Computability (09/11/2021)

* Smn (parametrisation) theorem

let $f : \mathbb{N}^2 \to \mathbb{N}$ computable $\qquad f = \varphi_e^{(2)}$

for any $x \in \mathbb{N}$ $\qquad f_x(y) = f(x,y) = \varphi_e^{(2)}(x,y)$ computable

$$= \varphi_{s(e,x)}(y)$$

$s : \mathbb{N}^2 \to \mathbb{N}$ total function $\qquad$ COMPUTABLE

$e, x \rightsquigarrow s(e,x)$

in general $\qquad f : \mathbb{N}^{m+n} \to \mathbb{N}$

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

SMN - THEOREM: Given $m, n \geq 1$ there is $s_{m,n} : \mathbb{N}^{m+1} \to \mathbb{N}$ total computable such that for all $e \in \mathbb{N}$, $\vec{x} \in \mathbb{N}^m$ $\forall \vec{y} \in \mathbb{N}^n$

$$\varphi_e^{(m+n)}(\vec{x}, \vec{y}) = \varphi^{(n)}_{\boxed{s_{m,n}(e,\vec{x})}}(\vec{y})$$

$\uparrow_{P'}$

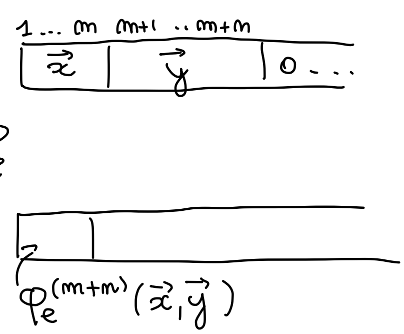
proof

intuitively:

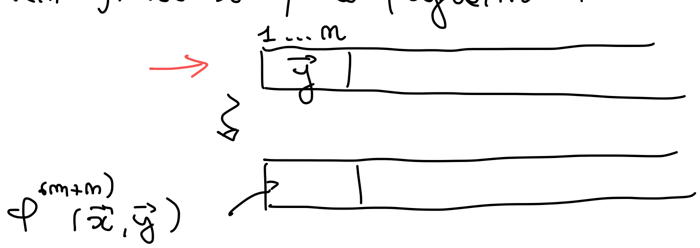
given $e \in \mathbb{N}$, $\vec{x} \in \mathbb{N}^m$

$\& \ \gamma^{-1}$

$P_e = \gamma^{-1}(e)$



we want, for a given fixed $\vec{x}$, a program P'

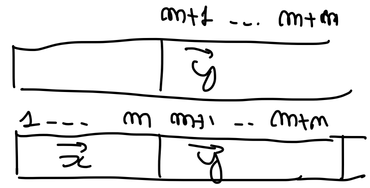


P' will be

$* \to$ move $\vec{y}$ to $R_{m+1} \dots R_{m+n}$

$* \to$ write $\vec{x}$ in $R_1 \dots R_m$

$\to$ execute P_e



P':

$T(1, m+1)$ /* $y_1 \rightsquigarrow R_{m+1}$ */

$\vdots$

$T(n, m+n)$ /* $y_n \rightsquigarrow R_{m+n}$ */

$Z(1)$

$\left.\begin{array}{l} S(1) \\ \vdots \\ S(1) \end{array}\right\}$ x_1 times /* x_1 to R_1 */

$\vdots$

$Z(m)$

$\left.\begin{array}{l} S(m) \\ \vdots \\ S(m) \end{array}\right\}$ x_m times /* x_m to R_m */

P_e

$= \gamma^{-1}(e)$

$$S_{m,n}(e, \vec{x}) = \gamma(P')$$

$\hookrightarrow$ computable

① sequential composition of programs

(idea: $e_1, e_2 \rightsquigarrow \gamma\left(\begin{array}{l} P_{e_1} \\ P_{e_2} \end{array}\right)$)

(1.a) $upd : \mathbb{N}^2 \to \mathbb{N}$

$upd(e, h) = \gamma$ (program obtained from $P_e = \gamma^{-1}(e)$ by updating each jump $J(m,n,t) \rightsquigarrow J(m,n,t+h)$)

first

$\boxed{\widetilde{upd}(i, h)} = \beta$ (instruction $\beta^{-1}(i)$, updated if it is a jump)

$$= \begin{cases} i & \text{if } rm(4,i) \neq 3 \\ \nu(\nu_1(q), \nu_2(q), \nu_3(q)+h)*4+3 & \text{if } rm(4,i) = 3 \end{cases}$$

