

# *Partial Recursive Functions*

## *Problem solving in Automata, Languages and Automata*

*Ding-Zhu Du and Keri Ko*

*John Wiley and Sons, 2001*

### *4.1 Turing Machine*

### *4.5 Unrestricted Grammar*

### *4.6 Primitive Recursive Function*

### *4.7 Pairing Function and Gödel Numbering*

### *4.8 Partial Recursive Function*

## 4.1 Turing Machine

One-tape deterministic Turing machine  $M = (Q, \Sigma, \Gamma, \delta, s, h, B)$ ,

1.  $Q$  is a finite set of **states**,
2.  $\Sigma$  is a finite set of **input** symbols,
3.  $\Gamma$  is a finite set of **tape** symbols (where  $\Sigma \subseteq \Gamma$ ),
4.  $\delta$  is a **transition function (instructions)**,

$$\delta: (Q - \{h\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

5.  $s \in Q$  is an **initial** state,
6.  $h \in Q$  is a **halting (final)** state, and
7.  $B \in \Gamma$  is a **blank** tape symbol.

### Configuration of TM

$$(\alpha, q, \beta) \in \Gamma^* \times Q \times \Gamma^*.$$

1:  $\beta \in \Gamma$  is the **current tape** symbol.

Blank symbols are assumed to be surrounding the tape string

$$(\alpha, q, \beta) = (B^* \alpha, q, \beta B^*).$$

Let  $(\alpha, q, \beta)$  be a current configuration. Then

i) if  $\delta(q, 1:\beta) = (p, X, R)$  then  $(\alpha, q, \beta) \Rightarrow_M (\alpha X, p, \beta:|\beta|-1)$

note that if  $\beta = \varepsilon$  and  $\delta(q, B) = (p, X, R)$  then  $(\alpha, q, \varepsilon) \Rightarrow_M (\alpha X, p, B)$

ii) if  $\delta(q, 1:\beta) = (p, X, L)$  then  $(\alpha, q, \beta) \Rightarrow_M (|\alpha|-1:\alpha, p, \alpha:1X\beta:|\beta|-1)$

note that if  $\alpha = \varepsilon$  then  $(\varepsilon, q, \beta) \Rightarrow_M (B, p, BX\beta:|\beta|-1)$

We may write  $\Rightarrow$  instead of  $\Rightarrow_M$  when it is clear.

We write reflexive-transitive closure of  $\Rightarrow$  as  $\Rightarrow^*$ .

A string  $x \in \Sigma^*$  is **accepted** by a DTM  $M$ , if

$(\varepsilon, s, x) \Rightarrow_M^* (\alpha, h, \beta)$  for some  $\alpha, \beta \in \Sigma^*$ .

For configuration  $(\alpha, q, \beta)$ , if  $\nexists \delta(q, 1:\beta)$ , then  $M$  is said to be **hang on**.

A string  $x \in \Sigma^*$  is **not accepted** by a DTM  $M$ , if

either  $M$  **hangs on**  $x$  or  $M$  **does not halt (loops forever)** on  $x$ .

$L(M) = \{x \in \Sigma^* \mid (\varepsilon, s, x) \Rightarrow_M^* (\alpha, h, \beta)\}$

*TM as a partial function*

*If  $\exists$  a DTM  $M \ni L = L(M)$ . Then  $L$  is called **Turing-acceptable**.*

*Let  $f: (\Sigma_1^*)^k \rightarrow \Sigma_2^*$  be a partial function.*

*We write  $f(x_1, x_2, \dots, x_k) \downarrow$  to denote  $f$  is **defined** at  $(x_1, x_2, \dots, x_k)$  and  
 $f(x_1, x_2, \dots, x_k) \uparrow$  to denote  $f$  is **not defined** at  $(x_1, x_2, \dots, x_k)$ .*

*A partial function  $f$  is called **Turing-computable**, if  $\exists$  a DTM  $M \ni$ .*

*if  $f(x_1, x_2, \dots, x_k) = y$ , then  $(\varepsilon, s, x_1 B x_2 B \dots B x_k B) \Rightarrow_M^* (\varepsilon, h, y)$ ; and  
 if  $f(x_1, x_2, \dots, x_k) \uparrow$ , then  $(\varepsilon, s, x_1 B x_2 B \dots B x_k B)$   
**does not halt (loops forever).***

$f: (\Sigma_1^*)^k \rightarrow \Sigma_2^*$  is called (**total**) function,

if  $\forall (x_1, x_2, \dots, x_k) \in (\Sigma_1^*)^k, f(x_1, x_2, \dots, x_k) \downarrow$ .

$f$  is called **Turing-decidable**, if  $f$  is a total **Turing-computable** function.

A language  $L$  is **Turing-decidable**, if its characteristics function

$\chi_L(x) = 1$ , if  $x \in L$ ,

0, otherwise.

$\forall x \in \Sigma^*, \exists$  a DTM  $M$  . $\exists$ . if  $x \in L$ , DTM halts; if  $x \notin L$ , it hangs on.

