..., V_m)$$

习 题

用四元组 Turing 机计算（先写算法）

（1）f(x)=x+[x/2]

字母表Σ={1,B,a,b}　　状态集 Q={$q_1,q_2,q_3,q_4,q_5,q_6,q_7,q$}

算法：① 消去一个 1

② 从第一个 1 开始，第奇数个 1 不变，第偶数个 1 改写为 a，同时在尾部写上一个 b，直到扫描一遍

③ 将 a 和 b 改写为 1，再加上一个 1

q_1 1 B q_1
q_1 B R q_2　　消去一个 1

q_2 1 R q_3　　第奇数个 1 不变

q_3 1 a q_3　　第偶数个 1 改写为 a

q_3 a R q_4
q_4 1 R q_4
q_4 b R q_4
指针右移过程

q_4 B b q_5　　尾部写上一个 b

q_5 b L q_5
q_5 1 L q_5
q_5 a R q_2
指针左移过程

q_2 b L q_6
q_3 b L q_6
q_6 1 L q_6
q_6 a L q_6
q_6 B R q_7
扫描完毕，指针移到最左端

q_7 1 R q_7
q_7 a 1 q_7
q_7 b 1 q_7
q_7 B 1 q
将 a 和 b 改写为 1，再加上一个 1

q_2 B 1 q　　x=0 情况
q_3 B 1 q　　x=1 情况

x=0 变化情况：
B 1 B
B B B
B B 1

x=1 变化情况：
B 1 1 B
B B 1 B
B B 1 1 B

x=4 变化情况：
B 1 1 1 1 1 B
B B 1 1 1 1 B
B B 1 a 1 a b b B
B B 1 1 1 1 1 1 1 B

x=5 变化情况：
B 1 1 1 1 1 1 B
B B 1 1 1 1 1 B
B B 1 a 1 a 1 b b B
B B 1 1 1 1 1 1 1 1 B

（2）$f(x,y)=x\ominus y$

字母表Σ={1,B,b}　　　状态集 Q={q_0,q_0’,q_0’’,q_1,q_2,q_2’,q_3,q_4,q_5,q_6,q}

算法：① x 段加 1，x 与 y 之间的 B 改为 b

② 把 x 段最左端 1 改成 B，同时把 y 最左端 1 改成 b，转③

③ 重复②，直到出现下面的情况

如果 y 中的 1 全被改成 b，则 x≥y，将全部 b 改为 B，x 段剩下的 1 为所求结果

如果 x 中的 1 全被改成 B，则 x＜y，不停机

q_0 1 L q_0’

q_0’ B 1 q_0’’　x 段加 1

q_0’’ 1 R q_0’’

q_0’’ B b q_1　x 与 y 之间的 B 改为 b

$\left.\begin{array}{l} q_1\ b\ L\ q_1 \\ q_1\ 1\ L\ q_1 \\ q_1\ B\ R\ q_2 \end{array}\right\}$ 指针左移过程

q_2 1 B q_3　x 段最左端 1 改成 B

$\left.\begin{array}{l} q_2\ b\ R\ q_2' \\ q_2'\ b\ L\ q_2 \end{array}\right\}$ x 中的 1 全被改成 B，不停机

$\left.\begin{array}{l} q_3\ B\ R\ q_3 \\ q_3\ 1\ R\ q_3 \\ q_3\ b\ R\ q_4 \\ q_4\ b\ R\ q_4 \end{array}\right\}$ 指针右移过程

q_4 1 b q_1　把 y 最左端 1 改成 b

q_4 B L q_5　y 中的 1 全被改成 b

$\left.\begin{array}{l} q_5\ b\ B\ q_6 \\ q_6\ B\ L\ q_5 \\ q_5\ 1\ R\ q \end{array}\right\}$ 将全部 b 改为 B，停机

（3）f(x)= [x/y]

字母表∑={1,B,a,b,c,d}　　状态集 Q={q_0,q_0’,q_0’’, q_0^*,q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9,q_{10},q_{11},q_{12},q}