$\uparrow$ computable

$$q = qt(4,i)$$

$$\beta^{-1}(i) = J(\nu_1(q)+1, \nu_2(q)+1, \underbrace{\nu_3(q)+1}_{+h})$$

$$= \left(\mathrm{sg}\,|\mathrm{rm}(4,i)-3|\right)\cdot i \; +$$
$$\left(\overline{\mathrm{sg}}\,|\mathrm{rm}(4,i)-3|\right)\cdot\left(\nu(\nu_1(q), \nu_2(q), \nu_3(q))*4+3\right)$$

now

$$\mathrm{upd}(e,h) = \tau\left(\widetilde{\mathrm{upd}}(a(e,1),h), \widetilde{\mathrm{upd}}(a(e,2),h), \ldots, \widetilde{\mathrm{upd}}(a(e,\ell(e)),h)\right)$$
$$= \left(\prod_{i=1}^{\ell(e)-1} p_i^{\widetilde{\mathrm{upd}}(a(e,i),h)}\right)\cdot p_{\ell(e)}^{\widetilde{\mathrm{upd}}(a(e,\ell(e)))+1} \dot{-} 2$$

- $c : \mathbb{N}^2 \to \mathbb{N}$

$$c(e_1, e_2) = \tau\left(a(e_1,1), \ldots, a(e_{\ell(e_1)}, \ell(e_1))\; a(e_2,1) \ldots a(e_2, \ell(e_2))\right)$$
$$= \ldots$$

- $\mathrm{seq} : \mathbb{N}^2 \to \mathbb{N}$

$$\mathrm{seq}(e_1, e_2) = \gamma\begin{pmatrix}\gamma^{-1}(e_1)\\ \gamma^{-1}(e_2)\end{pmatrix} = c(e_1, \mathrm{upd}(e_2, \ell(e_2)))$$

* $\mathrm{transf} : \mathbb{N}^2 \to \mathbb{N}$

$$\mathrm{transf}(m,n) = \gamma\begin{pmatrix}T(1, m+1)\\ T(n, m+n)\end{pmatrix} = \ldots$$

* $\mathrm{set} : \mathbb{N}^2 \to \mathbb{N}$

$$\mathrm{set}(i,x) = \gamma\begin{pmatrix}z(i)\\ \left.\begin{matrix}s(i)\\ \vdots\\ s(i)\end{matrix}\right\} x\end{pmatrix} = \ldots$$

* finally

$S_{m,n}(e, \vec{x}) =$

$\text{seq}(\text{transf}(m,n),$
$\quad \text{seq}(\text{set}(1, x_1),$
$\quad\quad \text{seq}(\text{set}(2, x_2),$
$\quad\quad\quad \ddots$
$\quad\quad\quad\quad \text{seq}(\text{set}(m, x_m), e) \dots)$

P':

$T(1, m+1)$ /* y_1 in R_{m+1} */
$\vdots$
$T(n, m+n)$ /* $y_n \rightsquigarrow R_{m+n}$ */

$Z(1)$
$S(1)$
$\vdots$
$S(1)$ } x_1 times /* x_1 to R_1 */
$\vdots$
$Z(m)$
$S(m)$
$\vdots$
$S(m)$ } x_m times /* x_m to R_m */

P_e

composition of computable total functions $\Rightarrow$ computable total

$\square$

Corollary: Let $f: \mathbb{N}^{m+n} \to \mathbb{N}$ computable.

There exists $s: \mathbb{N}^m \to \mathbb{N}$ total & computable such that

$$f(\vec{x}, \vec{y}) = \varphi^{(n)}_{s(\vec{x})}(\vec{y})$$

proof

since f is computable

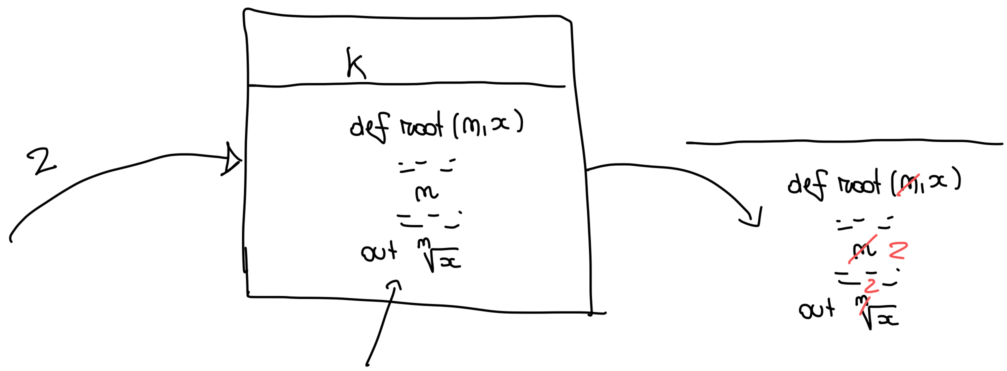
$$f(\vec{x}, \vec{y}) = \varphi_e^{(m+n)}(\vec{x}, \vec{y}) \quad \text{for some } e \in \mathbb{N}$$
$$= \varphi^{(n)}_{S_{m,n}(e, \vec{x})}(\vec{y}) \quad \text{just let } s(\vec{x}) = S_{m,n}(e, \vec{x})$$
$$= \varphi^{(n)}_{s(\vec{x})}(\vec{y})$$