**Turing-decidable**(type 1) is a proper subclass of **Turing-acceptable**(type 0).

### 4.5 Unrestricted grammar

$G = (V, \Sigma, P, S)$  is a unrestricted grammar if

$$\alpha \rightarrow \beta \in P \quad \text{with } \alpha \in (V \cup \Sigma)^+ \text{ and } \beta \in (V \cup \Sigma)^*.$$

*Example 4.15*  $L(G) = \{a^n b^n c^n \mid n \geq 0\}$ .

$$S \rightarrow aSBC \mid \varepsilon, \quad CB \rightarrow BC,$$

$$aB \rightarrow ab, \quad bB \rightarrow bb,$$

$$bC \rightarrow bc, \quad cC \rightarrow cc.$$

$$S \Rightarrow^{n+1} a^n (BC)^n \Rightarrow^{n(n-1)/2} a^n B^n C^n \Rightarrow^{2n} a^n b^n c^n.$$

**Example 4.16** Find a grammar  $G \ni L(G) = \{a^{2^n} \mid n \geq 0\}$ .

$$\begin{array}{ll}
 S \rightarrow [Ra] \mid a, & Ra \rightarrow aaR, \quad \text{doubles } a, \text{ move } R \text{ right} \\
 R] \rightarrow L] \mid L_h, & \text{change } R] \text{ to } L] \text{ or } L_h \text{ (terminate)} \\
 aL \rightarrow La, & [L \rightarrow [R, \quad \text{move } L \text{ left and change it to } [R \\
 aL_h \rightarrow L_h a, & [L_h \rightarrow \varepsilon. \quad \text{move } L_h \text{ left and remove } [.
 \end{array}$$

**basis** i)  $n=0$ ,  $S \Rightarrow_{S \rightarrow a} a = a^{2^0}$ .

ii)  $n = 1$ ,  $S \Rightarrow_{S \rightarrow [Ra]} [Ra] \Rightarrow_{Ra \rightarrow aaR} [aaR] \Rightarrow_{RL_h] \rightarrow L_h} [aaL_h$   
 $\Rightarrow_{aL_h \rightarrow L_h a} [aL_h a] \Rightarrow_{aL_h \rightarrow L_h a} [L_h aa] \Rightarrow_{[L_h \rightarrow \varepsilon} aa = a^{2^1}$ .

**induction**  $S \Rightarrow_{S \rightarrow [Ra]} [Ra] \Rightarrow_{Ra \rightarrow aaR}^* [Ra^{2^k}] \Rightarrow_{Ra \rightarrow aaR}^{2^k} [a^{2^{k+1}} R]$   
 $\Rightarrow_{R] \rightarrow L] [a^{2^{k+1}} L] \Rightarrow_{aL \rightarrow La}^{2^{k+1}} [La^{2^{k+1}}] \Rightarrow_{[L \rightarrow R} [Ra^{2^{k+1}}] \Rightarrow^* a^{2^{k+1}}$ .

*Nonterminal  $R$  cannot move to left until it meets  $]$ ,  $R$  doubles  $a$ 's.*

**Example 4.17** Find a grammar  $G \ni L(G) = \{ww \mid w \in \{a, b\}^*\}$ .

$$\begin{array}{ll}
 S \rightarrow T ], & T \rightarrow a T A \mid b T B \mid [ R, \\
 R A \rightarrow A R, & R B \rightarrow B R, \\
 A R ] \rightarrow L_a ], & B R ] \rightarrow L_b ], \\
 A L_a \rightarrow L_a A, & B L_b \rightarrow L_b B, \\
 [ L_a \rightarrow a [ R, & [ L_b \rightarrow a [ R, \quad [ R ] \rightarrow \varepsilon.
 \end{array}$$

$$\begin{aligned}
 S \Rightarrow T ] &\Rightarrow^4 abaaTAABA] \Rightarrow abaa[RAABA] \Rightarrow^4 abaa[AABAR] \\
 &\Rightarrow abaa[AABL_a] \Rightarrow^3 abaa[L_aAAB] \Rightarrow abaaa[RAAB] \Rightarrow^3 abaaa[AABR] \\
 &\Rightarrow abaaa[AAL_b] \Rightarrow^2 abaaa[L_bAA] \Rightarrow abaaab[RAA] \Rightarrow^2 abaaab[AAR] \\
 &\Rightarrow^* abaaaba[AR] \Rightarrow^* abaaaba[R] \Rightarrow abaaaba.
 \end{aligned}$$

$$S \Rightarrow T ] \Rightarrow^n wT\tilde{w}^R] \Rightarrow w[R\tilde{w}^R] \text{ where } \tilde{w} = w(\text{replace } a, b \text{ to } A, B \text{ resp.})$$

$$[R\tilde{w}^R A] \Rightarrow^n [w^R AR] \Rightarrow_{AR \rightarrow L_a} [w^R L_a] \Rightarrow^{n-1} [L_a w^R] \Rightarrow_{[L_a \rightarrow a[R a]} [R w^R].$$