算法：① 带上 y 在左，x 在右，去掉 y 段第一个 1 和 x 段最后一个 1，并把 x、y 分界改为 b

② 在最左端写一个 c，把 y 段的 1 逐一改为 a，同时把 x 段的 1 逐一改为 d

如果 y 扫描完毕，把 a 全部改为 1，转②

如果 x 扫描完毕，转③

③ 把带上 a、1、b、d 改为 B，c 改为 1

q_0 1 B q_0’　去掉 y 段第一个 1

q_0' B R q_0''
q_0'' 1 R q_1
q_0'' B L q_0^*
q_0^* B R q_0''　y=0，不停机
q_1 1 R q_1
q_1 B b q_2　x、y 分界改为 b
q_2 b R q_2
q_2 1 R q_2
q_2 B L q_3
q_3 1 B q_4　去掉 x 段最后一个 1
q_4 B L q_5
q_5 1 L q_5
q_5 b L q_5
q_5 B c q_6　在最左端写一个 c
q_6 c R q_6

q_6 1 a q_7　把 y 段的 1 改为 a
q_7 a R q_7
q_7 1 R q_7
q_7 b R q_8　指针右移过程
q_8 d R q_8　把 y 段的 1 逐一改为 a，同时把 x 段的 1 逐一改为 d
q_8 1 d q_9　把 x 段的 1 改为 d
q_9 d L q_9
q_9 b L q_9
q_9 1 L q_9　指针左移过程
q_9 a R q_6

q_6 b L q_5
q_5 a 1 q_5　y 扫描完毕，把 a 全部改为 1
q_5 c L q_5

q_8 B L q_{10}
q_{10} d B q_{11}
q_{10} b B q_{11}
q_{10} 1 B q_{11}
q_{10} a B q_{11}　x 扫描完毕，把带上 a、1、b、d 改为 B，c 改为 1
q_{11} B L q_{10}
q_{10} c 1 q_{12}
q_{12} 1 L q_{10}
q_{10} B R q

（4）$f(x)=[\log_2 x]$

字母表 $\Sigma=\{1,B,a,b,c\}$　　状态集 $Q=\{q_1,q_2,q_3,q_4,q_4',q_5,q_6,q_7,q\}$

算法：十进制数 x 的二进制位数为$[\log_2 x]+1$

用 a、b 分别表示二进制的 0 和 1，在左端从 0 开始，做加 1 的二进制加法，高位在右，低位在左
每做一次二进制加法，就用一个 c 替换一个 1
当所有 1 被 c 替换时，带上 a 和 b 的总数就是$[\log_2 x]+1$，最后把 c 改为 B，把 a 和 b 改为 1

q_1 1 a q_4　　写 a，开始右移

q_4 a R q_4
q_4 b R q_4
q_4 c R q_4　右移至 1 改为 c，开始左移
q_4 1 c q_2

q_4 B L q_4’
q_4’ a R q_4　　x=0，不停机

q_2 c L q_2
q_2 b L q_2
q_2 a L q_2　左移至最左端
q_2 B R q_3

q_3 a b q_4　　最低位为 0，则改为 1
q_3 b a q_5　　最低位为 1，则向右进位
q_5 a R q_3
q_3 c b q_4

q_4 B L q_6
q_6 c B q_7
q_7 B L q_6
q_6 b 1 q_6　c 改为 B，a 和 b 改为 1
q_6 a 1 q_6
q_6 1 L q_6
q_6 B R q

（5）$f(x)= [\log_3(1+x)]$
字母表Σ={1,B,a,b,c,d }　　　状态集 Q={q_0,q_1,q_2,q_3,q_3’,q_4,q_5,q_6,q_7,q_8,q_9,q}
算法：类似（4），用 a、b、c 分别表示三进制的 0、1 和 2

q_0 1 L q_1
q_1 B 1 q_2　加 1，写 a，开始右移
q_2 1 a q_3

q_3 a R q_3
q_3 b R q_3
q_3 c R q_3 右移至 1 改为 d，开始左移
q_3 d R q_3
q_3 1 d q_4

q_3 B L q_3’
q_3’ a R q_3 x=0，不停机

q_4 d L q_4
q_4 c L q_4
q_4 b L q_4 左移至最左端
q_4 a L q_4
q_4 B R q_5

q_5 a b q_3 最低位为 0，则改为 1
q_5 b c q_3 最低位为 1，则改为 2
q_5 c a q_6 最低位为 2，则向右进位
q_6 a R q_5
q_5 d b q_3

q_3 B L q_8
q_8 d B q_9
q_9 B L q_8
q_8 c 1 q_8 d 改为 B，a、b、c 改为 1
q_8 b 1 q_8
q_8 a 1 q_8
q_8 1 L q_8
q_8 B R q

