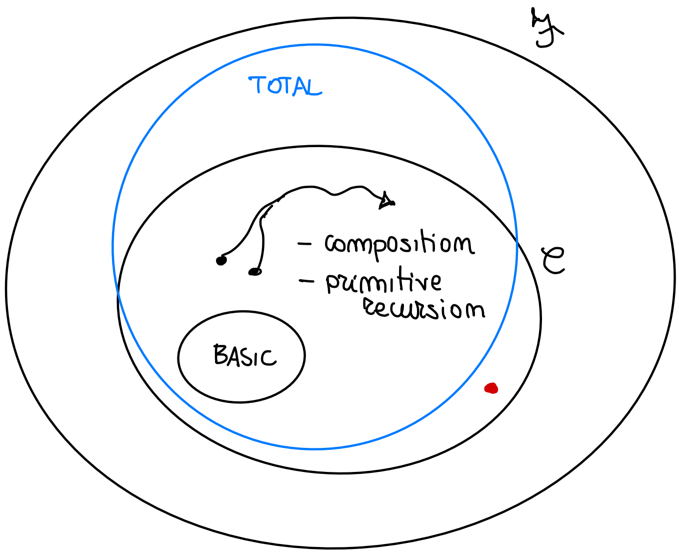
COMPUTABILITY (27/10/2025)



* Unbounded Minimalisation

Given $f : \mathbb{N}^{k+1} \to \mathbb{N}$ (partial)

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

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

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

= " least y s.t. $f(\vec{x}, y) = 0$ "

two problems :
- such a y st. $f(\vec{x}, y) = 0$ might not exist
- $f(\vec{x}, z)$ could be $\uparrow$ for $z < y$ s.t. $f(\vec{x}, y) = 0$

$$= \begin{cases} y & \text{if } y \text{ is such that } f(\vec{x},y) = 0 \text{ and } \forall z < y\ f(\vec{x}, z)\downarrow \neq 0 \\ \uparrow & \text{otherwise} \end{cases}$$

to compute $\mu y.\ f(\vec{x}, y)$

$f(\vec{x}, 0) = 0$? $\xrightarrow{yes}$ out 0

$\xrightarrow{NO}$

$f(\vec{x}, 1) = 0$? $\xrightarrow{yes}$ out 1

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

$\vdots$

Example :

$$f(x) = \begin{cases} x \dot{-} 1 & \text{if } x > 0 \\ \uparrow & \text{if } x = 0 \end{cases}$$

$\mu x.\ |f(x) - y| \uparrow$

not $y+1$ since $f(0)\uparrow$

Proposition : Class $\mathcal{C}$ is closed under (unbounded) minimalisation

proof

Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ in $\mathcal{C}$

and let P a program for f (in standard form) and we construct a program for $h(\vec{x}) = \mu y. f(\vec{x}, y)$

1		k		m+1		m+k	m+k+1	m+k+2	
x_1	...	x_k		x_1	...	x_k	i	0	

$m = \max\{\rho(P), k+1\}$

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

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

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

$S(m+k+1)$ // $i = i+1$

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

END: $T(m+k+1, 1)$ // output i

$\square$

<u>Example</u>: $f : \mathbb{N} \to \mathbb{N}$

$$f(x) = \begin{cases} x/2 & x \text{ even} \\ \uparrow & \text{otherwise} \end{cases}$$

0 if $x = 2y$

$$f(x) = \mu y . |x - 2y|$$
$$= \mu y . |x - (y+y)|$$ computable

<u>Example</u>: $g : \mathbb{N}^2 \to \mathbb{N}$

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

$\times$ $\mu z . |z \times y - x|$

$y = 0,\ x = 0$ we get 0

we want $\uparrow$

$$= \mu z . (|z \times y - x| + \overline{sg}(y))$$

OBSERVATION: Every finite (domain) function is computable

proof

Let $\vartheta : \mathbb{N} \to \mathbb{N}$ be a finite (domain) function

$$\vartheta(x) = \begin{cases} y_1 & \text{if } x = x_1 \\ y_2 & \text{if } x = x_2 \\ \vdots & \\ y_m & \text{if } x = x_m \\ \uparrow & \text{otherwise} \end{cases} \qquad dom(\vartheta) = \{x_1, .., x_m\}$$

