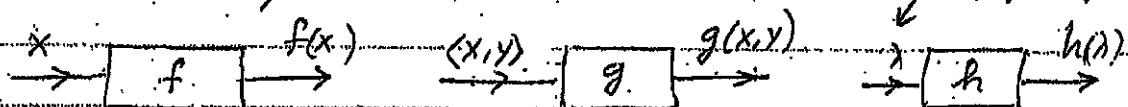


Ch. 6 - RECURSIVE FUNCTIONS & RELATIONS

S1. Primitive Recursive Functions. : In this chapter, it will be helpful if we think of a function as a rule which assigns an output to given input. The basic idea is to express the output in terms of the input. A function is still mathematically, a set of ordered pairs which satisfies the condition $(x,y) \in f \wedge (x,z) \in f \Rightarrow y = z$. An n -ary function on \mathbb{N} is just a function from \mathbb{N}^n to \mathbb{N} . When $n=0$, \mathbb{N}^0 = set of 0-tuples = $\{\emptyset\}$.



function of 1 variable function of 2 variables function of zero variables
 $f: \mathbb{N} \rightarrow \mathbb{N}$ $f: \mathbb{N}^2 \rightarrow \mathbb{N}$ $f: \mathbb{N}^0 \rightarrow \mathbb{N}$

A constant can be viewed as a function of 0 variables. $\pi \rightarrow$
 no input

Def. A function $f: \mathbb{N}^k \rightarrow \mathbb{N}$ is said to be primitive recursive if it can be obtained from the initial functions by a finite number of applications of cartesian products, compositions, & primitive recursions.

Def. The initial functions are: (a) the zero function of 0 var. : 0
 (b) the zero function of 1 variable : $z(x) = 0$ for all $x \in \mathbb{N}$
 (c) the successor function : $s(x) = \text{successor of } x$ for all $x \in \mathbb{N}$
 (d) the projective functions : $I_{k,n}(x_1, \dots, x_n) = x_k$ for $1 \leq k \leq n$; and $I_{0,n}(x_1, \dots, x_n) = ?$

(2)

Example 1: Here are some values of some initial functions.

- (a) $Z(2) = 0, Z(5) = 0, Z(8) = 0$
- (b) $S(2) = 3, S(5) = 6, S(8) = 9$
- (c) $I_{1,2}(5,7) = 5, I_{2,3}(3,4,5) = 4, I_{1,1}(4) = 4,$
 $I_{2,2}(4,6) = 6, I_{0,2}(3,4) = 7, I_{0,1}(7) = 7$

Def. Let $g: \mathbb{N}^n \rightarrow \mathbb{N}^k$ and $h: \mathbb{N}^m \rightarrow \mathbb{N}^l$ be functions. The cartesian product of the functions g & h is the function $f: \mathbb{N}^n \rightarrow \mathbb{N}^{k+m}$ defined by

$$f(x) = \langle g(x), h(x) \rangle \text{ & we write } f = g \wedge h$$

Here x abbreviates $\langle x_1, \dots, x_n \rangle$

Example 2: Let $f = s \wedge I_{1,1}$. Then $f: \mathbb{N} \rightarrow \mathbb{N}^2$ is given by $f(x) = \langle s(x), I_{1,1}(x) \rangle = \langle x+1, x \rangle$

Def. Let $h_1, h_2, \dots, h_k: \mathbb{N}^n \rightarrow \mathbb{N}$ and $g: \mathbb{N}^k \rightarrow \mathbb{N}$ be functions. Then we can define a new function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ by putting

$$f(x) = g(h_1(x), h_2(x), \dots, h_k(x)).$$

Here x again abbreviates $\langle x_1, \dots, x_n \rangle$. The function f is said to be obtained from g and h_1, \dots, h_k by composition & we write $f = g \circ (h_1 \wedge h_2 \wedge \dots \wedge h_k)$

Example 3 Let $g: \mathbb{N}^3 \rightarrow \mathbb{N}$ be defined by

$g(x_1, y, z) = 2x + 3y + z^2$. Also let $h_i: \mathbb{N}^2 \rightarrow \mathbb{N}$ be defined by $h_1(x_1, x_2) = x_1 + x_2$, $h_2(x_1, y) = x_1 y$ and $h_3(x_1, x_2) = x_1 - x_2$. Then if $f = g \circ (h_1 \wedge h_2 \wedge h_3)$ we will have

$$f(x_1, x_2) = g(h_1(x_1, x_2), h_2(x_1, x_2), h_3(x_1, x_2)) = 2(x_1 + x_2) + 3x_1 x_2 + (x_1 - x_2)^2$$

(3)

Def. Let $g: \mathbb{N}^n \rightarrow \mathbb{N}$ & $h: \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ be functions where $n \geq 0$.

Then we can define a new function $f: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ by putting $f(x, 0) = g(x)$ & $f(x, s(y)) = h(x, y, f(x, y))$.

Here \underline{x} again abbreviates (x_1, \dots, x_n) . The function f is said to be obtained from g and h by primitive recursion and we write $f = \text{prec}(g, h)$.

Ex.4 Let $g: \mathbb{N}^1 \rightarrow \mathbb{N}$ be defined by $g(x) = x$ and let $h: \mathbb{N}^3 \rightarrow \mathbb{N}$ be defined by $h(x_1, x_2, x_3) = 2x_3 + x_2 + 1$.

Now if $f = \text{prec}(g, h)$, then $f: \mathbb{N}^2 \rightarrow \mathbb{N}$ and

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

$$f(x, 1) = f(x, s(0)) = h(x, 0, f(x, 0))$$

$$= 2 \cdot f(x, 0) + 0 + 1 = 2x + 1$$

$$f(x, 2) = f(x, s(1)) = h(x, 1, f(x, 1))$$

$$= 2 \cdot f(x, 1) + 1 + 1 = 2(2x+1) + 1 + 1$$

$$= 4x + 4$$

$$f(x, 3) = f(x, s(2)) = h(x, 2, f(x, 2))$$

$$= 2 \cdot f(x, 2) + 2 + 1 = 2(4x+4) + 2 + 1$$

$$= 8x + 11$$

$$f(x, 4) = f(x, s(3)) = h(x, 3, f(x, 3))$$

$$= 2 \cdot f(x, 3) + 3 + 1 = 2(8x+11) + 3 + 1$$

$$= 16x + 26.$$

In general $f(x, y) = 2^y \cdot x + 2^{y+1} - (y+2)$, but it is not easy to see this.

