COMPUTABILITY (09/11/2026)

UNIVERSAL FUNCTION

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

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

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

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

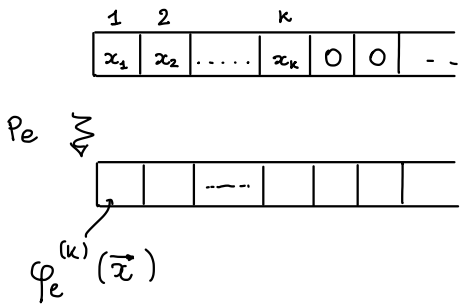
proof

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

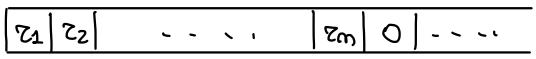
we want $\psi_U^{(k)}(e, \vec{x}) = \varphi_e^{(k)}(\vec{x})$

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

→ execute P_e



→ configuration of registers



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

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

$c_k(e, \vec{x}, t)$ = configuration of registers after t steps of $P_e(\vec{x})$
(if $P_e(\vec{x})\downarrow$ in t steps or fewer then final configuration

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

$$J_k(e, \vec{x}, t) = \begin{cases} \text{index of next instruction to be executed after } t \text{ steps of } P_e(\vec{x}) \\ 0 \quad \text{if } P_e(\vec{x}) \text{ halts in } t \text{ or fewer steps} \end{cases}$$

Observe that

- $P_e(\vec{x})\downarrow$ then P_e terminates on $\vec{x}$ in a number of steps

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

and

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

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

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

Therefore

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

hence if we show $c_K, J_K \in \underset{\mathcal{R}}{\underset{\shortparallel}{\mathcal{C}}}$ we conclude $\psi_U^{(K)} \in \underset{\mathcal{R}}{\underset{\shortparallel}{\mathcal{C}}}$

$\boxed{\text{AIM: } c_K, J_K \text{ computable}}$

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

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

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

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

$\text{Torg}_2(i) = \pi_2(qt(4,i))+1$

$\text{Jorg}_1, \text{Jorg}_2, \text{Jorg}_3$ $\qquad i = \beta(J(m,n,t)) = 4*\nu(m-1,n-1,t-1)+3$

computable

* effect of the execution of an arithmetic instruction on c

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

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

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

r_1	r_2	....	r_n	

$c = p_1^{r_1}\cdot p_2^{r_2}\cdots\cdot p_n^{\cancel{r_n}\,0}\cdots$

$(c)_n$

computable

* effect of executing an instruction with code i over a configuration c

$$change(c,i) = \begin{cases} zero(c, Zorg(i)) & \text{if } rm(4,i) = 0 \\ succ(c, Sorg(i)) & \text{if } rm(4,i) = 1 \\ transf(c, Torg_1(i), Torg_2(i)) & \text{if } rm(4,i) = 2 \\ c & \text{if } rm(4,i) = 3 \end{cases}$$

computable

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

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

(recall $e = \tau(i_1 \dots i_k)$ $\quad a(e,j) = i_j$ $\quad \ell(e) = k$) computable

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



$$ni(c,i,t) = \begin{cases} t+1 & \text{if } rm(4,i) \neq 3 \\ & \text{or } (rm(4,i) = 3 \text{ and } (c)_{Jorg_1(i)} \neq (c)_{Jorg_2(i)}) \\ Jorg_3(i) & \text{otherwise} \end{cases}$$

computable

* next instruction to be executed, if we execute in configuration c instruction in position t of program P_e

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

computable

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

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

1		k		
x_1	...	x_k	0	...

$$c_K(e, \vec{x}, t+1) = \text{nextconf}(e, c_K(e, \vec{x}, t), J_K(e, \vec{x}, t))$$

$$J_K(e, \vec{x}, t+1) = \text{nextinstr}(e, c_K(e, \vec{x}, t), J_K(e, \vec{x}, t))$$

c_K, J_K defined by primitive recursion using computable functions (primitive recursive)

$\rightsquigarrow$ c_K, J_K computable (primitive recursive)

hence

$$\psi_U^{(k)}(e, \vec{x}) = \varphi_e^{(k)}(\vec{x}) = \left(c_K(e, \vec{x}\ \mu t.\ J_K(e, \vec{x}, t)) \right)_1$$

composition / minimalisation of computable functions $\rightarrow$ $\psi_U^{(k)}$ computable