3、用五元组 Turing 机计算谓词 P(x, y) $\Leftrightarrow$ (3x=2y)的特征函数。

$$\delta(x,y)=\begin{cases} 0 & ,\ x/2=y/3 \\ 1 & ，否则 \end{cases}$$

算法：① 把 x 前的 B 改为 a，y 后的 B 改为 b
② 从 x 和 y 中间的 B 开始扫描，每当 x 减去两个 1，y 减去三个 1，转③
③ 重复②，直到下面情况
如果某次扫描 x 后遇到 a 且扫描 y 后遇到 b，在带上写一个 1；否则，在带上写两个 1

第五章 半可计算性

$f(x_1,x_2,\ldots,x_n)\downarrow$：f 对$(x_1,x_2,\ldots,x_n)$有定义。 $f(x_1,x_2,\ldots,x_n)\uparrow$：f 对$(x_1,x_2,\ldots,x_n)$无定义。

半可计算谓词：对于谓词 $P(x_1,x_2,\ldots,x_n)$，如果存在一个部分可计算函数 $g(x_1,x_2,\ldots,x_n)$，使得 $P(x_1,x_2,\ldots,x_n) \Leftrightarrow g(x_1,x_2,\ldots,x_n)\downarrow$，则称 P 为半可计算谓词。

半可计算集合：对于集合 S，如果存在一个部分可计算函数 $g(x_1,x_2,\ldots,x_n)$，使得 $(x_1,x_2,\ldots,x_n)\in S \Leftrightarrow g(x_1,x_2,\ldots,x_n)\downarrow$，则称 S 为半可计算集合。

$\bar{X} = x_1,x_2,\ldots,x_n$

可判定谓词：对于谓词 $P(\bar{X})$，如果存在一个算法 A，使得对任给 $\bar{X}$ 能在有限步内回答 P(X)是否为真，则称 P 为可判定谓词。

半可判定谓词：对于谓词 $P(\bar{X})$，如果存在一个算法 A，使得对任给 $\bar{X}$ 如果 $P(\bar{X})$为真，则在有限步内回答是，否则不能给出回答，则称 P 为半可判定谓词。

递归集合：对于集合 S，如果存在一个算法 A，使得对任给 $\bar{X}$ 能在有限步内回答是否 $\bar{X}\in S$，则称 S 为递归集合。

递归可枚举集合：对于集合 S，如果存在一个算法 A，使得对任给 $\bar{X}$ 如果 $\bar{X}\in S$，则在有限步内回答是，否则不能给出回答，则称 S 为递归可枚举集合。

谓词的可计算性 $\Leftrightarrow$ 可判定性，半可计算性 $\Leftrightarrow$ 半可判定性

集合的可计算性 $\Leftrightarrow$ 递归性，半可计算性 $\Leftrightarrow$ 递归可枚举性

定理：若谓词 P 和 Q 是半可计算的，则 $P\wedge Q$、$P\vee Q$ 是半可计算的。

若集合 S_1 和 S_2 是半可计算的，则 $S_1\bigcap S_2$、$S_1\bigcup S_2$ 是半可计算的。

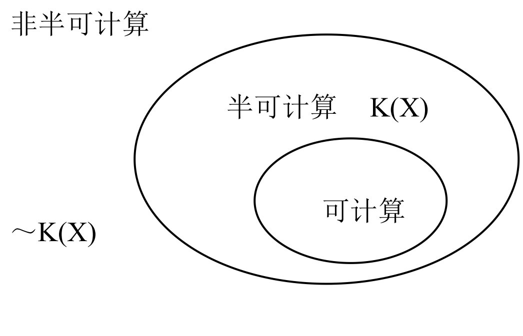
若谓词 $H(v,\vec{X})$是半可计算的，则 $(\exists v)H(v,\vec{X})$、$(\forall v)_{\le z}H(v,\vec{X})$ 是半可计算的。

范式定理：$H(\vec{X})$是半可计算谓词，当且仅当存在可计算谓词 $C(y,\vec{X})$，使 $H(\vec{X}) \Leftrightarrow (\exists y)C(y,\vec{X})$。

Post 定理：$P(\vec{X})$是可计算的，当且仅当 $P(\vec{X})$和～$P(\vec{X})$半可计算的。

一元谓词 K(X)：$K(X) \Leftrightarrow \Phi^{(1)}(X,X)\downarrow$。表示：K(X)为真，当且仅当歌德尔数为 X 的 P-T 程序对输入 X 停机。