Ex.5 Let g = the constant 1 and $h(x_1, x_2) = 2x_2 + 1$. Then $g: \mathbb{N}^0 \rightarrow \mathbb{N}$ & $h: \mathbb{N}^2 \rightarrow \mathbb{N}$. So if we put $f = \text{prec}(g, h)$ we will get a function $f: \mathbb{N}^1 \rightarrow \mathbb{N}$.

(4)

Ex.5 Now $f(0) = g = 1$.

$$\begin{aligned}
 f(1) &= f(s(0)) = h(0, f(0)) \\
 &= 2 \cdot f(0) + 1 = 2 + 1 = 3 \\
 f(2) &= f(s(1)) = h(1, f(1)) \\
 &= 2 \cdot f(1) + 1 = 2(3) + 1 = 7 \\
 f(3) &= f(s(2)) = h(2, f(2)) \\
 &= 2 \cdot f(2) + 1 = 2(7) + 1 = 15.
 \end{aligned}$$

In general, it is very easy to see that

$$f(y) = 2^{y+1} - 1.$$

Ex.6 Now let $g = 1$ and $h(x_1, x_2) = 2x_2 + x_1$.Then $g: \mathbb{N}^0 \rightarrow \mathbb{N}$ and $h: \mathbb{N}^2 \rightarrow \mathbb{N}$. So if $f = \text{prec}(g, h)$ then f will be a function from \mathbb{N} to \mathbb{N} . Also,

$$\begin{aligned}
 f(0) &= g = 1 \\
 f(1) &= f(s(0)) = h(0, f(0)) \\
 &= 2 \cdot f(0) + 0 = 2 \\
 f(2) &= f(s(1)) = h(1, f(1)) \\
 &= 2 \cdot f(1) + 1 = 2(2) + 1 = 5 \\
 f(3) &= f(s(2)) = h(2, f(2)) \\
 &= 2 \cdot f(2) + 2 = 2(5) + 2 = 12 \\
 f(4) &= f(s(3)) = h(3, f(3)) \\
 &= 2 \cdot f(3) + 3 = 2(12) + 3 = 27 \\
 f(5) &= f(s(4)) = h(4, f(4)) \\
 &= 2 \cdot f(4) + 4 = 2(27) + 4 = 58 \\
 f(6) &= f(s(5)) = h(5, f(5)) \\
 &= 2 \cdot (58) + 5 = 2(58) + 5 = 121
 \end{aligned}$$

In general $f(y) = 2^{y+1} - (y+1)$ but this is moderately difficult to see.

(5)

92. Demonstrating that certain functions are primitive recursive.

Ex. 1 Let $f(x) = 3$ for each $x \in N$. Show that $f: N \rightarrow N$ is a primitive recursive function.

Sol. $f(x) = s(s(s(z(x))))$. So $f = s \circ s \circ s \circ z$. Hence f can be obtained from the initial functions by a finite number of applications of compositions. So f is primitive recursive.

Ex. 2 Let $g(x, y) = y + z$. Show that $g: N^2 \rightarrow N$ is primitive recursive.

Sol. $g(x, y) = s(s(I_{2,2}(x, y)))$ So $g = s \circ s \circ I_{2,2}$. Hence g is a primitive recursive function.

Ex. 3 Let $\text{ADD}(x, y) = x + y$ Show that $\text{ADD}: N^2 \rightarrow N$ is a primitive recursive function.

Sol. We will find primitive recursive functions g and h such that $\text{ADD} = \text{prec}(g, h)$. Now

$$\begin{aligned}\text{ADD}(x, 0) &= x && \leftarrow g(x) \\ \text{ADD}(x, s(y)) &= \text{ADD}(x, y+1) = x+y+1 \\ &= (x+y)+1 = \text{ADD}(x, y)+1 \\ &= s(\text{ADD}(x, y)) && \leftarrow h(x, y, \text{ADD}(x, y))\end{aligned}$$

Hence $g(x) = x$ So $g = I_{1,1}$.

And $h(x, y, \text{ADD}(x, y)) = s(\text{ADD}(x, y))$. So $h = s \circ I_{3,3}$. Thus $\text{ADD} = \text{prec}(g, h) = \text{prec}(I_{1,1}, s \circ I_{3,3})$. Hence ADD is primitive recursive.

(6)

Ex.4 Let $MULT(x, y) = x \cdot y$, Show that $MULT: \mathbb{N}^2 \rightarrow \mathbb{N}$ is a primitive recursive function.

Sol.

Again we will find primitive recursive functions g and h such that $MULT = \text{prec}(g, h)$. Now

$$MULT(x, 0) = x \cdot 0 = 0 \quad \leftarrow g(x)$$

$$MULT(x, s(y)) = MULT(x, y+1) = x \cdot (y+1)$$

$$= x \cdot y + x = MULT(x, y) + x$$

$$= ADD(MULT(x, y), x) \quad \leftarrow h(x, y, MULT(x, y))$$

$\therefore g(x) = 0$ for each $x \in \mathbb{N}$. So $g = z$. And

$$h(x, y, MULT(x, y)) = ADD(MULT(x, y), x). \quad \text{So}$$

$$h = ADD \circ (I_{3,3} \wedge I_{1,3}) \quad \text{Hence } MULT = \text{prec}(g, h)$$

$$= \text{prec}(z, ADD \circ (I_{3,3} \wedge I_{1,3}))$$

$$= \text{prec}(z, \text{prec}(I_{1,1}, s \circ I_{3,3}) \circ (I_{3,3} \wedge I_{1,3}))$$

Thus $MULT$ is primitive recursive.

