COMPUTABILITY (13/11/2023)

UNIVERSAL FUNCTION

Def: Given $k \geq 1$ the universal function (for functions of arity k) is

$\Psi_U^{(k)} : \mathbb{N}^{k+1} \to \mathbb{N}$

$\Psi_U^{(k)}(e, \vec{x}) = \varphi_e^{(k)}(\vec{x})$ (well-defined)

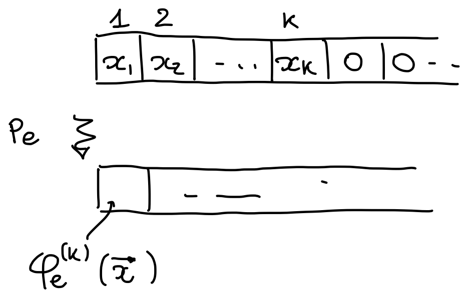
Theorem: $\Psi_U^{(k)} : \mathbb{N}^{k+1} \to \mathbb{N}$ is computable

proof

given $e \in \mathbb{N}$ $\quad \vec{x} \in \mathbb{N}^k$

we want $\Psi_U^{(k)}(e, \vec{x}) = \varphi_e^{(k)}(\vec{x})$

intuitive idea
- $\to$ get the program $P_e = \gamma^{-1}(e)$
- $\to$ execute P_e



$\to$ configuration of memory

r_1	r_2	- - - -	r_m	0	0	- - -

$c = \prod_{i \geq 1} p_i^{r_i}$ $\qquad r_i = (c)_i$

$C_k : \mathbb{N}^{k+2} \to \mathbb{N}$

$C_k(e, \vec{x}, t) =$ configuration of the memory after t step of $P_e(\vec{x})$
(if $P_e(\vec{x})$ terminates in t steps or less $\leadsto$ final configuration)

$J_k : \mathbb{N}^{k+2} \to \mathbb{N}$

$$J_k(e, \vec{x}, t) = \begin{cases} \text{index of instruction to be executed after } t \text{ steps of } P_e(\vec{x}) \\ \text{if } P_e(\vec{x}) \text{ does not halt in } t \text{ steps or fewer} \\ 0 \quad \text{otherwise} \end{cases}$$

Observe

$\rightarrow$ $P_e(\vec{x})\downarrow$ then it stops in $t_0 = \mu t.\ J_K(e, \vec{x}, t)$ steps

hence $\varphi_e^{(K)}(\vec{x}) = \left(C_K(e, \vec{x}, \mu t.\ J_K(e,\vec{x},t))\right)_1$

$\rightarrow$ $P_e(\vec{x})\uparrow$ then $\mu t.\ J_P(e, \vec{x}, t)\uparrow$

$$\varphi_e^{(K)}(\vec{x}) = \left(C_K(e, \vec{x}, \mu t.\ J_K(e,\vec{x},t))\right)_1 \uparrow$$

Therefore

$$\Psi_U^{(K)}(e, \vec{x}) = \varphi_e^{(K)}(\vec{x}) = \left(C_K(e, \vec{x}, \mu t.\ J_K(e,\vec{x},t))\right)_1$$

if we show C_K, J_K computable $\Rightarrow$ $\Psi_U^{(K)}$ computable.

$\boxed{\text{AIM: show } C_K, J_K \text{ computable}}$

$*$ given $i \in \mathbb{N}$ instruction code i.e. $i = \beta(\text{Instr})$

$\text{zarg}(i) = qt(4, i) + 1$ $\qquad$ $i = \beta(Z(n)) = 4 * (n-1)$

$\text{Sarg}(i) = qt(4, i) + 1$ $\qquad$ $i = \beta(S(n)) = 4 * (n-1) + 1$

$\text{Targ}_1(i) = \pi_1(qt(4,i)) + 1$ $\qquad$ $i = \beta(T(m,n)) = 4 * \pi(m-1, n-1) + 2$

$\text{Targ}_2(i) = \pi_2(\text{-----}) + 1$

$\text{Jarg}_1, \text{Jarg}_2, \text{Jarg}_3$

computable

$*$ effect of executing some algebraic instruction on configuration c

r_1	r_2	...	r_m	...

$c = p_1^{r_1} \cdot p_2^{r_2} \cdots p_m^{\cancel{r_m}\ 0} \cdots$ $\qquad$ $(c)_m$

$\text{zero}(c, n) = qt(p_n^{(c)_n}, c)$

$\text{succ}(c, n) = c \cdot p_n$

$\text{transf}(m, n) = \text{zero}(c, n) \cdot p_n^{(c)_m}$ $\qquad$ computable

$*$ effect on configuration c of executing instruction with code i

$$\text{change}(c, i) = \begin{cases} \text{zero}(c, \text{zarg}(i)) & \text{if } rm(4,i) = 0 \\ \text{succ}(c, \text{sarg}(i)) & \text{if } rm(4,i) = 1 \\ \text{transf}(c, \text{Targ}_1(i), \text{Targ}_2(i)) & \text{if } rm(4,i) = 2 \\ c & \text{if } rm(4,i) = 3 \end{cases}$$

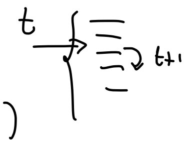
computable

* configuration of registers starting from configuration c and executing instruction number t in program P_e

$$
\mathrm{nextconf}(e, c, t) = \begin{cases} \mathrm{change}(c, a(e,t)) & \text{if } 1 \leq t \leq \ell(e) \\ c & \text{otherwise} \end{cases}
$$

