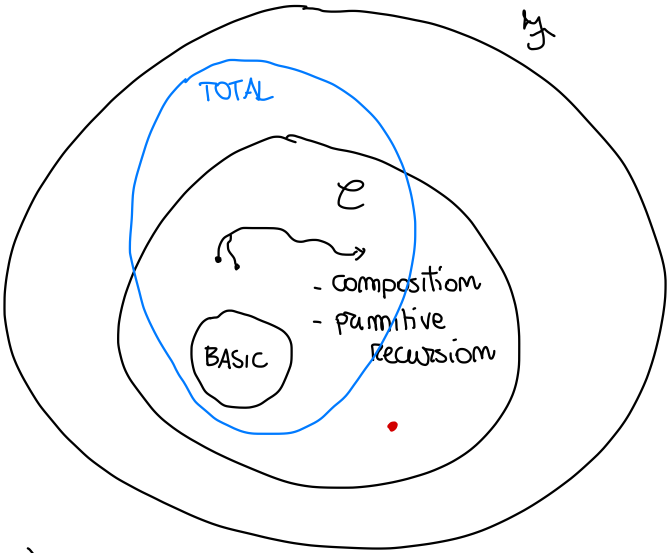
Computability (29/10/2024)



* Unbounded minimalisation

Given a function (possibly partial)

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

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

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

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

two issues :

- → there could be no y s.t. $f(\vec{x}, y) = 0$
- → $f(\vec{x}, z)$ might be undefined "before" finding y s.t. $f(\vec{x}, y) = 0$

↳ minimalisation is $\uparrow$

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

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

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

NO

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

NO

$f(\vec{x}, 2) = 0$? - - -

⋮

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

<u>proof</u>

Let $f : \mathbb{N}^{k+1} \to \mathbb{N}$ be a computable function

and let P be a (standard form) program for f

We show that $h(\vec{x}) = \mu y.\, f(\vec{x}, y)$ is computable by showing a program for it

1	k	m	m+1	m+k	m+k+1	m+k+2
$x_1 \ldots$	x_k		x_1	x_k	i	0

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

$f(\vec{x}, i)$ $i = 0$
$= 1$
$= 2$
$\vdots$

program for h is

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

LOOP: $P[m+1, \ldots, m+k, m+k+1 \to 1]$ // $f(\vec{x}, i)$ in R_1

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

$S(m+k+1)$ // $i{+}{+}$

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

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

$\square$

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

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

$$= \mu y.\ |2 * y - x|$$

$$= \mu y.\ |y + y - x|$$

computable by minimalisation

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

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

$$\neq \mu z.\ |z * y - x|$$

$y = 0$ & $x = 0$

we get 0

we want $\uparrow$

$$= \mu z.\ \left(|z \times y - x| + \underbrace{\overline{sg}(y)}_{\begin{cases} 1 & \text{if } y=0 \\ 0 & \text{if } y \neq 0 \end{cases}}\right)$$

<u>OBSERVATION</u> : Every finite (domain) function is computable

<u>proof</u>

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

$$\vartheta(x) = \begin{cases} y_1 & \text{if } x = x_1 \\ y_2 & \text{if } x = x_2 \\ \vdots & \vdots \\ y_m & \text{if } x = x_m \\ \uparrow & \text{otherwise} \end{cases} \qquad 0 \qquad\qquad dom(\vartheta) = \{x_1, \dots, x_m\}$$

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

ϑ is computable

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

$\square$

<u>Example</u> :

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

$$f(x) = \begin{cases} 1 & \text{if } x = 0 \text{ and } P = NP \\ 0 & \text{if } x = 0 \text{ and } P \neq NP \\ \uparrow & \text{if } x \neq 0 \end{cases}$$

computable

$g : \mathbb{N} \to \mathbb{N}$ fix a program P

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

computable

<u>OBSERVATION</u> : Let $f : \mathbb{N} \to \mathbb{N}$ computable and injective
TOTAL

Then

$$f^{-1}(y) = \begin{cases} x & \text{if there is } x \text{ st. } f(x)=y \\ \uparrow & \text{if there is no such } x \end{cases}$$

is computable.