$$\vartheta = \{(x_1, y_1), \dots, (x_m, y_m)\}$$

it is computable

$$\vartheta(x) = \underbrace{\sum_{i=1}^{m} y_i \cdot \underbrace{\overline{sg}(|x - x_i|)}_{\begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if } x \neq x_i \end{cases}}}_{\begin{cases} y_i & \text{if } x = x_i \\ 0 & \text{otherwise} \end{cases}} + \overbrace{\mu z . \underbrace{\prod_{i=1}^{m} |x - x_i|}_{\begin{cases} 0 & \text{if } x \in dom(\vartheta) \\ \neq 0 & \text{otherwise} \end{cases}}}^{\begin{cases} 0 & \text{if } x \in dom(\vartheta) \\ \uparrow & \text{otherwise} \end{cases}}$$

Example

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

$$f(x) = \begin{cases} 1 & \text{if } x = 0 \text{ and } P = NP \\ 0 & \text{if } x = 0 \text{ '' } P \neq NP \\ \uparrow & \text{otherwise} \end{cases} \qquad \text{computable} \quad (dom(f) = \{0\})$$

$g : \mathbb{N} \to \mathbb{N}$, fixed program P

$$g(x) = \begin{cases} 0 & \text{if } x = 0 \text{ and } P(0)\downarrow \\ 1 & \text{if } x = 0 \text{ and } P(0)\uparrow \\ \uparrow & \text{otherwise} \end{cases} \qquad \text{computable}$$

OBSERVATION: Let $f : \mathbb{N} \to \mathbb{N}$ be computable and injective total

Then
$$f^{-1}(y) = \begin{cases} x & \text{s.t. } f(x) = y \text{ , if it exists} \\ \uparrow & \text{otherwise} \end{cases}$$

is computable

proof

$$f^{-1}(y) = \mu x. \; |f(x) - y|$$

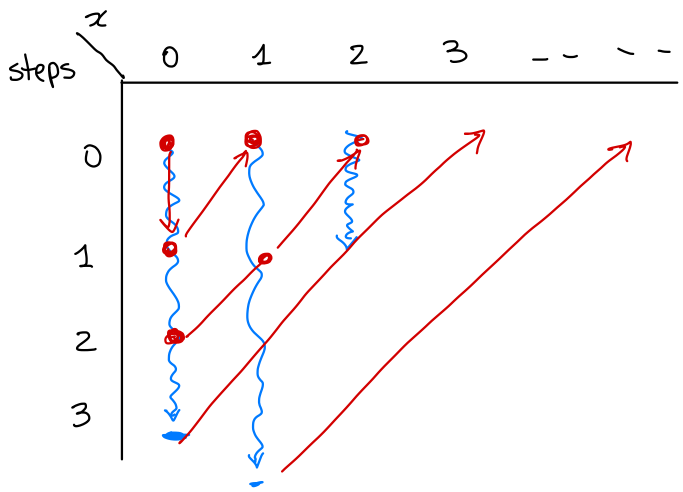
Not working for non-total functions

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

$$f(x) = \begin{cases} x - 1 & \text{if } x > 0 \\ \uparrow & \text{otherwise} \end{cases} \overset{\text{computable}}{=} (x \dot{-} 1) + \mu z. \; \overline{sg}(x)$$

$$f^{-1}(y) = y + 1 \neq \underbrace{\mu x. \; |f(x) - y|}_{\text{always undefined}}$$

The result is true also for non-total functions



$f^{-1}(y) =$

find x s.t. $f(x) = y$

$\downarrow$

f computed by P

$\downarrow$

check $P(x)$
- for any x
- any number of steps

PARTIAL RECURSIVE FUNCTIONS

computational models : URM, TM, λ-calculus, ...