Ex.5 Let $PRED(y) = \begin{cases} 0 & \text{if } y = 0, \\ y-1 & \text{if } y > 0. \end{cases}$

Show that $PRED: \mathbb{N} \rightarrow \mathbb{N}$ is primitive recursive.

Sol. We will find primitive recursive functions g and h such that $PRED = \text{prec}(g, h)$. Now

$$PRED(0) = 0 \quad \leftarrow g$$

$$PRED(s(y)) = s(y) - 1 = y \quad \leftarrow h(y, PRED(y))$$

So $g = 0$ and $h(y, PRED(y)) = y$. Thus
 $h = I_{1,2}$. Hence

$$PRED = \text{prec}(g, h)$$

$$= \text{prec}(0, I_{1,2})$$

Thus $PRED$ is a primitive recursive function.

(7)

Ex.6 Let $\text{MONUS}(x, y) = \begin{cases} x - y & \text{if } x \geq y, \\ 0 & \text{if } x < y. \end{cases}$ We usually write $x - y$ for $\text{MONUS}(x, y)$. Show that MONUS is a primitive recursive function.

Sol. We will show that $\text{MONUS} = \text{prec}(g, h)$ where g and h are primitive recursive functions. Now

$$\text{MONUS}(x, 0) = x - 0 = x \quad \leftarrow g(x)$$

$$\text{MONUS}(x, s(y)) = x - s(y) = x - (y + 1)$$

$$= \text{PRED}(x - y) \quad \leftarrow h(x, y, \text{MONUS}(x, y))$$

So $g(x) = x$ for each $x \in \mathbb{N}$. Thus $g = I_{1,1}$. Also

$h(x, y, \text{MONUS}(x, y)) = \text{PRED}(\text{MONUS}(x, y))$. Hence

$$\text{MONUS} = \text{prec}(g, h) = \text{prec}(I_{1,1}, \text{PRED} \circ I_{3,3})$$

$$= \text{prec}(I_{1,1}, \text{prec}(0, I_{1,2}) \circ I_{3,3})$$

Thus MONUS is a primitive recursive function.

Ex.7 Let $\text{SWITCH}(x, y) = \langle y, x \rangle$. Show that $\text{SWITCH}: \mathbb{N}^2 \rightarrow \mathbb{N}^2$ is a primitive recursive function.

Sol. We have $\text{SWITCH}(x, y) = \langle I_{2,2}(x, y), I_{1,2}(x, y) \rangle$.

$$\text{So } \text{SWITCH} = I_{2,2} \wedge I_{1,2}.$$

Ex.8 Let $\text{SIGN}(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0 \end{cases}$ & $\overline{\text{SIGN}}(0) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x > 0. \end{cases}$

Show that SIGN & $\overline{\text{SIGN}}$ are primitive recursive.

Sol. $\text{SIGN}(0) = 0$ & $\text{SIGN}(s(y)) = 1$. So $\text{SIGN} = \text{prec}(0, s_0 z_0 I_{1,2})$

$\overline{\text{SIGN}}(0) = 1$ & $\overline{\text{SIGN}}(s(y)) = 0$. So $\overline{\text{SIGN}} = \text{prec}(s_0 0, z_0 I_{1,2})$.

Ex.9 Let $\text{ABS}(x, y) = |x - y|$. Show that ABS is primitive recursive.

$$\text{Ans. } \text{ABS} = \text{ADD} \circ (\text{MONUS} \wedge \text{MONUS} \circ (I_{2,2} \wedge I_{1,2}))$$

$$\text{bec. } |x-y| = |x-y| + |y-x|.$$

(8)

83. Ackermann's function.

Most of the "natural" functions that we encounter in Number Theory are primitive recursive. It is, in fact, quite difficult to find a function from \mathbb{N} to \mathbb{N} which is not primitive recursive.

Def. The Ackermann's function $A: \mathbb{N}^2 \rightarrow \mathbb{N}$ is defined by simultaneous recursion as follows.

(a) $A(x+1, y+1) = A(x, A(x+1, y)),$

(b) $A(x+1, 0) = A(x, 1), \text{ and}$

(c) $A(0, y) = y+1.$

Theorem 1: let $f(x) = A(x, x)$. Then $f: \mathbb{N} \rightarrow \mathbb{N}$ is not a primitive recursive function.

The essence of the proof is to show that $f(x)$ grows faster than each primitive recursive function, but it is way too complicated to include in these notes.

Let us calculate $A(x, y)$ for various small values of x and y .

Ex. 1 $A(0, y) = y+1 \quad \text{by (c)}$

$$A(1, 0) = A(0+1, 0) = A(0, 1) = 2 \quad \text{by (b)}$$

$$\begin{aligned} A(1, 1) &= A(0+1, 0+1) = A(0, A(0+1, 0)) \quad \text{by (a)} \\ &= A(0, A(1, 0)) = A(0, 2) = 3. \end{aligned}$$

$$\begin{aligned} A(1, 2) &= A(0+1, 1+1) = A(0, A(0+1, 1)) \quad \text{by (a)} \\ &= A(0, A(1, 1)) = A(0, 3) = 4. \end{aligned}$$

(9)

Ex.2 In general, it can be shown that $A(0, y) = y + 2$.

$$\text{Now } A(2, 0) = A(1+1, 0) = A(1, 1) = 3 \text{ by (6)}$$

$$A(2, 1) = A(1+1, 0+1) = A(1, A(1+1, 0)) \text{ by (a)}$$

$$= A(1, A(2, 0)) = A(1, 3) = 5$$

$$A(2, 2) = A(1+1, 1+1) = A(1, A(1+1, 1)) \text{ by (a)}$$

$$= A(1, A(2, 1)) = A(1, 5) = 7$$

$$A(2, 3) = A(1+1, 2+1) = A(1, A(1+1, 2)) \text{ by (a)}$$

$$= A(1, A(2, 2)) = A(1, 7) = 9.$$

In general, it can be shown that $A(2, y) = 2y + 3$.

