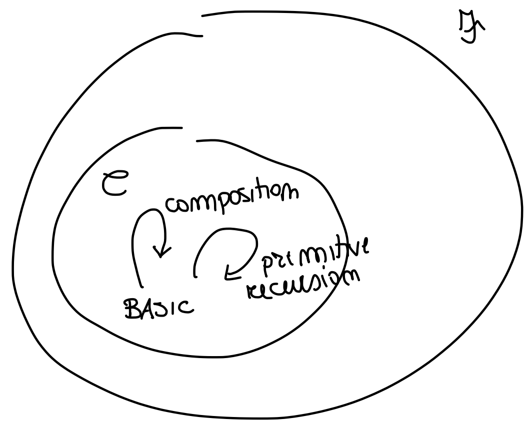
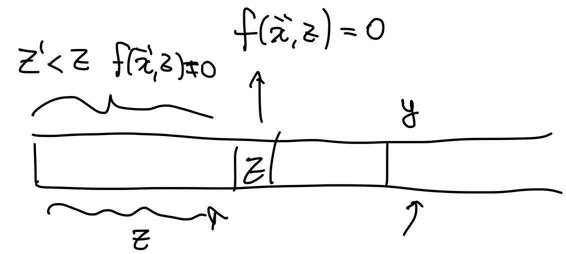
Computability (19/10/2021)



- → arithmetic functions
- → bounded sum/product

⋮

- → Bounded minimalisation



Given $f: \mathbb{N}^{k+1} \to \mathbb{N}$ total computable

define $h: \mathbb{N}^{k+1} \to \mathbb{N}$

$$h(\vec{x},y) = \mu z < y . \underset{\uparrow}{f(\vec{x},z)} = \begin{cases} \text{minimum } z<y \text{ such that } f(\vec{x},z)=0 & \text{if it exists} \\ y \text{ otherwise} \end{cases}$$

$(f(\vec{x},z)=0)$

$$= \min\left(\{ z \mid z < y \wedge \forall z'<z . f(\vec{x},z') \neq 0 \wedge f(\vec{x},z)=0 \} \cup \{y\} \right)$$

Then h is total computable

proof

$$\begin{cases} h(\vec{x},0) = 0 \\ h(\vec{x},y+1) = \begin{cases} \text{if } h(\vec{x},y) < y & \rightsquigarrow h(\vec{x},y) \\ \text{if } h(\vec{x},y) = y & \rightsquigarrow 2 \text{ subcases} \end{cases} \end{cases}$$

$sg(\ )=0$

$f(\vec{x},y)=0 \rightsquigarrow h(\vec{x},y+1)=y$

$f(\vec{x},y)\neq 0 \rightsquigarrow h(\vec{x},y+1)=y+1$

$sg(\ )=1$

$$= \boxed{h(\vec{x},y)} \cdot sg(y \dot{-} \boxed{h(\vec{x},y)}) + (y + sg(f(\vec{x},y))) \cdot \overline{sg}(y \dot{-} \boxed{h(\vec{x},y)})$$

computable by primitive recursion.

$\begin{cases} 1 & \text{if } y \text{ is divisor of } x \\ 0 & \text{otherwise} \end{cases}$

$*$ $D(x) = \#$ divisors of $x = \sum_{y \leq x} \overline{sg}(rm(y,x))$ computable

* $Pr(x) = \begin{cases} 1 & x \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$

x is prime if it only divisors are x and 1

and $x \neq 1$

$\Leftrightarrow$ x has exactly two divisors

NOTE

$$|x - y| = (x \dot{-} y) + (y \dot{-} x)$$

$$Pr(x) = \overline{sg}\left(\left|D(x) - 2\right|\right)$$

* p_x = x-th prime number

$p_0 = 0$ $\quad p_1 = 2 \quad p_2 = 3 \quad p_3 = 5 \quad p_4 = 7 \quad p_5 = 11 \quad \dots$

$$\left.\begin{array}{l} p_0 = 0 \\ p_{x+1} = \mu z \underbrace{(z \text{ is prime and } z > p_x)} \end{array}\right\}$$

$$\mu z \underset{?}{\leq} \left(\prod_{i=1}^{x} p_i\right)+1 \, . \, \overline{sg}\left(Pr(x) \cdot sg(z \dot{-} p_x)\right)$$

$$p_{x+1} \leq \underbrace{\left(\prod_{i=1}^{x} p_i\right) + 1}$$

let p be a prime such that

p divides $\underbrace{\left(\prod_{i=1}^{x} p_i\right)} + 1$

$p > p_x$

in fact if $p \leq p_x$ $\leadsto$ $\exists i \leq x$ st. $p_i = p$

$\Rightarrow$ $p = p_i$ divides $\left(\prod_{i=1}^{x} p_i\right)$

$\Rightarrow$ p divides 1 $\Rightarrow$ $p = 1$ ABSURD.

