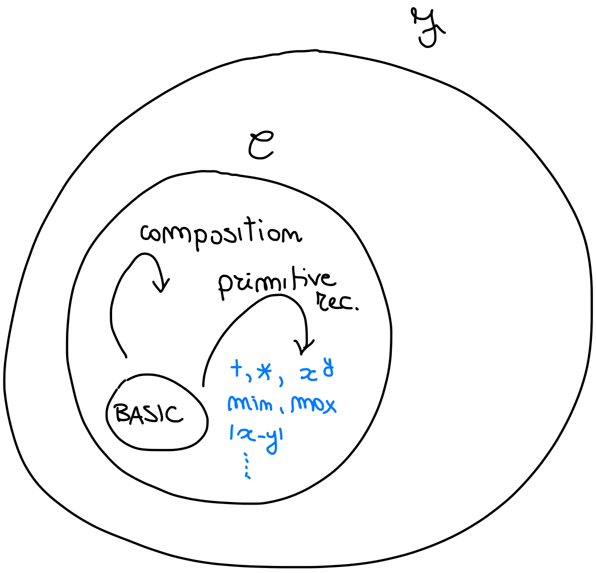
COMPUTABILITY (23/10/2023)

Class $\mathcal{C}$ of URM-computable functions



* contains the BASIC FUNCTIONS

  (a) zero
  (b) successor
  (c) projections

* closed under

  (1) (generalised) composition
  (2) primitive recursion
  (3) (unbounded) minimalisation

* OBSERVATION : Definition by cases

Let $f_1, \dots, f_n : \mathbb{N}^k \to \mathbb{N}$ functions computable total

$Q_1(\vec{x}), \dots, Q_n(\vec{x}) \subseteq \mathbb{N}^k$ decidable predicates $\quad \forall \vec{x} \ \exists! j \ Q_j(\vec{x})$

and let $f : \mathbb{N}^k \to \mathbb{N}$

$$f(\vec{x}) = \begin{cases} f_1(\vec{x}) & \text{if } Q_1(\vec{x}) \\ \vdots & \\ f_n(\vec{x}) & \text{if } Q_n(\vec{x}) \end{cases}$$

Then f is computable and total

proof

computable

$$f(\vec{x}) = f_1(\vec{x}) \cdot \chi_{Q_1}(\vec{x}) + f_2(\vec{x}) \cdot \chi_{Q_2}(\vec{x}) + \dots + f_n(\vec{x}) \cdot \chi_{Q_n}(\vec{x})$$

computable by hyp.

computable since it is the composition of computable functions
total ... total $\square$

Note: $n=2$ $\quad f_1(x) = x \ \forall x \quad$ computable $\quad Q_1(x) = \text{true} \ \forall x$
$\qquad f_2(x)\uparrow \ \forall x \qquad\qquad\qquad Q_2(x) = \text{false} \ \forall x$

$$f(x) = \begin{cases} f_1(x) & \text{if } \overset{\text{true}}{\overline{Q_1(x)}} \\ f_2(x) & \text{if } \underset{\text{false}}{\underline{Q_2(x)}} \end{cases} = f_1(x) = x \quad \forall x$$

$$\neq \quad \underbrace{f_1(x)\cdot\chi_{Q_1}(x) + \underbrace{f_2(x)}_{\uparrow\,\forall x}\cdot\chi_{Q_2}(x)}_{\uparrow\,\forall x}$$

* <u>Algebra of decidability</u>

Let $Q_1(\vec{x}), Q_2(\vec{x}) \subseteq \mathbb{N}^k$ be decidable predicates. Then

1) $\neg Q_1(\vec{x})$

2) $Q_1(\vec{x}) \wedge Q_2(\vec{x})$ decidable

3) $Q_1(\vec{x}) \vee Q_2(\vec{x})$

<u>proof</u>

① $$\chi_{\neg Q_1}(\vec{x}) = \begin{cases} 1 & \text{if } \underbrace{\neg Q_1(\vec{x})}_{\chi_{Q_1}(\vec{x})=0} \\ 0 & \text{if } \underbrace{Q_1(\vec{x})}_{\chi_{Q_1}(\vec{x})=1} \end{cases} = \overline{sg}(\chi_{Q_1}(\vec{x}))$$

computable $\leadsto$ $\chi_{\neg Q_1}$ computable

② $$\chi_{Q_1\wedge Q_2}(\vec{x}) = \chi_{Q_1}(\vec{x}) \cdot \chi_{Q_2}(\vec{x})$$

③ $$\chi_{Q_1\vee Q_2}(\vec{x}) = sg\left(\chi_{Q_1}(\vec{x}) + \chi_{Q_2}(\vec{x})\right)$$

* <u>Bounded Sum / Product</u>

$f(\vec{x},z)$ $\quad f: \mathbb{N}^{k+1} \to \mathbb{N}$ total computable

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

$$h(\vec{x},y) = f(\vec{x},0) + f(\vec{x},1) + \dots + f(\vec{x},y-1)$$
$$= \sum_{z<y} f(\vec{x},z)$$

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

primitive recursion of computable functions

* <u>Product</u> $\prod_{z<y} f(\vec{x},z)$

$$\begin{cases} \prod_{z<0} f(\vec{x},z) = 1 \\ \prod_{z<y+1} f(\vec{x},z) = \left(\prod_{z<y} f(\vec{x},z)\right) * f(\vec{x},y) \end{cases}$$

