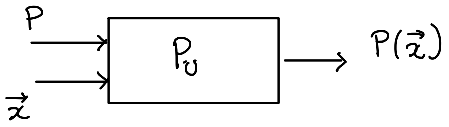
COMPUTABILITY (03/11/2025)



* Enumeration of URM programs

a set X is countable (denumerable) if $|X| \leq |\mathbb{N}|$

i.e. there is $f : \mathbb{N} \to X$ surjective (enumeration)

$$\underbrace{f(0) \quad f(1) \quad f(2) \quad f(3) \quad \text{-----}}_{X}$$

if f is also injective $\rightsquigarrow$ f is a bijective enumeration

f "effective"

Lemma : there are bijective enumerations of (effective)

(a) $\mathbb{N}^2$

(b) $\mathbb{N}^3$

(c) $\bigcup_{k \geq 1} \mathbb{N}^k$

(1) $\pi : \mathbb{N}^2 \to \mathbb{N}$

$\pi(x, y) = 2^x (2y+1) - 1$ computable

$\pi^{-1} : \mathbb{N} \to \mathbb{N}^2$

$\pi^{-1}(n) = (\pi_1(n), \pi_2(n))$ $\pi_1 : \mathbb{N} \to \mathbb{N}$ $\pi_2 : \mathbb{N} \to \mathbb{N}$ computable

$\pi_1(n) = (n+1)_1$ $n = \underbrace{2^x (2y+1)} - 1$

$$\pi_2(n) = \left(\left(\frac{n+1}{2^{\pi_1(n)}}\right) \dot{-} 1\right) / 2$$

$$= qt(2, qt(2^{\pi_1(n)}, (n+1)) \dot{-} 1))$$ computable

(2) $\nu : \mathbb{N}^3 \to \mathbb{N}$

$$\nu(x_1, x_2, x_3) = \pi(x_1, \pi(x_2, x_3))$$

$\nu^{-1} : \mathbb{N} \to \mathbb{N}^3$

$$\nu^{-1}(m) = (\nu_1(m), \nu_2(m), \nu_3(m))$$

$\nu_1(m) = \pi_1(m)$ $\qquad$ $\nu_2(m) = \pi_2(\pi_2(m))$ $\qquad$ $\nu_3(m) = \pi_2(\pi_2(m))$

(3) $\tau : \bigcup_{k \geq 1} \mathbb{N}^k \to \mathbb{N}$

$$\tau(x_1, \dots x_k) = \prod_{i=1}^{k} p_i^{x_i} - 1$$

not injective

$$(1,0) \to p_1^1 \cdot p_2^0 - 1 = 2^1 \cdot 3^0 - 1 = 1$$

$$(1,0,0) \to p_1^1 \cdot p_2^0 \cdot p_3^0 - 1 = 2^1 \cdot 3^0 \cdot 5^0 - 1 = 1$$

$$\tau(x_1 \dots x_k) = \left(\prod_{i=1}^{k-1} p_i^{x_i} \right) \cdot p_k^{x_k + 1} \dot{-} 2$$

$\tau^{-1} : \mathbb{N} \to \bigcup_{k \geq 1} \mathbb{N}^k$

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

$$\tau^{-1}(m) = \underbrace{(a(m,1) \quad a(m,2) \qquad a(m, \ell(m)))}_{\ell(m)}$$

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

$\ell(m) =$ largest k such that p_k divides $m+2$

(computable EXERCISE)

$$a(m,i) = \begin{cases} (m+2)_i & \text{for } 1 \leq i < \ell(m) \\ (m+2)_i \dot{-} 1 & \text{otherwise} \end{cases}$$

$\ell : \mathbb{N} \to \mathbb{N}$, $\quad a : \mathbb{N}^2 \to \mathbb{N}$ $\quad$ computable

OBSERVATION : Let $\mathcal{P}$ the set of URM programs.

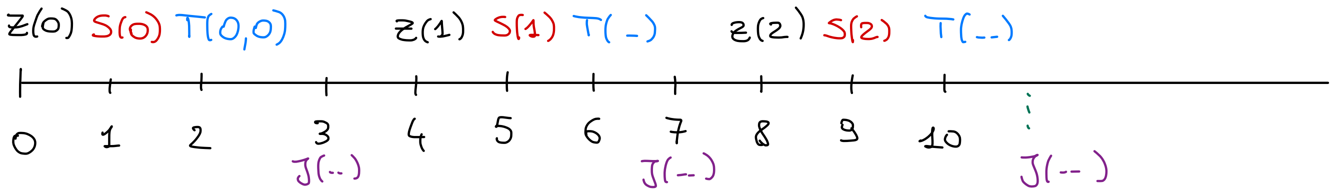
There is an effective bijective enumeration

$$\gamma : \mathcal{P} \to \mathbb{N}$$

Let $\mathcal{J} = \{ Z(n),\ S(n),\ T(m,n),\ J(m,n,t) \mid m,n,t \in \mathbb{N} \}$