Ex.3 $A(3, 0) = A(2+1, 0) = A(2, 1) \text{ by (6)}$

$$= 5 = 2^{0+3} - 3$$

$$A(3, 1) = A(2+1, 0+1) = A(2, A(2+1, 0)) \text{ by (a)}$$

$$= A(2, A(3, 0)) = A(2, 2^{0+3} - 3)$$

$$= 2 \cdot (2^{0+3} - 3) + 3 = 2^{1+3} - 3$$

$$A(3, 2) = A(2+1, 1+1) = A(2, A(2+1, 1)) \text{ by (a)}$$

$$= A(2, A(3, 1)) = A(2, 2^{1+3} - 3)$$

$$= 2 \cdot (2^{1+3} - 3) + 3 = 2^{2+3} - 3.$$

In general, it can be shown that $A(3, y) = 2^{y+3} - 3$.

Ex.4 $A(4, 0) = A(3+1, 0) = A(3, 1) \text{ by (6)}$

$$= 2^4 - 3 = 2^2 - 3$$

$$A(4, 1) = A(3+1, 0+1) = A(3, A(3+1, 0)) \text{ by (a)}$$

$$= A(3, 2^4 - 3) = 2^{2^4-3+3} - 3 = 2^{2^4} - 3$$

$$A(4, 2) = A(3+1, 1+1) = A(3, A(3+1, 1)) \text{ by (a)}$$

$$= A(3, 2^2 - 3) = 2^{2^2-3+3} - 3 = 2^{2^4} - 3.$$

In general, it can be shown that $(4+3) \cdot 2^y - 3$

$$A(4, y) = 2^{2^y} - 3.$$

(10)

Theorem 2: For each $x \in N$, we have

$$(i) A(1, y) = y+2, \quad (ii) A(2, y) = 2(y+3)-3, \quad \frac{(y+3)}{2} \text{ s } 2$$

$$(iii) A(3, y) = 2^{y+3} - 3, \quad (iv) A(4, y) = 2^{\frac{y+3}{2}} - 3.$$

Proof: The proofs are all by induction on y .

(i) For $y=0$, we have $A(1, 0) = A(0, 1) = 2 = 0+2$.

So the result is true for $y=0$. Suppose the result is true for y . Then $A(1, y) = y+2$. So

$$\begin{aligned} A(1, y+1) &= A(0+1, y+1) = A(0, A(0+1, y)) \text{ by (a)} \\ &= A(0, A(1, y)) = A(0, y+2) \text{ by ind. hyp.} \\ &= (y+2)+1 = (y+1)+2. \end{aligned}$$

So if the result is true for y , it will be true for $y+1$. Hence the result is true for all y .

(ii) $A(2, 0) = A(1, 1) = 3 = 2(0+3)-3$. So result is true for $y=0$. Now suppose the result is true for y . Then $A(2, y) = 2(y+3)-3$. So

$$\begin{aligned} A(2, y+1) &= A(1+1, y+1) = A(1, A(1+1, y)) = A(1, A(2, y)) \\ &= A(1, 2(y+3)-3) = [2(y+3)-3] + 2 = 2[(y+1)+3]-3. \end{aligned}$$

So if the result is true for y , it will be true for $y+1$. Hence the result is true for all y .

(iii) $A(3, 0) = A(2, 1) = 5 = 2^0 - 3$. So the result is true for $y=0$.

Suppose the result is true for y . Then $A(3, y) = 2^{y+3} - 3$. So

$$\begin{aligned} A(3, y+1) &= A(2+1, y+1) = A(2, A(2+1, y)) = A(2, A(3, y)) \\ &= A(2, 2^{y+3} - 3) = 2((2^{y+3} - 3)+3) - 3 = 2^{(y+1)+3} - 3. \end{aligned}$$

∴ result is true for all y by induction.

(iv) $A(4, 0) = A(3, 1) = 2^4 - 3 = 2^2 - 3$. So result is true for $y=0$.

Suppose result is true for y . Then $A(4, y) = 2^{\frac{y+3}{2}} - 3$, $\frac{(y+3)}{2}$'s.

So $A(4, y+1) = A(3+1, y+1) = A(3, A(4, y)) = \left\{ \begin{array}{l} A(4, y) - 2^{\frac{y+4}{2}} - 3 \\ 2^{\frac{y+4}{2}} - 3 \end{array} \right.$
with $(y+1)+3$ 2's. So result is true for all y . $2^{\frac{y+4}{2}} - 3 = 2^2 - 3$

(11)

84. Recursive functions.

We can define a wider class of functions than the primitive recursive ones by using simultaneous recursion instead of primitive recursion - but all the functions we obtain will be total functions.

In order to also obtain partial functions, we will introduce a new operation called minimization and this will lead us to the same wider class of total functions.

Def. Let $g: \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ be a total function and $n \geq 0$.

Then we can define a partial function $f: \mathbb{N}^n \rightarrow \mathbb{N}$ by putting

$$f(x) = \begin{cases} \text{smallest value of } y \text{ such that } g(x, y) = 0 \\ \text{undefined, if } g(x, y) > 0 \text{ for each } y \in \mathbb{N}. \end{cases}$$

Here x abbreviates (x_1, \dots, x_n) as usual. The function f is said to be obtained from g by minimization and we write $f = \mu(g, 0)$. We also sometimes write $f(x) = (\mu y) [g(x, y) = 0]$.

Ex. 1 Let $g(x, y) = x - 3y$. Then $g: \mathbb{N}^2 \rightarrow \mathbb{N}$ is a total function. If $f = \mu(g, 0)$, then

$$f(4) = (\mu y) [4 - 3y = 0] = 2$$

$$f(8) = (\mu y) [8 - 3y = 0] = 3$$

$$f(9) = (\mu y) [9 - 3y = 0] = 3$$

$$f(25) = (\mu y) [25 - 3y = 0] = 9$$

$$f(30) = (\mu y) [30 - 3y = 0] = 10.$$

In general $f(x) = (\mu y) [x - 3y = 0] = \lceil x/3 \rceil$ where $\lceil z \rceil = \text{smallest integer } \geq z = \text{ceiling function of } z$.