定理：K(X)是半可计算的，但不是可计算的。 推论：～K(X)不是半可计算的。



图形定理：函数 $f(x_1,x_2,...,x_n)$部分可计算$\Leftrightarrow$谓词 $V=f(x_1,x_2,...,x_n)$是半可计算的。

集合 W_Z：使得歌德尔数为 Z 的 P-T 程序停机的所有那些 x 的集合。

习 题

1. 举出一种运算，可计算谓词对这个运算是不封闭的（说明理由）

半可计算谓词对这个运算是封闭的（不必说明理由）。

①存在量词运算∃，对可计算谓词不是封闭的。

设 $C(y,\vec{X})$为可计算谓词，对于$(\exists y)C(y,\vec{X})$逐次对 y=0,1,2,...验证 $C(y,\vec{X})$是否为真，

若存在 y，则过程可终止，否则不终止。故$(\exists y)C(y,\vec{X})$是半可计算的。

②存在量词运算∃，对半可计算谓词是封闭的。

有定理：若谓词 $H(v,\vec{X})$是半可计算的，则 $(\exists v)H(v,\vec{X})$ 是半可计算的。

2. 举出一个不可计算的全函数。

$$f(X)=\begin{cases} 1 & ，若\ K(X) \\ 0 & ，若\sim K(X) \end{cases}$$

其中一元谓词 $K(X)\Leftrightarrow \Phi^{(1)}(X,X)\downarrow$

第六章 半图厄系统

1、半图厄系统

字母表 D={a,b}

产生式集 $\left.\begin{array}{l}ab\to aaa\\ ba\to bba\\ \ldots\ldots\end{array}\right\}$ 半图厄处理σ

aba⇒aaaa　　一步推导

aba⇒abba⇒aaaba 多步推导，记 $aba\overset{*}{\Rightarrow}aaaba$、$aba\overset{*}{\underset{\sigma}{\Rightarrow}}aaaba$

σ是半图厄处理，若 a→b 在σ中，则 b→a 在σ中，称σ为图厄处理。

（半）图厄系统：（半）图厄处理加上一个公理或初始字，记π=(P,A)。

公理 A 可推出的符号 W 称为定理。

2、半图厄系统与图灵机、半可计算集

Post 字：图灵机带上字符为 $S_1S_2S_3S_4$，指针指向 S_3 且状态为 q，则称 $hS_1S_2qS_3S_4h$ 为波斯特字。

Σ(M)：半图厄处理，Ω(M)：Σ(M)的逆

M 对 X 停机，当且仅当 $hq_1\bar{X}h\overset{*}{\underset{\Sigma(M)}{\Rightarrow}}hq'h$。　　M 对 X 停机，当且仅当 $hq'h\overset{*}{\underset{\Omega(M)}{\Rightarrow}}hq_1\bar{X}h$。

Φ(M)：使 M 停机的所有 X 的集合。

T(π)：π=(P,A)所有定理的集合。

N(π)={X| $\bar{X}\in$ T(π)}半图厄系统π产生 $\bar{X}$ 的整数 X 的集合。

$\boxed{\text{定理}}$：对任意图灵机 M，都有一个半图厄系统π，使得 X∈Φ(M) ⇔ $\bar{X}\in$ T(π)。

$\boxed{\text{定理}}$：对任意半可计算集 S，都有一个半图厄系统π，使得 S= N(π)。

3、不可判定问题

(1)Turing 机停机问题 I：对任给的 Turing 机 M 和任给的输入 X，判定 Turing 机 M 是否停机。

(2)Turing 机停机问题 II：对已给的 Turing 机 M 和任给的输入 X，判定 Turing 机 M 是否停机。

(3)半图厄系统的字问题：对已给的半图厄系统σ，判定对任给的两个字 ω_1 和 ω_2，是否有 $\omega_1\overset{*}{\underset{\sigma}{\Rightarrow}}\omega_2$。

(4)半图厄系统的判定问题：对已给的半图厄系统σ，判定对任给的一个字ω是不是σ的定理，

即是否有 $A\overset{*}{\underset{\sigma}{\Rightarrow}}\omega$。

(5)Post 对应问题：对给定的对偶序列 $(\omega_1,\varpi_1),(\omega_2,\varpi_2),\ldots,(\omega_n,\varpi_n)$，判定是否存在序列 $i_1,i_2,\ldots,i_n$