**Example 4.18** Find a grammar  $G \ni L(G) = \{a^n b^m c^{nm} \mid n, m \geq 0\}$ .

$$\begin{array}{lll}
 S \rightarrow [ A ], & A \rightarrow a A \mid B, & B \rightarrow b B \mid L, \\
 a L \rightarrow L a, & b L \rightarrow L b, & \bar{b} L \rightarrow L \bar{b}, \\
 [ L a \rightarrow a [ R_b, & [ L b \rightarrow b R, & [ L ] \rightarrow \varepsilon, \\
 R_b a \rightarrow a R_b, & R_b b \rightarrow \bar{b} R_c, & R_b ] \rightarrow L ], \\
 R_c b \rightarrow b R_c, & R_c ] \rightarrow L_b ] c, & \\
 b L_b \rightarrow L_b b, & \bar{b} L_b \rightarrow \bar{b} R_b, & \\
 R b \rightarrow b R, & R ] \rightarrow \varepsilon. & 
 \end{array}$$

$$\begin{aligned}
 S &\Rightarrow [ A ] \Rightarrow^n [a^n A] \Rightarrow [a^n B] \Rightarrow^m [a^n b^m B] \Rightarrow [a^n b^m L] \Rightarrow^{n+m} [L a^n b^m] \\
 &a^k [L a^{n-k} b^m] c^{km} \Rightarrow a^{k+1} [R_b a^{n-k-1} b^m] c^{km} \Rightarrow^* a^{k+1} [a^{n-k-1} R_b b b^{m-1}] c^{km} \\
 &\Rightarrow a^{k+1} [a^{n-k-1} \bar{b} R_c b^{m-1}] c^{km} \Rightarrow^* a^{k+1} [a^{n-k-1} \bar{b} b^{m-1} R_c] c^{km} \\
 &\Rightarrow a^{k+1} [a^{n-k-1} \bar{b} b^{m-1} L_b] c^{km} \Rightarrow^* a^{k+1} [a^{n-k-1} \bar{b} L_b b^{m-1}] c^{km+1} \\
 &\Rightarrow a^{k+1} [a^{n-k-1} \bar{b} R_b b^{m-1}] c^{km+1} \Rightarrow^* \dots \Rightarrow^* a^{k+1} [a^{n-k-1} \bar{b}^2 R_b b^{m-2}] c^{km+2} \\
 &\Rightarrow^* a^{k+1} [a^{n-k-1} \bar{b}^m R_b] c^{km+m} \Rightarrow a^{k+1} [a^{n-k-1} \bar{b}^m L] c^{(k+1)m} \\
 &\Rightarrow^* a^{k+1} [L a^{n-k-1} b^m] c^{(k+1)m}.
 \end{aligned}$$

**Theorem 4.19** For any one-tape DTM  $M = (Q, \Sigma, \Gamma, \delta, s)$ ,

$\exists$  a grammar  $G_M = (Q \cup (\Gamma - \Sigma) \cup \{[, ]\}, \Sigma, P, S) .\exists$ .

$\forall$  configurations  $(\alpha, q, X\beta)$  and  $(\alpha', p, Y\beta')$

where  $p, q \in Q, X, Y \in \Gamma, \alpha, \beta, \alpha', \beta' \in \Gamma^*$ ,

$(\alpha, q, X\beta) \Rightarrow_M^* (\alpha', p, Y\beta')$  if and only if  $[\alpha q X \beta] \Rightarrow_{G_M}^* [\alpha' p Y \beta']$ .

**Proof**

i)  $\forall \delta(q, X) = (p, Y, L)$  where  $Y \neq B$ ,  $\forall Z \in \Gamma, ZqX \rightarrow pZY \in P$ .

ii)  $\forall \delta(q, X) = (p, B, L)$ ,  $\forall Z, V \in \Gamma, ZqXV \rightarrow pZBV \in P$ ,

$\forall Z \in \Gamma, ZqX] \rightarrow pZ] \in P$ .

iii)  $\forall \delta(q, X) = (p, Y, R)$ ,  $\forall Z \in \Gamma, qXZ \rightarrow YpZ, qX] \rightarrow Yp] \in P$ .

$[\alpha q X \beta] \Rightarrow_{G_M} [\alpha' p Y \beta']$  if and only if  $(\alpha, q, X\beta) \Rightarrow_M (\alpha', p, Y\beta')$ .

$[\alpha q X \beta] \Rightarrow_{G_M}^* [\alpha' p Y \beta']$  if and only if  $(\alpha, q, X\beta) \Rightarrow_M^* (\alpha', p, Y\beta')$ .

**Theorem 4.20** *If a language  $L \subseteq \Sigma^*$  is Turing-acceptable, then  $L = L(G)$  for some grammar  $G$ .*

**proof** *Assume  $L = L(M)$  TM  $\exists M . \exists . M$  always halts with empty output. Thus the configuration of  $M$  is always  $(B, h, B)$ .*

*The rules of  $G$  includes the rules of  $G_M$ , plus*

$$\begin{aligned} S &\rightarrow [ B h B ], & s B ] &\rightarrow L, \\ a L &\rightarrow L a, & [ B L &\rightarrow \varepsilon. \text{ for } \forall a \in \Sigma, \text{ where } L \in V. \end{aligned}$$

$$\forall x \in \Sigma^* . \exists . x \in L(M), (\varepsilon, s, x) \Rightarrow_{G_M}^* (\varepsilon, h, \varepsilon)$$

$$S \Rightarrow_G [ B h B ] \Rightarrow_G^* [ B s x B ] \Rightarrow_G [ B x s L \Rightarrow_G^* [ B L x \Rightarrow_G x.$$