(122)

Ex. 2 Let $g(x,y) = (x-3y) + (3y-x)$. Then $g(x,y) = |x-3y|$.

If $f_i = \mu[g, 0]$, then

$$f(0) = (\mu y)[|0-3y|=0] = 0$$

$$f(1) = (\mu y)[|1-3y|=0] = \text{undefined}$$

$$f(2) = (\mu y)[|2-3y|=0] = \text{undefined}$$

$$f(3) = (\mu y)[|3-3y|=0] = 1.$$

In general $f(x) = \begin{cases} x/3, & \text{if } x \text{ is a multiple of 3} \\ \text{undefined}, & \text{if } x \text{ is not a multiple of 3.} \end{cases}$

Def. A partial function $f: \mathbb{N}^k \rightarrow \mathbb{N}$ is said to be recursive (or μ -recursive) if it can be obtained from the initial functions by a finite number of applications of cartesian products, compositions, primitive recursions and minimization on total functions.

Ex. 1 Let $f_i(x) = \lceil x/2 \rceil = \text{smallest integer } \geq x/2$. Show that f_i is a recursive function.

Sol. Let $g(x,y) = x-2y$. Then $g: \mathbb{N}^2 \rightarrow \mathbb{N}$ is a total function and if we put $f_i = \mu[g, 0]$, then

$$f_i(x) = (\mu y)[g(x,y)=0] = (\mu y)[x-2y=0]$$

= smallest y such that $(x-2y=0) = \lceil x/2 \rceil$

$$\begin{aligned} \text{Now } g(x,y) &= \text{MONUS}(x, 2y) = \text{MONUS}(x, y+4) \\ &= \text{MONUS}(\text{I}_{1,2}(x,y), \text{ADD}(\text{I}_{2,2}(x,y), \text{I}_{2,2}(x,y))) \end{aligned}$$

$$\text{So } f_i = \mu[g, 0]$$

$$= \mu[\text{MONUS} \circ (\text{I}_{1,2} \wedge \text{ADD} \circ (\text{I}_{2,2} \wedge \text{I}_{2,2})), 0].$$

Hence f_i is a recursive function because it is obtained from the initial functions using the 4 operations.

(13)

Ex.2 Let $f_2(x) = \lceil 2x/3 \rceil$. Show that f_2 is a recursive function.

Sol. Let $g(x, y) = 2x - 3y$. Then g is a total function.

$$\text{Also } (\mu y)[g(x, y) = 0] = (\mu y)[2x - 3y = 0]$$

= smallest y such that $(2x - 3y = 0)$

$$= \lceil 2x/3 \rceil = f_2(x)$$

$$\text{Now } g(x, y) = \text{MONUS}(2x, 3y)$$

$$= \text{MONUS}(\text{MULT}(2, x), \text{MULT}(3, y))$$

$$\therefore g = \text{MONUS} \circ (\text{MULT} \circ (\text{SOSOZo} I_{1,2}, I_{1,2}), \text{MULTo}(\text{SOSOZo} I_{1,2}, I_{2,2}))$$

$$\therefore f_2 = \mu[g, 0]$$

$$= \mu[\text{MONUS} \circ (\text{MULT} \circ (\text{SOSOZo} I_{1,2} \wedge I_{1,2}), \text{MULTo}(\text{SOSOZo} I_{1,2} \wedge I_{1,2})), 0]$$

Ex.3 Let $f_3(x) = \lceil \sqrt{x} \rceil$, Show that f is a recursive function.

Sol. Let $g(x, y) = x - y^2$. Then g is a total function. Also

$$(\mu y)[g(x, y) = 0] = (\mu y)[x - y^2 = 0]$$

= smallest y such that $(x - y^2 = 0) = \lceil \sqrt{x} \rceil$.

$$\text{Now } g(x, y) = \text{MONUS}(x, \text{MULT}(y, y))$$

$$= \text{MONUS} \circ [I_{1,2}(x, y), \text{MULT}(I_{2,2}(x, y), I_{2,2}(x, y))]$$

$$\therefore g = \text{MONUS} \circ (I_{1,2} \wedge \text{MULTo}(I_{2,2} \wedge I_{2,2}))$$

$$\therefore f_3 = \mu[g, 0]$$

$$= \mu[\text{MONUS} \circ (I_{1,2} \wedge \text{MULTo}(I_{2,2} \wedge I_{2,2})), 0]$$

Ex.4 Let $f_4(x) = \begin{cases} x/2 & \text{if } x \text{ is an even integer} \\ \text{undefined} & \text{if } x \text{ is an odd integer.} \end{cases}$

Show that f_4 is a recursive partial function.

Sol. Let $g(x, y) = (x - 2y) + (2y - x)$: Then g is a total function and $(\mu y)[g(x, y) = 0] = (\mu y)[(x - 2y) = 0]$

14

Ex.4. = smallest y such that $\{\text{ABS}(x, 2y) = 0\} = f(x)$ b/c.

If x is even, $x/2$ is the only integer for which $|x - 2y| = 0$ and when x is odd, $|x - 2y| \geq 1$, for all y . Now

$$\begin{aligned} g(x, y) &= \text{ABS}(x, 2y) = \text{ABS}(I_{1,2}(x, y), \text{ADD}(I_{2,2}(x, y), I_{2,2}(x, y))) \\ &= \text{ABS} \circ (I_{1,2} \wedge \text{ADD} \circ (I_{2,2} \wedge I_{2,2})) \end{aligned}$$

$$\therefore f_4 = \mu [g, 0]$$

$$= \mu [\text{ABS} \circ (I_{1,2} \wedge \text{ADD} \circ (I_{2,2} \wedge I_{2,2})), 0]$$

Hence f_4 is a recursive partial function because ABS & ADD are primitive recursive total functions.

$$\text{ABS} = \text{ADD} \circ (\text{MONUS} \wedge \text{MONUS} \circ (I_{2,2} \wedge I_{1,2})).$$

Ex.5. Let $f_5(x) = \begin{cases} x^{1/3}, & \text{if } x \text{ is a perfect cube} \\ \text{undefined}, & \text{if } x \text{ is not a perfect cube.} \end{cases}$

Show that f_5 is recursive.