使 $\omega_{i_1}\omega_{i_2}\ldots\omega_{i_n}=\varpi_{i_1}\varpi_{i_2}\ldots\varpi_{i_n}$，其中 $1\leqslant i_1,i_2,\ldots,i_n\leqslant n$。

习 题

1、给出半图厄系统π（先写算法）

(1) $N(\pi)=\{x\,|\,(\exists n)(x=2^n)\}$

设公理为 A，半图厄处理 P 为：

A→aA　　　　用 n 次

A→1h　　　　aa...a1h（n 个 a）

a1→11a　　　11...1aa...ah（2^n个 1,n 个 a）

ah→h　　　　消去 a

h→1　　　　11...1（2^n+1 个 1）

(2) $N(\pi)=\{x|(\exists n)(x=2^{3^n})\}$

设公理为 A，半图厄处理 P 为：

A→111　　　　n=0

A→aA　　　　用 n 次，n≥1

A→bch　　　　aa...abch（n 个 a）

ab→bbba　　　bb...baa...ach（3^n个 b,n 个 a）

ac→1a

a1→1a　　　　bb...b1aa...ah（3^n个 b,n 个 a）

b1→11b　　　11...1bb...baa...ah（2^{3^n} 个 1,3^n个 b,n 个 a）

ah→h　　　　消去 a

bh→h　　　　消去 b

h→1　　　　11...1（2^{3^n} +1 个 1）

(3) $N(\pi)=\{x|(\exists n)(x=n2^n)\}$

设公理为 Ah，半图厄处理 P 为：

Ah→1　　　　n=0

A→aA1

A→a1　　　　aa...a11...1h（n 个 a，n 个 1）

a1→11a　　　11...1aa...ah（$n2^n$个 1,n 个 a）

ah→h　　　　消去 a

h→1　　　　11...1（$n2^n$+1 个 1）

(4) $N(\pi)=\{x|(\exists n)(x=n2^{[\sqrt{n}]})\}$

令[$\sqrt{n}$]=m，则 m^2≤n＜$(m+1)^2$=m^2+2m+1，类似(3)构造 c...c1...1（m 个 c，n 个 1）

设公理为 hAh，半图厄处理 P 为：

hAh→1　　m=0

$A \to aAb$

$A \to ab$　　　ha...ab...bh（m 个 a，m 个 b，m≥1）

$a \to cd$

$dc \to cd$　　　hc...cd...db...bh（m 个 c，m 个 d，m 个 b）

$db \to b1d$

$d1 \to 1d$

$dh \to h$　　　hc...cb...b1...1h（m 个 c，m 个 b，m^2 个 1）

$hc \to ch$

$hb \to 11h$

$hb \to 1h$

$hb \to h$

$h1 \to 1h$

$hh \to h$　　　c...c1...1h（m 个 c，n 个 1，$m^2 \leqslant n \leqslant m^2+2m$）

$c1 \to 11c$

$ch \to h$

$ch \to 1$

(5) $N(\pi)=\{x\,|\,(\exists n)(\exists m)(x=nm)\}$

设公理为 hAh，半图厄处理 P 为：

$A \to 1$　　　n=0 或 m=0

$A \to ab$

$ha \to haa$　　　n 次，ha...abh(n 个 a，1 个 b)