## 4.6 Primitive Recursive Functions

*Three initial functions and two operations on functions*

**Initial Functions:**

**Zero:**  $\zeta: N \rightarrow \{0\} .\exists. \forall n \in N: \zeta(n) \cong 0.$

**Successor:**  $\sigma: N \rightarrow N .\exists. \forall n \in N: \sigma(n) \cong n+1.$

**Projection:**  $\forall k \in N, 1 \leq \forall i \leq k: \pi_i^k: N^k \rightarrow N .\exists. \pi_i^k(n_1, \dots, n_i, \dots, n_k) \cong n_i.$

**Compositon:** Given  $\forall k \in N: g: N^k \multimap N, 1 \leq \forall i \leq k: h_i: N^m \multimap N;$  define

$f: N^k \multimap N .\exists. \forall X \in N^m: f(X) \cong g(h_1(X), \dots, h_k(X)) \cong [g \circ (h_1, \dots, h_k)](X)$

**Primitive recursion:** Let  $k \geq 0.$

Given  $\forall k \in N: g: N^k \multimap N, h: N^{k+2} \multimap N;$  define

$f: N^{k+1} \multimap N .\exists. \forall X \in N^k:$

$f(X, 0) \cong g(X),$  **basis**

$f(X, \sigma(m)) \cong h(X, m, f(X, m)), m \geq 0.$  **(primitive) recursion**

***Definition of primitive recursive functions***

- (1) *Initial functions are primitive recursive.*
- (2) *If  $g: N^k \rightarrow N$ ,  $h_1, \dots, h_k: N^m \rightarrow N$  are primitive recursive, composition  $g \circ (h_1, \dots, h_k)$  is also primitive recursive.*
- (3) *If  $g: N^k \rightarrow N$  and  $h: N^{k+2} \rightarrow N$  are primitive recursive, the function obtained from  $g$  and  $h$  by primitive recursion is also primitive recursive.*
- (4) *No other function is primitive recursive.*

***Primitive recursive function***

*obtained from the initial functions by finite number of applications of composition and primitive recursive operations.*

**Constants** are primitive recursive

**Example 4.23 Constants**  $\forall k \in 1, \forall a \geq 0: K_a^k: N^k \rightarrow N$

$\exists. \forall X \in N^k: K_a^k(X) \cong a$  are primitive recursive.

$$\begin{aligned} K_j^k &= \sigma(\sigma \dots (\sigma(\zeta(\pi_1^k(n_1, \dots, n_k)))) \dots) \\ &= [\sigma^{(j)} \circ \zeta \circ \pi_1^k](n_1, \dots, n_k) \end{aligned}$$

**Arithmetic functions** are primitive recursive

**Example 4.21**  $add(m, n) \cong m + n$  is primitive recursive

$$add(m, 0) = \pi_1^1(m) = m,$$

$$add(m, n+1) = [\sigma \circ \pi_3^3](m, n, add(m, n)) = \sigma(add(m, n)).$$

**Example 4.22**  $mult(m, n) \cong m \cdot n$  is primitive recursive

$$mult(m, 0) = \zeta(m) = 0.$$

$$\begin{aligned} multi(m, n+1) &= [add \circ (\pi_1^3, \pi_3^3)](m, n, mult(m, n)) \\ &= add(m, multi(m, n)) \end{aligned}$$

*Example 4.24*  $\text{minus}(m, n) \cong m \dot{-} n = 0$ , if  $m \leq n$ ;  $m - n$ , if  $m > n$  is p.r.

$$\text{minus}(m, 0) = \pi_1^1(m)$$

$$\begin{aligned} \text{minus}(m, n+1) &= [\text{pred} \circ \pi_3^3](m, n, \text{minus}(m, n)) \\ &= \text{pred}(\text{minus}(m, n)) \end{aligned}$$

where  $\text{pred}(0) = 0$

$$\text{pred}(m+1) = \text{pred}(m) = \pi_1^2(m, \text{pred}(m))$$

*Example 4.25*  $g(m, n) \cong f^{(n)}(m) = f(f(\dots(f(m))\dots))$  is primitive recursive

$$g(m, 0) = \pi_1^1(m)$$

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

**Boolean functions and if-then-else are primitive recursive**

**Example 4.26** Following functions are primitive recursive.

(a)  $\text{neg}(x) \cong 0$  if  $x \geq 1$ ;  $1$  if  $x = 0$ .

$$\text{neg}(x) = \text{minus}(1, x).$$

(b)  $\text{and}(x, y) \cong 1$  if  $x \geq 1 \wedge y \geq 1$ ;  $0$  otherwise.

$$\text{and}(x) = \text{neg}(\text{neg}(\text{mult}(x, y))).$$

(c)  $\text{or}(x, y) \cong 1$  if  $x \geq 1 \vee y \geq 1$ ;  $0$  otherwise.

$$\text{and}(x) = \text{neg}(\text{and}(\text{neg}(x), \text{neg}(y))).$$

(d)  $\text{if-then-else}(x, y, z) \cong y$  if  $x \geq 1$ ;  $z$  otherwise.

$$\text{if-then-else}(x, y, z) = \text{add}(\text{mult}(\text{neg}(\text{neg}(x)), y), \text{mult}(\text{neg}(x), z)).$$

We write “ $x$  and  $y$ ” for  $\text{and}(x, y)$ , “ $x$  or  $y$ ” for  $\text{or}(x, y)$ , and “**if  $x$  then  $y$  else  $z$  fi**” for  $\text{if-then-else}(x, y, z)$ .