Sol. Let $g(x, y) = |x - y^3| = \text{ABS}(x, y^3)$. Then it is not difficult to see that $(\mu y)[g(x, y) = 0] = f(x)$. So

$$f_5 = \mu [g, 0]$$

$$= \mu [\text{ABS} \circ (I_{1,2} \wedge \text{MULT} \circ (\text{MULT} \circ (I_{2,2} \wedge I_{2,2}) \wedge I_{2,2}), 0)].$$

Hence f_5 is recursive.

Ex.6. Let $h_1(x) = \begin{cases} 0, & \text{if } x \text{ is even} \\ 1, & \text{if } x \text{ is odd} \end{cases}$ Show that f is rec.

Sol. $h_1(x) = 2 \lceil x/2 \rceil - x = \text{MONUS}(2; f(x), x)$ where $f(x) = \lceil x/2 \rceil$. Then

$$h_1 = \text{MONUS} \circ (\text{MULT} \circ (5050 \circ I_{1,1}) \wedge f) \wedge I_{1,1})$$

is a recursive function because f is a recursive function from Ex.1.

(75)

Ex.7 Let $h_2(x) = \begin{cases} 0 & \text{if } x \text{ is a perfect square,} \\ 1 & \text{if } x \text{ is not a perfect square.} \end{cases}$

Also let $h_3(x) = \begin{cases} 1 & \text{if } x \text{ is a perfect square} \\ 0 & \text{if } x \text{ is not a perfect square} \end{cases}$

Show that h_2 & h_3 are recursive functions.

Sol. $h_2(x) = \text{SIGN}((\lceil \sqrt{x} \rceil)^2 - x)$ $\text{SIGN}(z) = \begin{cases} 0 & \text{if } z=0 \\ 1 & \text{if } z>0 \end{cases}$
 $= \text{SIGN}(\text{MONUS}(\text{MULT}(f_3(x), f_3(x)), I_{1,1}(x)))$, where
 f_3 is the recursive function from Example 3. So
 $h_2 = \text{SIGN} \circ \text{MONUS} \circ (\text{MULT} \circ (f_3 \wedge f_3) \wedge I_{1,1})$ is
 a recursive function.

$h_3(x) = 1 - h_2(x)$. So $h_3 = \text{MONUS} \circ (S_0 \circ I_{1,1} \wedge h_2)$
 is a recursive function.

Ex.8 Let $h_4(x) = \lfloor \sqrt{x} \rfloor = \text{largest integer } \leq \sqrt{x}$. Show
 that h_4 is a recursive function.

Sol. Recall that $|z| = \begin{cases} \lceil z \rceil & \text{if } z \in N, \\ \lceil z \rceil - 1 & \text{if } z \notin N. \end{cases}$

So $h_4(x) = \lfloor \sqrt{x} \rfloor = \begin{cases} \lceil \sqrt{x} \rceil - 0 & \text{if } \sqrt{x} \in N, \\ \lceil \sqrt{x} \rceil - 1 & \text{if } \sqrt{x} \notin N \end{cases}$

$= \lceil \sqrt{x} \rceil - h_2(x)$: where h_2 is from Ex.7

$= f_3(x) - h_2(x)$ where f_3 is from Ex.3

$\therefore h_4 = \text{MONUS} \circ (f_3 \wedge h_2)$ and so is
 a recursive function.

Remember $\text{ABS}(x,y) = (x-y)+(y-x)$ & $\text{SIGN}(x) = \begin{cases} 0 & \text{if } x=0, \\ 1 & \text{if } x>0. \end{cases}$
 $= |x-y|$

(16.)

§5.

Recursive & Semi-recursive relations.

Def.

Let $R \subseteq N^n$ be an n -ary relation on N . We define the characteristic function χ_R of R by

$$\chi_R(x) = \begin{cases} 1 & \text{if } x \in R, \\ 0 & \text{if } x \notin R. \end{cases} \quad \text{Here } x = (x_1, \dots, x_n)$$

We define the affirmative function α_R of R by

$$\alpha_R(x) = \begin{cases} 1 & \text{if } x \in R \\ \text{undefined,} & \text{if } x \notin R. \end{cases}$$

Def.

The n -ary relation $R \subseteq N^n$ is a recursive relation if its characteristic function, χ_R , is a recursive function. R is said to be a semi-recursive relation if its affirmative function, α_R , is a recursive function.

Ex. 1

Let R be the 1-ary relation defined by $x \in R$ if x is odd. Then $\chi_R(x) = \begin{cases} 1 & \text{if } x \text{ is odd} \\ 0 & \text{if } x \text{ is not odd.} \end{cases}$

Then χ_R is a recursive function because $\chi_R = h_1$ from Ex. 6 of the previous section. So R is a recursive relation.

Ex. 2

Let R be the binary relation defined by $(x_1, x_2) \in R$ if $x_1 > x_2$. Then $\chi_R(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 > x_2 \\ 0 & \text{if } x_1 \leq x_2 \end{cases}$

$$= \text{SIGN}(x_1 - x_2) = \text{SIGN}(\text{MONUS}(x_1, x_2)). \quad \text{So}$$

$\chi_R = \text{SIGN} \circ \text{MONUS}$ is a recursive function.

Hence R is a recursive relation.

77

Proposition 3: If $A \& B$ are n -ary recursive relations then so are (a) A^c (b) $A \cap B$ (c) $A \cup B$.