$bh \to bbh$　　　m 次，ha...ab...bh（n 个 a，m 个 b）

$ab \to b1a$

$a1 \to 1a$　　　hb...b1...1a...ah（m 个 b，nm 个 1，n 个 a）

$hb \to h$

$ah \to h$

$h1 \to 1h$

$hh \to 1$

(6) $N(\pi)=\{x|(\exists n)(x=n\cdot 2^{[\frac{n}{4}]})\}$

令$[\frac{n}{4}]$=m，则 4m≤n＜4(m+1)=4m+4，设公理为 hAch，半图厄处理 P 为：

hAch→1　　m=0

A→aAb

A→ab　　ha...ab...bch（m 个 a，m 个 b，m≥1）

ha→ah

hb→1111h

hc→1h

hc→11h

hc→111h

hc→h

hh→h　　a...a1...1h（m 个 a，n 个 1，4m≤n≤4m+3）

a1→11a

ah→h

ah→1

(7) $N(\pi)=\{x|(\exists n)(x=n[\log_2 n])\}$

令$[\log_2 n]$=m，则 2^m≤n＜2^{m+1}=2×2^m，类似(5)构造 c...cb...b（n 个 c，m 个 b）

设公理为 hAh，半图厄处理 P 为：

hAh→1　　m=0

A→abc

ab→abb　　hab...bch（m 个 b）

bc→ccb　　hac...cb...bh（2^m个 c，m 个 b）

acc→cac

acc→ccac

acb→cb　　hc...cb...bh（n 个 c，m 个 b，2^m≤n≤2×2^m-1）

cb→b1c

c1→1c　　hb...b1...1c...ch（m 个 b，nm 个 1，n 个 c）

hb→h

ch→h

h1→1h

hh→1

(8) $N(\pi)=\{x\,|\,(\exists n)(\exists m)(x=n^{m})\}$

设公理为 hA_1A_2ch，半图厄处理 P 为：

$hA_1A_2ch \to 1$ （n=0，m=0 或 1）

$hA_1A_2ch \to 11$ （n=1，m=0 或 1）

$A_1 \to aA_1$

$A_1 \to a$

$A_2 \to bA_2c$

$A_2 \to bc$

$$h\underbrace{a\ldots a}_{m-1}\underbrace{b\ldots b}_{n-1}\underbrace{c\ldots c}_{n}h$$ （n≥2，m≥2）

ab→bda

ac→ca

ah→h

db→bd

$$h\underbrace{a\ldots a}_{m-2}\underbrace{b\ldots b}_{n-1}\underbrace{d\ldots d}_{n-1}\underbrace{c\ldots c}_{n}h$$

dc→ced

de→ed

dh→h

$$h\underbrace{a\ldots a}_{m-2}\underbrace{b\ldots b}_{n-1}\underbrace{c\ldots c}_{n}\underbrace{e\ldots e}_{n(n-1)}h$$

ce→cc

$$h\underbrace{a\ldots a}_{m-2}\underbrace{b\ldots b}_{n-1}\underbrace{c\ldots c}_{n^2}h \;\ldots\ldots\; h\underbrace{b\ldots b}_{n-1}\underbrace{c\ldots c}_{n^m}h$$

hb→h

hc→1h

hh→1

(9) $N(\pi)=\{x\mid(\exists n)(x=[\frac{n}{2}]\cdot 2^{[\sqrt{n}]})\}$

令[$\sqrt{n}$]=m，则 $m^2\leqslant n<(m+1)^2=m^2+2m+1$

$$[\frac{m^2}{2}]\leqslant[\frac{n}{2}]<[\frac{m^2}{2}+m+\frac{1}{2}]$$

设公理为 hAch，半图厄处理 P 为：

A→aAb

A→ab

$$h\underbrace{a\ldots a}_{m}\underbrace{b\ldots b}_{m}ch$$

ab→bda

ad→da

$$h\underbrace{b\ldots b}_{m}\underbrace{d\ldots d}_{m^2}\underbrace{a\ldots a}_{m}ch$$

dd→e

da→a

$$h\underbrace{b\ldots b}_{m}\underbrace{e\ldots e}_{[\frac{m^2}{2}]}\underbrace{a\ldots a}_{m}ch$$