$$p_x < p \leq \left(\prod_{i=1}^{x} p_i\right) + 1$$

$\uparrow$ p divides $\left(\prod_{i=1}^{x} p_i\right) + 1$

* $(x)_y$ = exponent of p_y in the prime factorisation of x

$(4)_1$ = exp. of $p_1 = 2$ in 2^2

$= 2$

$(4)_2$ = exp. of $p_2 = 3$ in $2^2 \cdot 3^0$

$= 0$

= maximum exponent z such that p_y^z divides x

$= \max z \; (p_y^z \text{ divides } x)$

$= \min z \; \underbrace{p_y^{z+1} \text{ does not divide } x}$

$= \mu z \leq x \; . \; \text{div}(p_y^{z+1}, x)$ computable

$\underbrace{(x_1, x_2, \dots x_m)} \rightsquigarrow z$

$\ll p_1^{x_1} \cdot p_2^{x_2} \dots p_m^{x_m}$

$$x_i = (z)_i$$

* <u>Fibonacci</u>

$$\begin{cases} f(0) = 1 \\ f(1) = 1 \\ f(m+2) = f(m) + f(m+1) \end{cases}$$

not a def. by primitive recursion

$g : \mathbb{N} \to \mathbb{N}^2$

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

$D = \mathbb{N}^2$

$\pi : \mathbb{N}^2 \to \mathbb{N}$ bijective "effective" & π^{-1} effective computable

$\pi(x,y) = 2^x \cdot (2y+1) - 1$

$\pi^{-1} : \mathbb{N} \to \mathbb{N}^2$

$\pi^{-1}(n) = (\pi_1(n), \pi_2(n))$ $\qquad \pi_1, \pi_2 : \mathbb{N} \to \mathbb{N}$

$$n = \underbrace{2^x (2y+1)}_{} - 1$$

$\pi_1(n) = (n+1)_1$

$\pi_2(n) = \left(\frac{n+1}{2^{\pi_1(n)}} - 1\right) / 2$

$= qt(2, (qt(2^{\pi_1(n)}, n+1) \dot{-} 1))$

$g : \mathbb{N} \to \mathbb{N}$

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

primitive recursion

$$\begin{cases} g(0) = \pi(f(0), f(1)) = \pi(1,1) = 2^1(2 \cdot 1 + 1) - 1 = 5 \\ g(n+1) = \pi(\underbrace{f(n+1)}_{\pi_2(g(n))}, \underbrace{f(n+2)}_{f(n)+f(n+1)}) = \pi(\pi_2(g(n)), \pi_1(g(n)) + \pi_2(g(n))) \end{cases}$$

$= \pi_1(g(n)) + \pi_2(g(n))$

g computable

$f(n) = \pi_1(g(n))$ computable

<u>EXERCISE</u>: All function which are obtained from the basic functions using composition & primitive recursion are <u>total</u>.

* Unbounded minimalisation

Given $f : \mathbb{N}^{k+1} \to \mathbb{N}$ (not necessarily total)

$f(\vec{x}, y)$

define $h : \mathbb{N}^k \to \mathbb{N}$

$h(\vec{x}) = \mu y.\ f(\vec{x}, y) =$ least y such that $f(\vec{x}, y) = 0$

- → such y could not exist
- → f could be undefined on $y' < y$ where y is the least "zero"

↓

in these cases $h(\vec{x}, y)$ is undefined

$$= \begin{cases} y & \text{if there is a } y \text{ such that } \forall z < y \ f(\vec{x}, z)\downarrow \ f(\vec{x}, z) \neq 0 \text{ and } f(\vec{x}, y) = 0 \\ \uparrow & \text{if there is no such } y \end{cases}$$

in order to compute $h(\vec{x}, y)$

$f(\vec{x}, 0) = 0?$ → out 0

$f(\vec{x}, 1) = 0?$ → out 1

⋮

Proposition: $\mathcal{C}$ is closed under minimalisation

proof Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ computable

we want to show that $h : \mathbb{N}^k \to \mathbb{N}$ $\in \mathcal{C}$

$h(\vec{x}) = \mu y.\ f(\vec{x}, y)$

Let F be program in standard form for f

1		K		m	m+1		m+K	m+K+1	m+K+2
x_1	...	x_K			x_1	..	x_K	i	0

$$\underbrace{x_1 \; .. \; x_K \; i}_{f}$$

$m = \max\{\rho(F), K\}$

$f(\vec{x}, 0)$ $\quad i{+}{+}$

$f(\vec{x}, 1)$ $\quad \vdots$

the program H for function h is

$T(1, m+1)$ // save $\vec{x}$

$\vdots$

$T(K, m+K)$

LOOP : $F[m+1, \ldots, m+K \to 1]$ // compute $f(\vec{x}, i)$

$J(1, m+K+2, \text{END})$ // $f(\vec{x}, i) = 0$?

$S(m+K+1)$ // NO $\to$ $i = i+1$

$J(1, 1, \text{LOOP})$

END: $T(m+K+1, 1)$

<u>EXAMPLE</u> :

$$f(x) = \begin{cases} \sqrt{x} & \text{if } x \text{ is perfect square} \\ \uparrow & \text{otherwise} \end{cases}$$

$$f(x) = \mu y. \text{ `` } y * y = x \text{ ''}$$

$$= \mu y. \; |y * y - x|$$

$$g(x, y) = \begin{cases} x/y & \text{if } \overbrace{y \neq 0} \text{ and } y \text{ divides } x \\ \uparrow & \text{otherwise} \end{cases}$$

$$= \mu z. \left(|z * y - x| + \underbrace{\overline{sg}(y)}_{\substack{0 \quad y>0 \\ 1 \quad y=0}} \right)$$