Proof: Suppose $A \& B$ are recursive relations. Then χ_A and χ_B will be recursive functions.

$$(a) \text{ Now } \chi_{A^c}(x) = \begin{cases} 1 & \text{if } x \in A^c \\ 0 & \text{if } x \notin A^c \end{cases} = \begin{cases} 0 & \text{if } x \in A \\ 1 & \text{if } x \notin A \end{cases}$$

So $\chi_{A^c}(x) = 1 - \chi_A(x)$. Hence $\chi_{A^c} = \text{MONUS} \circ (S_0 Z_0 I_{1,n} \wedge \chi_A)$ is a recursive function. So A^c is a recursive relation.

$$(b) \text{ Also } \chi_{A \cap B} = \begin{cases} 1 & \text{if } x \in A \cap B \\ 0 & \text{if } x \notin A \cap B \end{cases} = \chi_A(x), \chi_B(x)$$

So $\chi_{A \cap B} = \text{MULT} \circ (\chi_A \wedge \chi_B)$ is a recursive function. Hence $A \cap B$ is a recursive relation.

$$(c) \chi_{A \cup B}(x) = \begin{cases} 1 & \text{if } x \in A \cup B \\ 0 & \text{if } x \notin A \cup B \end{cases} = 1 - \text{MULT}(1 - \chi_A(x), 1 - \chi_B(x))$$

$$\text{So } \chi_{A \cup B} = \text{MONUS} \circ (S_0 Z_0 I_{1,n} \wedge \text{MULT} \circ (\text{MONUS} \circ (S_0 Z_0 I_{1,n} \wedge \chi_A) \wedge \text{MONUS} \circ (S_0 Z_0 I_{1,n} \wedge \chi_B)))$$

is a recursive function. Hence $A \cup B$ is a recursive relation. We can also use $\chi_{A \cup B}(x) = \chi_A(x) + \chi_B(x) = \chi_A(x) \cdot 1^{\chi_B(x)}$ to get a shorter expression for $\chi_{A \cup B}$.

Proposition 4: If $A \& B$ are n -ary semi-recursive relations then so are (a) $A \cap B$ (b) $A \cup B$.

Proof: Suppose $A \& B$ are semi-recursive relations. Then α_A and α_B will be recursive functions.

(18)

Now it can be shown that a function f is recursive if and only if we can find a recursive total function g such that $f = \mu[g, 0]$. Since α_A and α_B are recursive functions, this means that we can find recursive total functions g_A and g_B such that $\mu[g_A, 0] = \alpha_A$ and $\mu[g_B, 0] = \alpha_B$.

(a) So $\alpha_{A \cap B}(x) = \begin{cases} 1 & \text{if } x \in A \cap B \\ \text{undefined, if } x \notin A \cap B \end{cases}$

$$= (\forall y) [g_A(x, y) \cdot g_B(x, y) = 0]$$

$$\therefore \alpha_{A \cap B} = \mu [\text{ADD} \circ (g_A \wedge g_B), 0] \text{ Hence}$$

$\alpha_{A \cap B}$ is a recursive function. So $A \cap B$ is a semi-recursive n -ary relation.

(b) Also $\alpha_{A \cup B}(x) = \begin{cases} 1 & \text{if } x \in A \cup B \\ \text{undefined, if } x \notin A \cup B \end{cases}$

$$= (\forall y) [g_A(x, y) + g_B(x, y) = 0]$$

$$\therefore \alpha_{A \cup B} = \mu [\text{MULT} \circ (g_A \wedge g_B), 0] \text{ Hence } \alpha_{A \cup B}$$

is a recursive function. So $A \cup B$ is a semi-recursive n -ary relation.

Note If A is a semi-recursive n -ary relation, it does not always follow that A^c is a semi-recursive n -ary relation. For example, let R be the binary relation on \mathbb{N} defined by

$C(M) R C(w) \iff \text{the TM } M \text{ halts on the input } w$

Here $C(M)$ & $C(w)$ are codings of M & w into natural numbers.

Then R is a semi-recursive relation which is not recursive. This forces R^c to be a non-semi-recursive relation.

19

Proposition 5. If $A \& A^c$ are both semi-recursive n -ary relations, then A must be a recursive relation.

Proof: Suppose $A \& A^c$ are semi-recursive relations. Then α_A and α_{A^c} will be recursive partial functions. So as in the proof of Prop 4, we can find recursive total functions g and h such that $\alpha_A = \mu[g, 0]$ and $\alpha_{A^c} = \mu[h, 0]$.

$$\begin{aligned} \text{Now } \chi_A(x) &= \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \in A^c \end{cases} \\ &= \begin{cases} S(y) [g(x, y) = 0] & \text{if } x \in A \\ (y) [h(x, y) = 0] - 1 & \text{if } x \in A^c \end{cases} \\ &= \begin{cases} S(y) [g(x, y) = 0] & \text{if } x \in A \\ (y) [h(x, y+1) = 0] & \text{if } x \in A^c \end{cases} \\ &= (y) [\text{MIN}\{g(x, y), h(x, y+1)\} = 0] \end{aligned}$$

But $\text{MIN}(x, y) = x \dot{-} (x \dot{-} y)$. So

$\text{MIN} = \text{MONUS} \circ (I_{1,2} \wedge \text{MONUS})$ and hence is a primitive recursive function. Thus

$\chi_A = \mu[\text{MIN} \circ (g \wedge h \circ (I_{1,n+1} \wedge \dots \wedge I_{n,n+1} \wedge \text{SI}_{n+1,n+1}), 0)]$ is a recursive total function. Hence A is a recursive n -ary relation.