* **Bounded Quantification**

$Q(\vec{x}, z)$ decidable

① $\forall z<y.\ Q(\vec{x},z)$

② $\exists z<y.\ Q(\vec{x},z)$

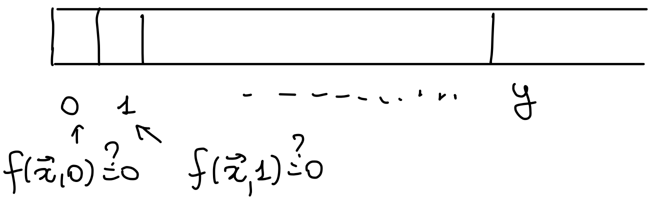
decidable [EXERCISE]

* **Bounded minimalisation**

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

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

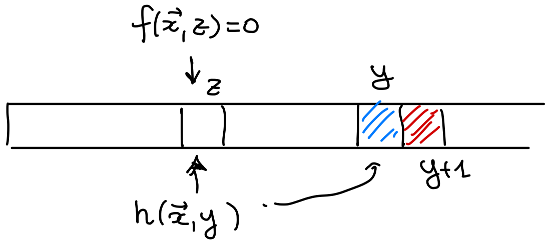
$$h(\vec{x},y) = \begin{cases} z & \text{minimum } z<y \text{ s.t. } f(\vec{x},z)=0 \\ y & \text{if there is no such } z \end{cases}$$



OBSERVATION : If f is computable then $h(\vec{x},y) = \mu z<y.\ f(\vec{x},z) \cancel{=0}$ computable

proof

definition by primitive recursion



$$\begin{cases} h(\vec{x},0) = 0 \\ h(\vec{x},y+1) = \begin{cases} \text{if } h(\vec{x},y) < y & \leadsto h(\vec{x},y) \\ \text{if } h(\vec{x},z) = y & \leadsto \begin{cases} \text{if } f(\vec{x},y) = 0 & \leadsto y \\ \text{if } f(\vec{x},y) \neq 0 & \leadsto y+1 \end{cases} \end{cases} \end{cases}$$

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

1 if $h(\vec{x},y) < y$
0 otherwise

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

computable by primitive recursion □

OBSERVATION: The following functions are computable

* $div : \mathbb{N}^2 \to \mathbb{N}$

$$div(x,y) = \begin{cases} 1 & \text{if } x \text{ divides } y \\ 0 & \text{otherwise} \end{cases}$$

$$= \overline{sg}(\,rm(x,y))$$

* $D(x)$ = number of divisors of x

$$= \sum_{y \le x} div(y,x)$$

$$\wr$$

$$y < x+1$$

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

x is prime iff the only divisors of x are x and 1 and $x \neq 1$

$$\Updownarrow$$

x has exactly 2 divisors

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

how do we compute

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

* p_x = x^{th} prime number

$p_0 = 0 \qquad p_1 = 2 \qquad p_2 = 3 \qquad p_3 = 5 \qquad p_4 = 7 \qquad \ldots$

by primitive recursion

$$\begin{cases} p_0 = 0 \\ p_{x+1} = \text{"" } \mu z \quad z \text{ prime and } z > p_x \text{ ""} \end{cases}$$

$$Pr(z) \wedge z > p_x$$

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

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

let p a prime divisor of

then $\quad p \neq p_i \quad \forall\, i = 1 \dots x$

otherwise if $p = p_J$ for $J \leq x$ then $p \mid \prod_{i=1}^{x} p_i$ divides

but $\quad p \mid \left(\prod_{i=1}^{x} p_i\right) + 1$

$\rightsquigarrow p \mid 1 \rightsquigarrow p = 1$ not prime

$\Rightarrow \quad p \geq p_{x+1}$

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

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

$20 = 2^2\, 3^0\, 5^1$

$(20)_1$ = exponent of $p_1 = 2 \rightsquigarrow (20)_1 = 2$

$(20)_2$ = ... " $p_2 = 3 \rightsquigarrow (20)_2 = 0$

$(20)_3 = 1$

$(20)_4 = 0$

$\vdots$

$(x)_y =$ max z s.t. p_y^z divides x

$=$ max z s.t. $div(p_y^z, x) = 1$

$=$ min z s.t. $div(p_y^{z+1}, x) = 0$

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

$\square$

EXERCISE: All functions obtained from the basic functions using composition and primitive recursion are total.

* Fibonacci

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

not exactly a primitive recursion

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

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

$D = \mathbb{N}^2$

$\pi : \mathbb{N}^2 \to \mathbb{N}$ bijective "effective" and

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

$\pi(x, y) = 2^x (2y+1) \dot{-} 1$ computable

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

$\pi^{-1}(n) = (\pi_1(n), \pi_2(n))$

$\pi_1, \pi_2 : \mathbb{N} \to \mathbb{N}$

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

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

$n+1 = 2^x (2y+1)$

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

π_1, π_2 computable

π^{-1} "effective"

$$\begin{cases} g : \mathbb{N} \to \mathbb{N} \\ g(n) = \pi(f(n), f(n+1)) \end{cases}$$

by primitive recursion

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

$$= \pi\big(\pi_2(g(n)),\, \pi_1(g(n)) + \pi_2(g(n))\big)$$

g computable

$\Downarrow$

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

$\square$