Church Turing Thesis : A function is computable by an effective procedure

iff

it is computable by URM - machine

Program

→ class $\mathcal{R}$ of partial recursive functions

→ $\mathcal{R} = \mathcal{C}$

<u>Def</u> : The class of <u>partial recursive function</u> $\mathcal{R}$ is the <u>least class</u> of functions which

w.r.t $\subseteq$

contains (a) zero and is closed by (1) composition
(b) successor (2) primitive recursion
(c) projections (3) minimalisation

in detail :

→ we call a class of functions $\mathcal{A}$ <u>rich</u> if it

contains (a) zero and is closed by (1) composition
(b) successor (2) primitive recursion
(c) projections (3) minimalisation

→ $\mathcal{F}$ is rich

→ $\mathcal{R}$ is a rich class and $\forall \mathcal{A}$ $\mathcal{A}$ rich $\mathcal{R} \subseteq \mathcal{A}$

→ NOTE : rich classes are closed by intersection $\mathcal{A}_i$ $i \in I$ rich

$\Rightarrow \bigcap_{i \in I} \mathcal{A}_i$ is rich

→ $$\mathcal{R} = \bigcap_{\mathcal{A} \text{ rich}} \mathcal{A}$$

Equivalently: $\mathcal{R}$ is the class of functions which you can obtain from $\underbrace{(a),(b),(c)}_{\text{BASIC}}$ using (1), (2), (3) a finite number of times

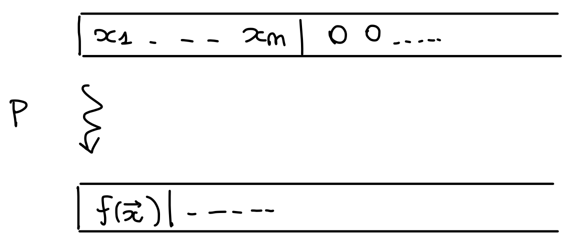
Theorem: $\mathcal{C} = \mathcal{R}$

($\mathcal{R} \subseteq \mathcal{C}$) $\quad \mathcal{C}$ is rich, $\mathcal{R}$ is the smallest rich class

$\leadsto \mathcal{R} \subseteq \mathcal{C}$

($\mathcal{C} \subseteq \mathcal{R}$) $\quad$ let $f \in \mathcal{C}$ $\quad f: \mathbb{N}^k \to \mathbb{N}$

there is P URM program



$$\begin{cases} C_P^1 : \mathbb{N}^{k+1} \to \mathbb{N} \\ C_P^1(\vec{x}, t) = \text{content of } R_1 \text{ after } t \text{ steps of computation of } P(\vec{x}) \end{cases}$$

$$\begin{cases} J_P : \mathbb{N}^{k+1} \to \mathbb{N} \\ J_P(\vec{x}, t) = \begin{cases} \text{instruction to be executed after } t \text{ steps of } P(\vec{x}) \\ 0 \quad \text{if } P(\vec{x}) \text{ terminate in } t \text{ or fewer steps} \end{cases} \end{cases}$$

let $\vec{x} \in \mathbb{N}$

→ if $f(\vec{x})\downarrow$ then $P(\vec{x})$ terminates in $\mu t. J_P(\vec{x}, t)$

$$f(\vec{x}) = C_P^1(\vec{x}, \mu t. J_P(\vec{x}, t))$$

→ if $f(\vec{x})\uparrow$ then $P(\vec{x})\uparrow$ hence $\mu t. J_P(\vec{x}, t)\uparrow$

$$f(\vec{x}) = C_P^1(\vec{x}, \mu t. J_P(\vec{x}, t))\uparrow$$

Hema

$$f(\vec{x}) = C_p^1(\vec{x}, \mu t.\ J_p(\vec{x}, t)) \qquad \text{for all } \vec{x} \in \mathbb{N}^k$$

If we can show $\boxed{C_p^1, J_p \in \mathcal{R}}$ we conclude that $f \in \mathcal{R}$

NEXT LESSON

TO BE CONTINUED