↖ computable

* number of the next instruction to be executed after executing $i = \beta(\mathrm{Instr})$ and this is in position t in the program



$$
ni(c, i, t) = \begin{cases} t+1 & \text{if } (rm(4, i) \neq 3) \text{ or} \\ & \quad (rm(4,i) = 3 \text{ and } (c)_{\mathrm{Jarg}_1(i)} \neq (c)_{\mathrm{Jarg}_2(i)}) \\ \mathrm{Jarg}_3(i) & \text{otherwise} \end{cases}
$$

↖ computable

* next instruction, if we execute instruction in position t of P_e in configuration c

$$
\mathrm{nextinstr}(e, c, t) = \begin{cases} ni(c, a(e,t), t) & \text{if } 1 \leq t \leq \ell(e) \\ & \text{and } 1 \leq ni(c, a(e,t), t) \leq \ell(e) \\ 0 & \text{otherwise} \end{cases}
$$

↖ computable

Now

$$
c_k(e, \vec{x}, 0) = \prod_{i=1}^{k} p_i^{x_i}
$$

| x_1 | ... | x_k | 0 | 0 ... |

$$
J_k(e, \vec{x}, 0) = 1
$$

$$
c_k(e, \vec{x}, t+1) = \mathrm{nextconf}(e, c_k(e, \vec{x}, t), J_k(e, \vec{x}, t))
$$

$$
J_k(e, \vec{x}, t+1) = \mathrm{nextinstr}(e, c_k(e, \vec{x}, t), J_k(e, \vec{x}, t))
$$

c_k, J_k defined by primitive recursion from computable functions are computable (actually they are in PR) [no minimalisation]

Now $\varphi_0^{(k)}(e, \vec{x}) = \left(C_k(e, \vec{x}, \mu t. J_p(e, \vec{x}, t)) \right)_1$

computable

$\square$

<u>Corollary</u>: The following predicates are decidable

(a) $H_k(e, \vec{x}, t)$ = " $P_e(\vec{x}) \downarrow$ in t steps or fewer"

(b) $S_k(e, \vec{x}, y, t)$ = " $P_e(\vec{x}) \downarrow y$ in t steps or fewer"

<u>proof</u>

(a) $\chi_{H_k} : \mathbb{N}^{k+2} \to \mathbb{N}$

$$\chi_{H_k}(e, \vec{x}, t) = \begin{cases} 1 & \text{if } H_k(e, \vec{x}, t) \\ 0 & \text{otherwise} \end{cases}$$

$$= \overline{sg}\left(J_k(e, \vec{x}, t) \right)$$

$\downarrow 0$ if $P_e(\vec{x}) \downarrow$ in t steps

$\neq 0$ otherwise

computable by composition

(b)

$$\chi_{S_k}(e, \vec{x}, y, t) =$$

$$= \chi_{H_k}(e, \vec{x}, t) \cdot \overline{sg}\left(|y - (C_k(e, \vec{x}, t))_1| \right)$$

computable by composition

When $k = 1$ we often omit it

$H(e, x, t)$ for $H_1(e, x, t)$

EXERCISE : Computability of the inverse, reprise

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

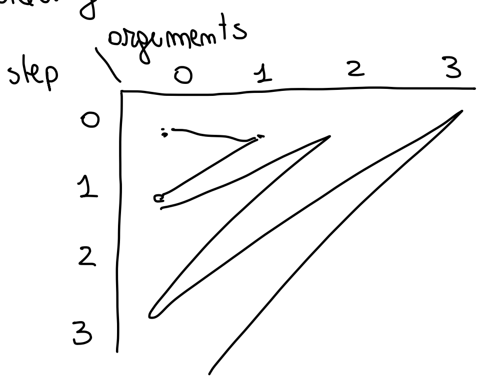
then $f^{-1} : \mathbb{N} \to \mathbb{N}$

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

is computable

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

without totality :



try n steps over argument x

for all possible n, x

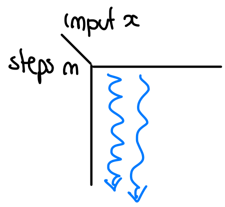
f is computable $\Rightarrow$ there is $e \in \mathbb{N}$ s.t. $f = \varphi_e$

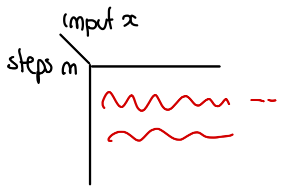
look for input x s.t. $\underbrace{\varphi_e(x) \downarrow y \text{ in } t \text{ steps}}_{S(e,\, x,\, y,\, t)}$

number steps n

$$f^{-1}(y) = \mu x . \mu m . \cancel{S(e, x, y, t)}$$

$$\mu m . \mu x . \cancel{S(e, x, y, t)}$$





$$f^{-1}(y) = (\mu w . \quad S(e, (w)_1, y, (w)_2))_1$$

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

$$w \to \underset{x}{\underline{(w)_1}}, \ \underset{m}{\underline{(w)_2}}$$

$w = 3 = 2^0 \cdot 3^1 \to (0, 1)$

$w = 6 = 2^1 \cdot 3^1 \to (1, 1)$

$w = 30 = 2^1 \cdot 3^1 \cdot 5^1 \to (1, 1)$

not injective

OBSERVATION: function which is total and not computable

$$f(x) = \begin{cases} \boxed{\varphi_x(x) + 1} & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{if } \varphi_x(x)\uparrow \end{cases}$$

NOT A PROBLEM

$$= \begin{cases} \Psi_U(x,x) + 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{otherwise} \end{cases}$$

EXERCISE: show that the predicate below is UNDECIDABLE

$$Halt(x) = \begin{cases} \text{true} & \text{if } \varphi_x(x)\downarrow \quad (x \in W_x) \\ \text{false} & \text{otherwise} \end{cases}$$

HALTING PROBLEM