**Example 4.27** Following functions are primitive recursive.

(a)  $eq(x, y) = 1$  if  $x = y$ ,  $0$  if  $x \neq y$ .

$$eq(x, y) = neg(add(minus(x, y), minus(y, x))).$$

(b)  $gr(x, y) = 1$  if  $x > y$ ,  $0$  if  $x \leq y$ .

$$gr(x, y) = neg(neg(minus(x, y))).$$

(c)  $geq(x, y) = 1$  if  $x \geq y$ ,  $0$  if  $x < y$ .

$$geq(x, y) = gr(x, y) \text{ or } eq(x, y).$$

(d)  $ls(x, y) = 1$  if  $x < y$ ,  $0$  if  $x \geq y$ .

$$ls(x, y) = neg(geq(x, y)).$$

(d)  $leq(x, y) = 1$  if  $x \leq y$ ,  $0$  if  $x > y$ .

$$leq(x, y) = neg(gr(x, y)).$$

**Example 4.28**  $max^k(n_1, \dots, n_k) \cong \max\{n_1, \dots, n_k\}$  is p.r. for  $k \geq 1$ .

**Solution**  $max^1(n_1) = \pi_1^1(n_1)$

$max^k(n_1, \dots, n_k) = \mathbf{if} \ n_k > max^{k-1}(n_1, \dots, n_{k-1}) \ \mathbf{then} \ n_k \ \mathbf{else} \ max^{k-1}(n_1, \dots, n_{k-1})$

$R: (\{0, 1\}^*)^k \rightarrow \{0, 1\}$  is a **Boolean function or predicate**.

We use 1 to represent TRUE and 0 to represent FALSE

If  $R(n_1, \dots, n_k)$  is a **predicate**,  $R(n_1, \dots, n_k)$  means  $[R(n_1, \dots, n_k) = 1]$

We often write **formula** instead of the name of the **predicate**

$g(m) \cong 1$  if  $(\exists n)[n^2 = m]$ ; 0 otherwise.

We write  $(\exists n)[n^2 = m]$  instead of  $g(m)$ .

**Theorem 4.29** Let  $f: N^{k+1} \rightarrow \{0, 1\}$  be a **primitive recursive predicate**.

Then following **three operations** on  $f$  are also **primitive recursive**.

(a) **bounded universal quantifier**

$$g(n_1, \dots, n_k, m) \cong (\forall i)_{i \leq m} f(n_1, \dots, n_k, i).$$

(b) **bounded existential quantifier**

$$h(n_1, \dots, n_k, m) \cong (\exists i)_{i \leq m} f(n_1, \dots, n_k, i).$$

(c) **bounded minimization**

$$t(n_1, \dots, n_k, m) \cong (\min i)_{i \leq m} f(n_1, \dots, n_k, i) \text{ if } (\exists i)_{i \leq m} f(n_1, \dots, n_k, i), \\ \cong 0 \text{ otherwise.}$$

**Proof**

$$(a) \quad g(n_1, \dots, n_k, 0) = f(n_1, \dots, n_k, 0)$$

$$g(n_1, \dots, n_k, m+1) = [f(n_1, \dots, n_k, m+1) \text{ and } g(n_1, \dots, n_k, m)]$$

$$(b) \quad h(n_1, \dots, n_k, 0) = f(n_1, \dots, n_k, 0)$$

$$h(n_1, \dots, n_k, m+1) = [f(n_1, \dots, n_k, m+1) \text{ or } h(n_1, \dots, n_k, m)]$$

$$(c) \quad t(n_1, \dots, n_k, 0) = f(n_1, \dots, n_k, 0)$$

$$t(n_1, \dots, n_k, m+1) = [\text{if } (\exists i)_{i \leq m} f(n_1, \dots, n_k, i) \text{ then } t(n_1, \dots, n_k, m) \\ \text{else if } f(n_1, \dots, n_k, m+1) \text{ then } m+1 \text{ else } 0 \text{ fi fi}]$$

**Example 4.30** The following functions are primitive recursive.

(a)  $\text{quot}(m, n) \cong \lfloor m/n \rfloor$  if  $n > 0$ ; 0 otherwise.

$$\text{quot}(m, n) = \mathbf{if } n > 0 \mathbf{ then } (\min i)_{i \leq m} [(i+1) \cdot n > m] \mathbf{ else } 0.$$

(b)  $\text{mod}(m, n) \cong m - \lfloor m/n \rfloor \cdot n$  if  $n > 0$ ; 0 otherwise.