□

Corollary: The following predicates are decidable

(a) $H_K(e, \vec{x}, t) \equiv$ "$P_e(\vec{x})\downarrow$ in t steps or less"

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

proof

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

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

$$= \overline{sg}(J_K(e, \vec{x}, t))$$

$J_K(e, \vec{x}, t)$: 0 if $P_e(\vec{x})\downarrow$ in t steps; $\neq 0$ otherwise

computable by composition

(b) $$\chi_{S_K}(e, \vec{x}, y, t) = \chi_{H_K}(e, \vec{x}, t) \cdot \overline{sg}(|y - c_K(e, \vec{x}, t)|)$$

$\chi_{H_K}(e, \vec{x}, t)$: 1 if $P_e(\vec{x})\downarrow$ in t steps; 0 otherwise

computable by composition

NOTE: When $K=1$ often omitted

	$K=1$
$\varphi_e^{(K)}(\vec{x})$	φ_e
$W_e^{(K)} \subseteq \mathbb{N}^K$	W_e
$E_e^{(K)} \subseteq \mathbb{N}$	E_e
$H_K(e, \vec{x}, t)$	H
$S_K(e, \vec{x}, y, t)$	S

* Computability of the inverse function

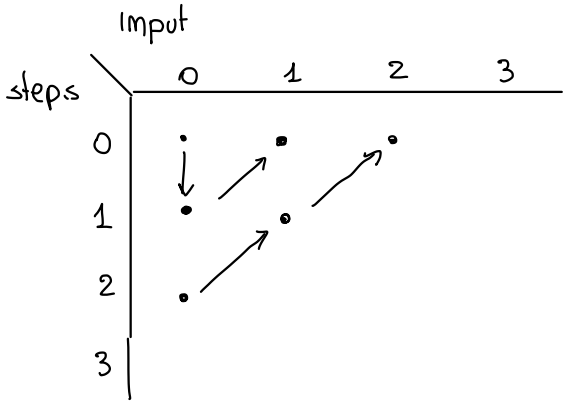
Let $f: \mathbb{N} \to \mathbb{N}$ ~~total~~ and injective

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

is computable

$f^{-1}(y) \not= \mu x. \; |f(x) - y|$

Without totality, in order to find x s.t. $f(x) = y$



try n steps over input x

for all possible pairs n, x

$\Leftrightarrow$

if there is x s.t. $f(x)=y$

you'll find it

formal proof

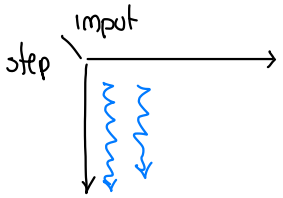
f is computable $\Rightarrow$ there is $e \in \mathbb{N}$ s.t. $f = \varphi_e$

look for input x

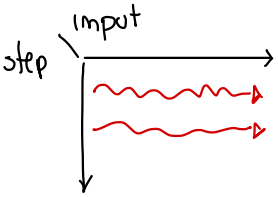
number of steps n

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

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



$$= \mu n.\ \mu x.\ S(e, x, y, t)$$



We have to search for a pair

$$f^{-1}(y) = \pi_1\left(\mu w.\ S(e, \pi_1(w), y, \pi_2(w))\right)$$

$$\pi^{-1}(w) = (\underbrace{\pi_1(w)}_{x}, \underbrace{\pi_2(w)}_{t})$$

more "practical"

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

$$w \rightsquigarrow (w)_1 \quad (w)_2$$

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

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

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

not injective
but surjective

OBSERVATION : function which is total and not computable

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

↑ computable

↑ problem

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

EXERCISE Show that the Halting Problem i.e. the predicate below is not decidable:

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

Use: (to be proved) If $f_1, f_2 : \mathbb{N} \to \mathbb{N}$ computable (possibly partial) and $Q(x) \subseteq \mathbb{N}$ decidable predicate

Then

$$f(x) = \begin{cases} f_1(x) & \text{if } Q(x) \\ f_2(x) & \text{otherwise} \end{cases}$$

is computable