ac→ce

ac→c

ec→ce

生成 0～m 个 e，$h\underbrace{b\ldots b}_{m}c\underbrace{e\ldots e}_{[\frac{n}{2}]}h$

bc→ccb

be→eb

bh→h

$$h\underbrace{c\ldots c}_{2^m}\underbrace{e\ldots e}_{[\frac{n}{2}]}h$$

ce→e1c

c1→1c

he→h

ch→h

h1→1h

hh→1

2、设S是所有能被3整除的二进制数组成的集合，给出半图厄系统σ，使得$T(\sigma)\cap\{0,1\}^*=S$。

设x为十进制数，$\overline{x}$为其二进制表示，并引入三个字符a_0,a_1,a_2

约定：字符串“$\overline{x}a_0$”表示：x被3整除

字符串“$\overline{x}a_1$”表示：x被3除，余1

字符串“$\overline{x}a_2$”表示：x被3除，余2

因为$\overline{x}0=2x$，$\overline{x}1=2x+1$，所以有

若$\overline{x}a_0$，则$\overline{x}0a_0$，$\overline{x}1a_1$

若$\overline{x}a_1$，则$\overline{x}0a_2$，$\overline{x}1a_0$

若$\overline{x}a_2$，则$\overline{x}0a_1$，$\overline{x}1a_2$

设公理为A，半图厄处理P为：

$A\to 0a_0$

$A\to 1a_1$

$a_0\to 0a_0$

$a_0\to 1a_1$

$a_1\to 0a_2$

$a_1\to 1a_0$

$a_2\to 0a_1$

$a_2\to 1a_2$

$0a_0\to 0$ //保证以a_0结尾，被3整除的偶数

$1a_0\to 1$ //保证以a_0结尾，被3整除的奇数

改造1：若S是所有能被3整除，并且是奇(偶)数的二进制数组成的集合，去掉最后两行相应的某行。

改造2：若S是所有能被3整除，但不能被4整除的二进制数组成的集合，最后两位不能为00，最后两行改为

$1a_0\to 1$ //保留奇数

$10a_0\to 10$ //保留以10结尾的

改造3：若要求其公理和所有产生式左端的字长都为1

最后两行改为

$a_0\to 0$ //6n

$a_1\to 1$ //6n+3

第一行改为 $A\to 0$

十进制	二进制
0	0
3	11
6	110
9	1001
12	1100
15	1111
18	10010
21	10101
24	11000

3、L={w|w∈{a,b}*，且 w 的任何字头中的 0≤(a 的数目－b 的数目)≤3}

算法：用 A_i(i=0,1,2,3)表示 A_i 左侧字符中 a 的数目－b 的数目

设公理为 A_0，半图厄处理 P 为：

$A_0 \to aA_1$

$A_1 \to aA_2$

$A_1 \to bA_0$

$A_2 \to aA_3$

$A_2 \to bA_1$

$A_3 \to bA_2$

$A_0 \to a$

$A_1 \to a$

$A_1 \to b$

$A_2 \to a$

$A_2 \to b$

$A_3 \to b$

$M=\{Q,\Sigma,\Gamma,\delta,q_0,B,F\}$

Q：有穷状态集
Σ：不含 B 的输入符号集
Γ：带上允许的有穷符号集，$\Sigma\subset\Gamma$
δ：次动作函数，$Q\times\Gamma\xrightarrow{\delta}Q\times\Gamma\times\{L,R\}$
q_0：初始状态，$q_0\in Q$
B：空白符，$B\in\Gamma$
F：终结状态集，$F\subseteq Q$

语言 L 被 M 接受：L 放在 M 带的左端，M 的带头(即指针)在最左单元上，L 使 M 进入一个终结状态。

类型：
①单向无限单带
②双向无限单带
③多带：双向无限多带
④离线：多带，输入带只读$\varnothing$......\$

$\boxed{\text{定理}}$：L 被双向无限单带 TM 接受$\Leftrightarrow$L 被单向无限单带 TM 接受
L 被多带 TM 接受$\Rightarrow$L 被单带 TM 接受

习 题

1、设语言$L=\{WCW^RCW|W\in\{0,1\}^*\}$，给出接受 L 的多带 TM。
2、给出计算函数 $f(x)=[\log_2\log_2 x]$的多带 TM。
3、用多带 TM 计算 x^x。
4、给出计算谓词 7|x 特征函数的离线 TM。这里输入带上不是$\overline{x}$，而是 x 的二进制表示，并且要求 M 的空间复杂度的阶最小。
（说明：若$\lim\limits_{n\to\infty}\dfrac{S_1(n)}{S_2(n)}=0$，则称 $S_1(n)$的阶小于 $S_2(n)$的阶，如上述极限为常数则阶相等）
5、用多带 TM（允许指针不动），求 x_1, x_2 的最大公约数。(指出结果所在的带）
6、以$\overline{x}$作为输入，在一条带上（要指明哪条）给出结果$\overline{y}$，用多带 TM 计算下面函数 $Y=X-2^{[\log_2 X]}$。
7、给出计算 x_1，x_2的最小公倍数的多带 TM。
8、用离线 TM 计算谓词 Prime(x)的特征函数，并使其空间复杂度最小，并计算出其空间复杂度。
9、用多带 TM 计算级数 1，2，4，7，11，……，$\dfrac{n(n-1)}{2}+1$，……的前 n 项和，并计算其时间复杂度。