$$\text{mod}(m, n) = \mathbf{if } n > 0 \mathbf{ then } (\min i)_{i \leq m} [\text{quot}(m, n) \cdot n + i = m] \mathbf{ else } 0.$$

(c)  $\text{prime}(n) \cong 1$  if  $n$  is prime; 0 otherwise.

$$\text{prime}(n) = (n \geq 2) \text{ and } (\forall j)_{2 \leq j \leq n-1} [\text{mod}(n, j) > 0]; \text{ or}$$

$$\text{prime}(n) = \text{prime}'(n, \text{pred}(n))$$

$$\text{where } \text{prime}'(n, k) = (n \geq 2) \text{ and } (\forall j)_{j \leq k} [j \leq 1 \text{ or } \text{mod}(n, j) > 0].$$

$\therefore (\forall i)_{l(n) \leq i \leq u(n)}, (\exists i)_{l(n) \leq i \leq u(n)}, \text{ and } (\min i)_{l(n) \leq i \leq u(n)}$

are **primitive recursive**, if  $l(n)$  and  $u(n)$  are primitive recursive.

**Example 4.31**  $f(0) \cong 1$

$f(n) \cong$  *nth digit to the right of the decimal point of  $\text{srqt}(2)$ .*

**Solution**

Let  $g(n)$  be the integer  $m$  whose decimal expansion is equal to  $f(0)...f(n)$

e.g.  $g(3) = 1414$ . Then

$$g(n) = (\min k)_{k \leq 10^{n+1}} [(k+1)^2 > 2 \cdot 10^{2n}].$$

$$f(n) = \text{mod}(g(n), 10).$$

## 4.7 Pairing Function and Gödel Numbering

To show Fibonacci function  $F: N \rightarrow N$

$$F(0) = 0, \quad F(1) = 1,$$

$F(n+2) = F(n) + F(n+1)$  is recursive primitive

Pairing function  $f: N^2 \rightarrow N$

i) (**Bijectivity**)  $f$  is one-to-one and onto.

ii) (**Primitive recursiveness**)  $f$  is primitive recursive.

iii) (**Monotonicity**)  $f(i, j) < f(i+1, j), f(i, j) < f(i, j+1) \quad \forall i, j \in N.$

Suppose  $f$  is a pairing function.

$l_f, r_f: N \rightarrow N \ .\exists. f(l_f(n), r_f(n)) \cong n$  are

**well defined and primitive recursive.**

$$l_f(n) = (\min i)_{i \leq m} (\exists j)_{j \leq n} [f(i, j) = n],$$

$$r_f(n) = (\min j)_{j \leq n} [f(l_f(n), j) = n].$$

**Example 4.32**  $\pi(i, j) = (i+j)(i+j+1)/2 + j$  is a **pairing function**.

**Solution** 1-1 onto, primitive recursive, monotonic.

We write  $\langle i, j \rangle = \pi(i, j)$ ,  $l(n) = l_\pi(n)$ ,  $r(n) = r_\pi(n)$ .

$$\therefore \langle l(n), r(n) \rangle = n.$$

**Example 4.33** Fibonacci function  $F$  is primitive recursive.

$$F(0) = 0$$

$$F(1) = 1$$

$$F(n+2) = F(n) + F(n+1)$$

**Solution**  $G(n) = \langle F(n), F(n+1) \rangle$

$$G(0) = \langle 0, 1 \rangle$$

$$G(n+1) = \langle F(n+1), F(n+2) \rangle = \langle F(n+1), F(n) + F(n+1) \rangle$$

$$= \langle r(G(n)), l(G(n)) + r(G(n+1)) \rangle$$

$\therefore F(n) = l(G(n))$  is **primitive recursive**.

$f: \cup_{k \geq 1} N^k \rightarrow N$  is a Gödel numbering (of integer sequences)

i) (**Bijectivity**)  $f$  is one-to-one and onto.

ii) (**Primitive recursiveness**)  $f|_{N^k}$  is primitive recursive  $\forall k \geq 1$ .

where  $f|_{N^k}$  is the function restricted to the domain  $N^k$ .

iii) (**Monotonicity**)  $1 \leq \forall i \leq k$

$$f(n_1, \dots, n_{i-1}, n_i, n_{i+1}, \dots, n_k) < f(n_1, \dots, n_{i-1}, n_i+1, n_{i+1}, \dots, n_k).$$

### Example 4.34

$\tau(n_1, \dots, n_{k-1}, n_k) = \langle k \dot{-} 1, \langle n_1, \langle \dots \langle n_{k-1}, n_k \rangle \dots \rangle \rangle \rangle$  is a Gödel numbering

### Solution

Suppose  $\tau(n_1, \dots, n_k) = \tau(m_1, \dots, m_s)$

$$\langle k \dot{-} 1, \langle n_1, \langle \dots \langle n_{k-1}, n_k \rangle \dots \rangle \rangle \rangle = \langle s \dot{-} 1, \langle m_1, \langle \dots \langle m_{s-1}, m_s \rangle \dots \rangle \rangle \rangle$$

$k = s$ , since  $\pi$  is one-to-one.