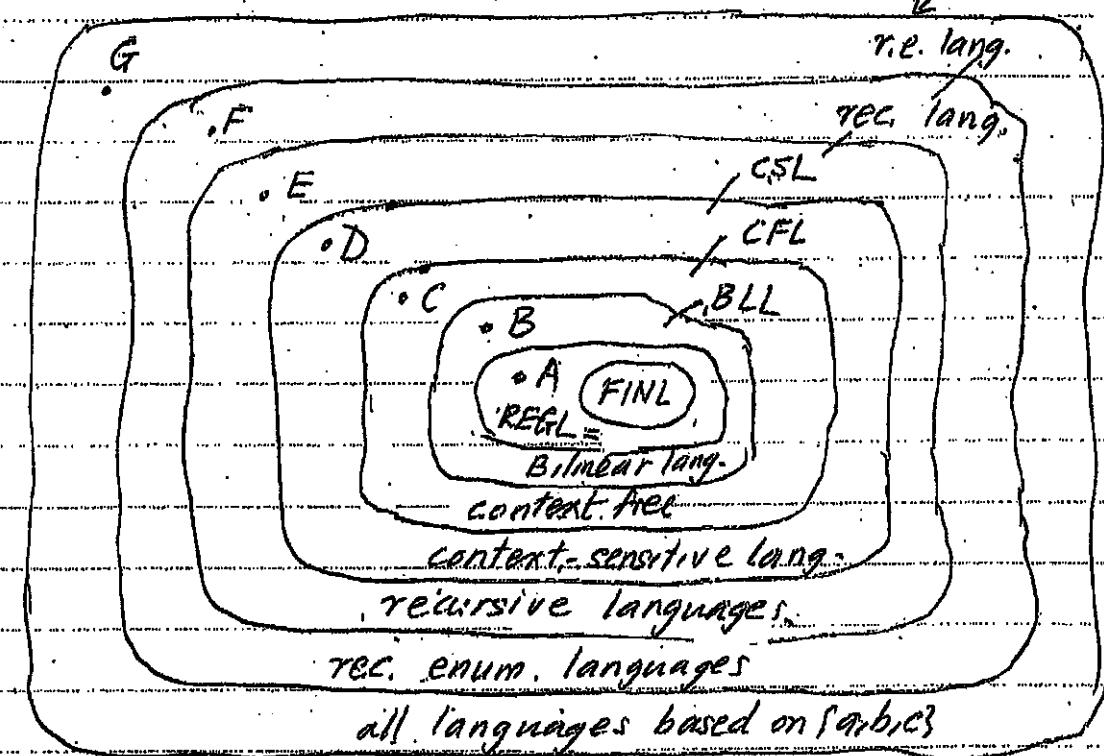
Def. A language $L \subseteq T^*$ is said to be recursively enumerable (r.e.) if $c(L) = \emptyset$ or $c(L) = \{g(n) : n \in N\}$ for some recursive total function $g : N \rightarrow N$. Here $c(L)$ is a coding of the strings of L into natural numbers.

26

It can be shown that $c(L)$ is a recursively enumerable unary relation $\Leftrightarrow c(L)$ is a semi-recursive relation.
So we have the following correspondences.

| MACHINES | EXPRESSIONS |
|------------------------------------|---|
| 1. Turing computable function. | Recursive function |
| 2(a) Turing decidable relation | Recursive relation |
| (b) Turing semi-decidable relation | Semi-recursive relation |
| 3(a) Turing decidable language | Recursive language |
| (b) Turing semi-decidable language | Semi-recursive language Recursively enumerable lang. |

A hierarchy of languages:



$$A = \{a^n : n \geq 0\}, \quad B = \{a^n b^n : n \geq 0\}, \quad C = \{a^n b^m a^k b^k : n, k \geq 1\}$$

$$D = \{a^n b^n c^n : n \geq 1\}, \quad E \text{ is too complicated to describe}$$

$$F = \{w \in \{a, b, c\}^*: M_u \text{ halts on } w\}, \quad G = F^c$$

Here M_u = a fixed universal Turing Machine

End of Ch 8

Example 1: Let $\text{EXP}(x,y) = x^y$. Show that EXP is a primitive recursive function.
[You may use the fact that MULT, ADD, MONUS, PRED, & SIGN
are primitive recursive functions, if needed.]

Solution: We will find primitive recursive functions g & h such that $\text{EXP} = \text{prec}(g, h)$.
We have $\text{EXP}(x,0) = x^0 = 1$, [So, $g(x) = 1$, a function of 1 variable.]
& $\text{EXP}(x, s(y)) = x^{y+1} = x \cdot x^y = x \cdot \text{EXP}(x,y)$. [So, $h(x, y, \text{EXP}(x,y)) = x \cdot \text{EXP}(x,y)$]
Hence $g = s \circ I_{1,1}$ and $h = \text{MULT} \circ (I_{1,3} \wedge I_{3,3})$.
Thus $\text{EXP} = \text{prec}(g, h) = \text{prec}(s \circ I_{1,1}, \text{MULT} \circ (I_{1,3} \wedge I_{3,3}))$.
So, EXP is a primitive recursive function.

Example 2: Let $F(x,y) = (x+3)^{2+y}$. Show that F is a primitive recursive function.
[You may use the fact that MULT, ADD, MONUS, PRED, & SIGN
are primitive recursive functions, if needed.]

Solution: We will find primitive recursive functions g & h such that $F = \text{prec}(g, h)$.
We have $F(x,0) = (x+3)^{2+0} = (x+3)^2 = (x+3) \cdot (x+3)$, [So, $g(x) = (x+3) \cdot (x+3)$]
& $F(x, s(y)) = (x+3)^{2+y+1} = (x+3) \cdot (x+3)^{2+y} = (x+3) \cdot F(x,y)$. [So $h(x, y, F(x,y)) = (x+3) \cdot F(x,y)$]
Hence $g = \text{MULT} \circ (s \circ s \circ s \circ I_{1,1} \wedge s \circ s \circ s \circ I_{1,1})$ and $h = \text{MULT} \circ (s \circ s \circ s \circ I_{1,3} \wedge I_{3,3})$.
Thus $F = \text{prec}(g, h) = \text{prec}(\text{MULT} \circ (s \circ s \circ s \circ I_{1,1} \wedge s \circ s \circ s \circ I_{1,1}), \text{MULT} \circ (s \circ s \circ s \circ I_{1,3} \wedge I_{3,3}))$.
So, F is a primitive recursive function.

f

Example 3: Let $f(x,y) = y^x$. Show that ~~EXP~~ f is a primitive recursive function.

Solution: We have $f(x, y) = \text{EXP}(y, x) = \text{EXP}\{I_{2,2}(x, y), I_{1,2}(x, y)\}$.
So, $f = \text{EXP} \circ (I_{2,2} \wedge I_{1,2}) = \{\text{prec}(s \circ I_{1,1}, \text{MULT} \circ (I_{1,3} \wedge I_{3,3}))\} \circ (I_{2,2} \wedge I_{1,2})$ from Ex. 1.
Hence f is a primitive recursive function. END