Example:

Prove that there exists a total computable function $k: \mathbb{N} \to \mathbb{N}$ such that for all $n, x \in \mathbb{N}$

$$\varphi_{k(n)}(x) = \lfloor \sqrt[n]{x} \rfloor$$

$n \rightarrow \boxed{k} \rightarrow$ program computing $\sqrt[n]{\ }$



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

$$f(m, x) = \lfloor \sqrt[m]{x} \rfloor = \max z.\ \text{"}z^m \leq x\text{"}$$

$$= \mu z.\ \text{"}(z+1)^m > x\text{"}$$

$$= \mu z \leq x.\ \left(x + 1 \mathbin{\dot{-}} (z+1)^m\right)$$

f computable (bounded minimalisation of a composition of known computable function)

by (Corollary to) smn theorem there exist $k : \mathbb{N} \to \mathbb{N}$ total computable such that for all m, x

$$\varphi_{k(m)}(x) = f(m, x) = \lfloor \sqrt[m]{x} \rfloor$$

<u>EXAMPLE</u>: There exists $k : \mathbb{N} \to \mathbb{N}$ total computable such that

$\forall m$ $\varphi_{k(m)}$ is defined only on m-th powers (y^m for some y)

$$W_{k(m)} = \{ x \mid \exists y.\ x = y^m \}$$

$$f(m, x) = \begin{cases} \sqrt[m]{x} & \text{if } \exists y.\ x = y^m \\ \uparrow & \text{otherwise} \end{cases}$$

$\uparrow$

$$= \mu y.\ |x - y^m| \qquad \text{computable}$$

by corollary of smn theorem $\exists$ $k : \mathbb{N} \to \mathbb{N}$ total computable such that

$$\varphi_{k(n)}(x) = f(n, x)$$

We claim

$$W_{k(n)} = \{ x \mid \exists y .\ y^n = x \}$$

in fact

$$x \in W_{k(n)} \quad \text{iff} \quad \underset{\substack{\parallel \\ f(n,x)}}{\varphi_{k(n)}(x)} \downarrow \quad \text{iff} \quad \exists y \text{ s.t. } x = y^n$$

EXERCISE : show that there exists a total computable function $s : \mathbb{N} \to \mathbb{N}$ such that

$$W^{(k)}_{s(x)} = \{ (y_1, \dots, y_k) \mid \sum_{j=1}^{k} y_j = x \}$$

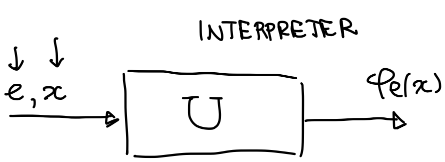
[try yourself]

* Universal Function

Let $\psi_U : \mathbb{N}^2 \to \mathbb{N}$

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

Is it computable ? $\boxed{\text{YES}}$

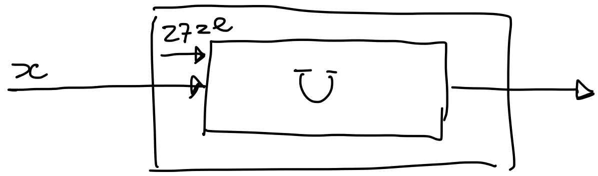


$\varphi_0 \qquad \varphi_1 \qquad \varphi_2 \qquad \varphi_3 \qquad \dots$

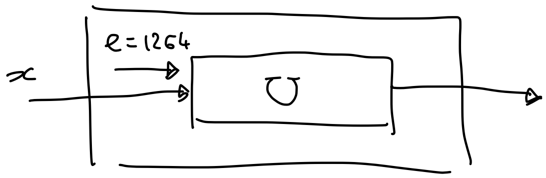
$\uparrow \qquad\qquad\qquad \uparrow$

1264

Turing ACME & c.



\$ 10000



\$ - - -

⋮

<u>Theorem</u> (Universal program) : Let $k \geq 1$

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

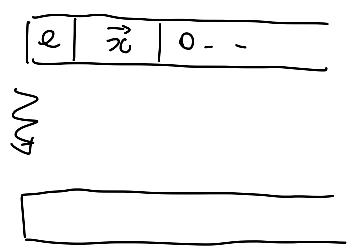
$$\psi_U^{(k)}(e, \vec{x}) = \varphi_e^{(k)}(\vec{x})$$

is computable

<u>proof</u>

fixed $k \geq 1$

given $\underset{\mathbb{N}}{\underset{\Uparrow}{e}}$, $\underset{\mathbb{N}^k}{\underset{\Uparrow}{\vec{x}}}$



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

→ determine $P_e = \gamma^{-1}(e)$

→ start

→ execute P_e

$\varphi_e^{(k)}(\vec{x})$

by Church - Turing thesis

computable

□