$\tau/N^k$  is one-to-one

If  $k=1$ ,  $\tau(n) = \langle 0, n \rangle$  is one-to-one

If  $k > 1$ ,  $\tau(n_1, n_2, \dots, n_k) = \langle k - 1, \langle n_1, r(\tau(n_2, \dots, n_k)) \rangle \rangle$

$\tau/N^{k-1}$  is one-to-one and  $\tau$  is one-to-one.

$\tau$  is onto.

$\forall n \in N, \exists (n_1, n_2, \dots, n_k) \ni \tau(n_1, n_2, \dots, n_k) = n.$

$k = l(n) + 1.$

$1 \leq \forall i \leq k$ ,  $item(n, i) \cong l(r^{(i)}(n))$  if  $1 \leq \forall i \leq k-1$ ;  $r^{(k)}(n)$  if  $i = k.$

$\forall n \in N, \exists^1 (item(n, 1), \dots, item(n, k)) \ni \tau(item(n, 1), \dots, item(n, k)) = n.$

$\therefore \tau$  is onto.

Note that  $\tau(n_1, \dots, n_{k-1}, n_k) = \langle k - 1, \langle n_1, \langle \dots \langle n_{k-1}, n_k \rangle \dots \rangle \rangle \rangle$

We write  $\langle n_1, n_2, \dots, n_k \rangle \cong r(\tau(n_1, n_2, \dots, n_k)) = \langle n_1, \langle \dots \langle n_{k-1}, n_k \rangle \dots \rangle \rangle$ .

We write  $[n_1, n_2, \dots, n_k] \cong \tau(n_1, n_2, \dots, n_k)$ . We write  $size(n) \cong l(n) + 1$ .

$\therefore [item(n, 1), \dots, item(n, size(n))] = n$ .

**Example 4.35**

(a)  $list(m, 0) \cong 0$  and  $list(m, k) = [m, m, \dots, m]$ , if  $k \geq 1$ .

$$list(m, 0) = 0$$

$$list(m, 1) = \langle 0, m \rangle$$

$$list(m, k+1) = \langle k, \langle m, r(list(m, k)) \rangle \rangle$$

$$\begin{aligned} list(m, k+1) &= \langle k, \langle m, r(list(m, k)) \rangle \rangle \\ &= \langle k, \langle m, \langle m, r(list(m, k-2)) \rangle \rangle \rangle \\ &= \dots \\ &= \langle k, \langle m, \langle m, \dots, \langle m, r(list(m, 1)) \rangle \dots \rangle \rangle \rangle \\ &= \langle k, \langle m, \langle m, \dots, \langle m, m \rangle \dots \rangle \rangle \rangle \\ &= [m, m, \dots, m] \end{aligned}$$

(b)  $find([n_1, \dots, n_k], m) \cong \min \{i \mid 1 \leq i \leq k, n_i = m\}$  if such an  $i$  exists,  
 $\cong 0$ , otherwise.

$$find(n, m) = (\min i)_{1 \leq i \leq size(n)} [item(n, i) = m]$$

(b)  $replace([n_1, \dots, n_k], m, i) \cong [n_1, \dots, n_{i-1}, m, n_{i+1}, \dots, n_k]$  if  $1 \leq i \leq k$ ,

$\cong 0$  otherwise.

**If**  $1 \leq i \leq \text{size}(n)$  **then**

$\text{replace}(n, m, i)$  ( $\min t$ ) $_{t \leq \text{list}(n+m, \text{size}(n))}$  [ $\text{size}(t) = \text{size}(n)$ ]

and

$\text{find}(n, m) = (\min i)_{1 \leq i \leq \text{size}(n)}$  [ $\text{item}(n, i) = m$ ]

$$[n_1] = \langle 0, n_1 \rangle =$$

$$[n_1, n_2] = \langle 1, \langle n_1, n_2 \rangle \rangle$$

$$[n_1, n_2, n_3] = \langle 2, \langle n_1, \langle n_2, n_3 \rangle \rangle \rangle$$

$$[n_1, n_2, n_3, n_4] = \langle 3, \langle n_1, \langle n_2, \langle n_3, n_4 \rangle \rangle \rangle \rangle$$

**Example 4.36**  $f: N \rightarrow N$ ,

$$f(0) = 1,$$

$$f(n+1) = f(0)^{n+1} + f(1)^n + \dots + f(n)^1 \text{ is primitive recursive.}$$

*Solution*

**Theorem 4.37** Let

$$f(n_1, \dots, n_k, 0) = g(n_1, \dots, n_k)$$

$$f(n_1, \dots, n_k, m+1) = h(n_1, \dots, n_k, m, [f_0, f_1, \dots, f_m])$$

where  $f_i = f(n_1, \dots, n_k, i)$ . If  $g$  and  $h$  are primitive recursive, then  $f$  is also primitive recursive.

## 4.8 Partial Recursive Function

**Unbounded minimization:** Given a total predicate  $g: N^{k+1} \rightarrow \{0, 1\}$ , define  $f: N^k \rightarrow N$ .

$$f(n_1, \dots, n_k) \cong (\min m) g(n_1, \dots, n_k, m) \text{ if } (\exists m) g(n_1, \dots, n_k, m),$$

$$\cong \uparrow(\text{undefined}) \text{ otherwise.}$$

We simply write  $f(n_1, \dots, n_k) = (\min m) g(n_1, \dots, n_k, m)$ .