空间复杂度：考虑离线 TM，有一条有端记号的只读输入带和 k 条半无穷存贮带。如果对于每个长度为 n 的输入字，M 在任意一条存贮带上扫视，都不超过 S(n)个单元，那么称 M 为 S(n)空间有界图灵机，或称 M 具有空间复杂度 S(n)。称被 M 识别的语言具有空间复杂度 S(n)。

时间复杂度：考虑多带 TM，有 k 条双向无穷存贮带。如果对于每个长度为 n 的输入字，M 在停机前最多做 T(n)个动作，那么称 M 为 T(n)时间有界图灵机，或称 M 具有时间复杂度 T(n)。称被 M 识别的语言具有时间复杂度 T(n)。

复杂性类

DSPACE(S(n))：具有空间复杂度 S(n)的语言族

NSPACE(S(n))：具有非确定空间复杂度 S(n)的语言族

DTIME(T(n))：具有时间复杂度 T(n)的语言族

NTIME(T(n))：具有非确定时间复杂度 T(n)的语言族

空间可构造函数：如果有某个图灵机 M 是 S(n)空间有界的，且对每个 n，都存在某个长度为 n 的输入，对于这个输入，M 实际使用了 S(n)个单元，则称 S(n)为空间可构造函数。
如果对于一切 n，M 对长度为 n 的任何一个输入，都实际使用了恰好 S(n)个单元，则称 S(n) 为完全空间可构造函数。

空间谱系定理：如果 $S_2(n)$是一个完全空间可构造函数，且 $\lim\limits_{n\to\infty}\frac{S_1(n)}{S_2(n)}=0$，$S_1(n)$和 $S_2(n)$都至少是 $\log_2 n$，那么有一个语言，它在 DSPACE($S_2(n)$)中，但不在 DSPACE($S_1(n)$)中。

时间可构造函数：如果存在一个 T(n)时间有界多带图灵机 M，使得对于每个 n，都存在某个输入，对于这个输入，M 实际做了 T(n)个动作，则称 T(n)为时间可构造函数。
如果对于一切长度为 n 的输入，都使用时间 T(n)，则称 T(n)为完全时间可构造函数。

时间谱系定理：如果 $T_2(n)$是一个完全时间可构造函数，且 $\lim\limits_{n\to\infty}\frac{T_1(n)\log T_1(n)}{T_2(n)}=0$，那么有一个语言，它在 DTIME($T_2(n)$)中，但不在 DTIME($T_1(n)$)中。

复杂性量度关系

(a)如果 L 在 DTIME(f(n))中，那么 L 在 DSPACE(f(n))中。

(b)如果 L 在 DSPACE(f(n))中，且 $f(n)\geqslant\log_2 n$，那么有某常数 c(它依赖于 L)，使得 L 在 DTIME($c^{f(n)}$)中。

(c)如果 L 在 NTIME(f(n))中，那么有某常数 c(它依赖于 L)，使得 L 在 DTIME($c^{f(n)}$)中。

Savitch 定理：如果 L 在 NSPACE(S(n))中，那么 L 在 DSPACE($S^2(n)$)中。
这里假定 S(n)是完全空间可构造的，且 $S(n)\geqslant\log_2 n$。

转换引理：设 $S_1(n)$、$S_2(n)$和 f(n)是完全空间可构造的，且 $S_2(n)\geqslant n$，$f(n)\geqslant n$，那么由 NSPACE($S_1(n)$)$\subseteq$NSPACE($S_2(n)$)，可以推出 NSPACE($S_1(f(n))$)$\subseteq$NSPACE($S_2(f(n))$)

习 题

1、证明完全时间可构造

（1）2^n

（2）$n[\log_2 n]$

（3）n^2

（4）$(n+1)^2$

2、证明空间可构造

（1）$[\sqrt{n}]$

（2）$[\sqrt{n}]+3$（用离线图灵机证明，①只有一条存贮带 ②输入带与存贮带都是单道的 ③带上的符号除空格 B 外只有一个符号“1”）

3、证明$\sqrt{x}+1$是完全空间可构造的。

4、证明$DTIME(2^{kn}) \subset DTIME(n2^{kn})$，其中 k 为一常数。

5、证明$DTIME(n) \subset DTIME(n^{1+E})$，其中 E 为大于零的任何实数（不必证明函数的完全可构造性）。

6、利用已知的复杂性度量间的关系，给出 NSPACE 和 NTIME 之间的关系。（给出关系并写明条件，不必证明）