<u>proof</u>

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

□

Not working for partial function

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

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

$$= (x \dot{-} 1) + \boxed{\mu z. \, \overline{sg}(x)}$$

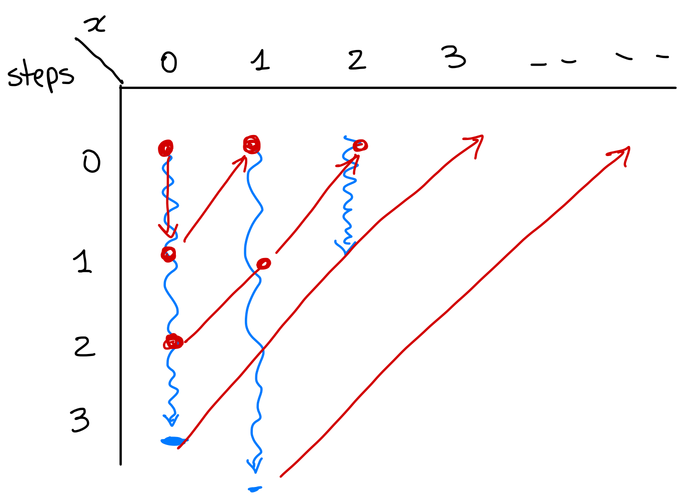
$g(x,z) = \overline{sg}(x)$

computable

$x=0$ $|f(0) - y| = 0$?

$$f^{-1}(y) = y+1 \neq \mu x. \, |f(x) - y| \uparrow \quad \forall y$$

* What if f is partial? Does the result hold? YES



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

f computed by P

you check $P(x)$
for any x
and any number of steps

Partial Recursive Functions

computational models : TM, λ-calculus, Post systems, ..., URM-machines

Church Turing thesis : A function is computable by an effective procedure
iff
it is URM-computable

TO DO LIST :

- → define the class $\mathcal{R}$ of partial recursive functions
- → $\mathcal{C} = \mathcal{R}$

Def. The class of partial recursive functions $\mathcal{R}$ is $\underbrace{\text{the least}}_{\text{w.r.t. } \subseteq}$ class of functions which

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

→ closed under
(1) composition
(2) primitive recursion
(3) minimalisation

→ define rich class of functions $\mathcal{A}$ as a class of functions which
* contains (a), (b), (c)
* is closed w.r.t. (1), (2), (3)

→ $\mathcal{R}$ is the least rich class, i.e. for all rich class $\mathcal{A}$ $\quad \mathcal{R} \subseteq \mathcal{A}$

→ OBSERVE : • given $\mathcal{A}_i$ $i \in I$ rich classes then $\bigcap_{i \in I} \mathcal{A}_i$ is rich
• the class of all functions is rich

define $\mathcal{R} = \bigcap_{\mathcal{A} \text{ rich class}} \mathcal{A}$

Equivalently : $\mathcal{R}$ is the class of functions that one obtains from the BASIC FUNCTIONS using operations (1), (2), (3) a finite number of times.

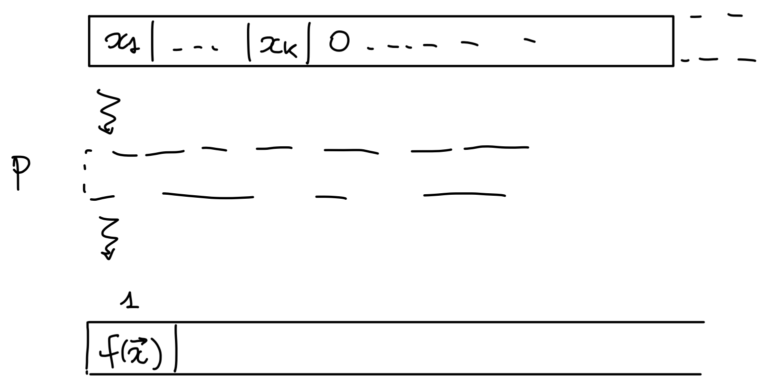
Theorem : $\mathcal{C} = \mathcal{R}$

proof

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

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

let P be a program for f



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

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

let $\vec{x} \in \mathbb{N}^k$. We distinguish two cases

→ $f(\vec{x})\downarrow$ $\quad$ then $P(\vec{x})\downarrow$ in a number steps

$$t_0 = \mu t . J_P(\vec{x}, t)$$

and

$$f(\vec{x}) = C_P^1(\vec{x}, t_0) = C_P^1(\vec{x}, \mu t . J_P(\vec{x}, t))$$

→ $f(\vec{x})\uparrow$ $\quad$ then $P(\vec{x})\uparrow$ and thus $\mu t . J_P(\vec{x}, t) \uparrow$

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

for all $\vec{x}$

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

If we are able to show $\underbrace{C_P^1, J_P \in \mathcal{R}}_{\text{NEXT LESSON}}$ then $f \in \mathcal{R}$