### Partial Recursive Function

(1) Initial functions and

(2) finite number of application of **composition**, **primitive recursion**, and **unbounded minimization**.

A partial recursive function  $f$  is called **recursive function**, if  $f$  is total.

**Recursive set:** A set  $A \subseteq N^k$  is recursive if its **characteristic function**

$$\begin{aligned}\chi_A(n_1, \dots, n_k) &\cong 1, \text{ if } (n_1, \dots, n_k) \in A \\ &\cong 0, \text{ if } (n_1, \dots, n_k) \notin A\end{aligned}$$

is a **(total) recursive function**.

**Recursively enumerable set:** A set  $A \subseteq N^k$  is recursively enumerable if its **semi-characteristic function**

$$\begin{aligned}\sigma_A(n_1, \dots, n_k) &\cong 1, \text{ if } (n_1, \dots, n_k) \in A \\ &\cong \uparrow, \text{ if } (n_1, \dots, n_k) \notin A\end{aligned}$$

is a **partial recursive function**.

We say a set  $A \subseteq N^k$  is **primitive recursive** if  $\chi_A$  is **primitive recursive**.

**Example 4.39**  $A = \{n \mid (\exists m) R(n, m)\}$  for some recursive set  $R$ ,  $A$  is r.e.

(a)  $\sigma_A(n) = \text{neg}(\text{neg}(1 + (\min m)R(n, m)))$ .

Any fixed alphabet  $\Sigma = \{s_1, \dots, s_k\}$  with a fixed ordering  $s_1 \triangleleft s_2 \triangleleft \dots \triangleleft s_k$  the **lexicographic ordering**  $\triangleleft$  on strings in  $\Sigma^*$  is defined as follow:

Let  $x = a_1 a_2 \dots a_m$  and  $y = b_1 b_2 \dots b_n$  where  $a_i, b_j \in \Sigma$  for  $1 \leq i \leq m, 1 \leq j \leq n$ .

We say  $x \triangleleft y$ , if

(a)  $|x| < |y|$  (i.e.,  $m < n$ ), or

(b)  $|x| = |y|$  and  $(\exists i)_{1 \leq i \leq m} [a_1 = b_1, \dots, a_{i-1} = b_{i-1}, a_i \triangleleft b_i]$ .

$\Sigma^* = \{\epsilon, s_1, s_2, \dots, s_k, s_1 s_1, s_1 s_2, \dots, s_k s_k, s_1 s_1 s_1, \dots\}$

$\iota_\Sigma: \mathbb{N} \leftrightarrow \Sigma^*$ .  $\exists$ .  $\iota_\Sigma(n)$  is  $n$ -th string in  $\Sigma^*$  under

the **lexicographic ordering**.  $\triangleleft$  (starting from  $\iota_\Sigma(0) = \epsilon$ ).

**Theorem 4.40** Let  $\Sigma = \{s_1, \dots, s_k\}$  and  $s_1 \triangleleft s_2 \triangleleft \dots \triangleleft s_k$ . Then

$\forall x = s_{i_n} s_{i_{n-1}} \dots s_{i_0} \in \Sigma^*$  with  $n \geq 0$ ,

$$\iota^{-1}(x) = i_n \cdot k^n + i_{n-1} \cdot k^{n-1} + \dots + i_1 \cdot k^1 + i_0.$$



*Example 4.41 Suppose*

*A partial function  $f: (\Sigma^*)^k \rightarrow \Sigma^*$  is called **primitive recursive** (or **partial recursive**) if the function  $\tilde{f}: N^k \rightarrow N$  is defined by*

$$\tilde{f}(n_1, \dots, n_k) = \iota_{\Sigma}^{-1}(f(\iota_{\Sigma}(n_1), \dots, \iota_{\Sigma}(n_k)))$$

*is **primitive recursive** (or, respectively, **partial recursive**)*

**Lemma 4.43** *For any grammar  $G$  the predicate*

$$d_G(\alpha, \beta, k) = [\alpha \Rightarrow^k \beta] \text{ is primitive recursive.}$$

**Proof**  $d_G(\alpha, \beta, 0) = eq(\alpha, \beta)$

$$d_G(\alpha, \beta, k+1) = (\exists \gamma)_{|\gamma| \leq |\beta|+1} [d_G(\alpha, \gamma, k) \text{ and } \gamma \Rightarrow \beta]$$

*where  $l$  is the length of the longest string in the LHS of any rules in  $G$ .*

**Theorem 4.44** *Every Turing-computable function is partial recursive.*

**Proof** If  $f: (\Sigma^*)^k \rightarrow \Sigma^*$  is Turing-computable, then  $\exists$  a grammar  $G_M$ ,  $\exists$ .

$$f(x_1, \dots, x_k) = y \Leftrightarrow (\exists t) d_{G_M}([Bx_1Bx_2B\dots Bx_k sB], [ByhB], t)$$