we define

$$\beta : \mathcal{J} \to \mathbb{N}$$



$$\beta(Z(n)) = 4 * (n-1)$$

$$\beta(S(n)) = 4 * (n-1) + 1$$

$$\beta(T(m,n)) = 4 * \pi(m-1, n-1) + 2$$

$$\beta(J(m,n,t)) = 4 * \nu(m-1, n-1, t-1) + 3$$

$\beta^{-1} : \mathbb{N} \to \mathcal{J}$

$r = rm(4, x)$
$q = qt(4, x)$

$$\beta^{-1}(x) = \begin{cases} Z(q+1) & r = 0 \\ S(q+1) & r = 1 \\ T(\pi_1(q)+1,\ \pi_2(q)+1) & r = 2 \\ J(\nu_1(q)+1,\ \nu_2(q)+1,\ \nu_3(q)+1) & r = 3 \end{cases}$$

Given a program $P \in \mathcal{P}$ URM program

$$P = \begin{cases} I_1 \\ I_2 \\ \vdots \\ I_s \end{cases} \qquad \gamma(P) = \tau(\beta(I_1)\ \beta(I_2)\ \dots\ \beta(I_s))$$

inverse $\quad \gamma^{-1} : \mathbb{N} \to \mathcal{P}$

$$\gamma^{-1}(m) = P = \begin{cases} I_1 \\ I_2 \\ \vdots \\ I_{e(m)} \end{cases} \qquad I_i = \beta^{-1}(a(m,i))$$

* γ fixed enumeration of URM programs

$\gamma(P)$ (Gödel) number of P

given $m \rightsquigarrow P_m = \gamma^{-1}(m)$

Example :

* P

$$\begin{cases} T(1,2) \\ S(2) \\ T(2,1) \end{cases} \rightsquigarrow \begin{aligned} &4 * \pi(1-1, 2-1) + 2 = 4 * \overbrace{\pi(0,1)}^{\underbrace{2^0 \cdot (2 \cdot 1 + 1) - 1}_{2}} + 2 = 10 \\ &4 * (2-1) + 1 = 5 \\ &\ldots\ldots = 6 \end{aligned}$$

$$\gamma(P) = \tau(10 \ 5 \ 6)$$
$$= p_1^{10} \cdot p_2^{5} \cdot p_3^{6+1} \dot{-} 2 = 2^{10} \cdot 3^5 \cdot 5^{6+1} \dot{-} 2$$
$$= 19.439.999.998$$

* P' $\quad S(1)$

$$\gamma(P') = \tau(\beta(S(1))) = \tau(4 * (1-1) + 1) =$$
$$= \tau(1) = p_1^{1+1} - 2 = 2^2 - 2 = 2$$

* given $m = 40$

what is $P_{40} = \gamma^{-1}(40)$

$$40 = \underbrace{\left(\prod_{i=1}^{k-1} p_i^{x_i}\right) \cdot p_k^{x_k+1}}_{42} \dot{-} 2$$

$$40 + 2 = 2 \cdot 3 \cdot 7$$
$$= p_1^1 \cdot p_2^1 \cdot p_3^0 \cdot p_4^1$$

$e(40) = 4$

$$\begin{array}{lll} I_1 & \beta^{-1}(1) \quad \nearrow \begin{smallmatrix} r = 1 \\ q = 0 \end{smallmatrix} & S(1) \\ I_2 & \beta^{-1}(1) & S(1) \\ I_3 & \beta^{-1}(0) & Z(1) \\ I_4 & \beta^{-1}(0) = \beta^{-1}(1 \cdot 1) = & Z(1) \end{array}$$

* Fixed $\gamma : \mathcal{P} \to \mathbb{N}$

it induces an enumeration of computable functions

$\varphi_n^{(k)} : \mathbb{N}^k \to \mathbb{N}$ function of k arguments computed by $P_n = \gamma^{-1}(n)$ $\nwarrow$ $f_{P_n}^{(k)}$

$$W_n^{(k)} = \mathrm{dom}(\varphi_n^{(k)}) = \{\vec{x} \in \mathbb{N} \mid \varphi_n^{(k)}(\vec{x})\downarrow\} \subseteq \mathbb{N}^k$$

$$E_n^{(k)} = \mathrm{cod}(\varphi_n^{(k)}) = \{\varphi_n^{(k)}(\vec{x}) \mid \vec{x} \in W_n^{(k)}\} \subseteq \mathbb{N}$$

When k is 1, we omit it

φ_n instead of $\varphi_n^{(1)}$

<u>Example</u>

$\varphi_{40} : \mathbb{N} \to \mathbb{N}$

$\varphi_{40}(x) = 0 \quad \forall x$

$W_{40} = \mathbb{N} \qquad E_{40} = \{0\}$

<u>Note</u> : The same program with different arities computes different functions

Take

$P \qquad T(2,1)$

$$\begin{aligned} \beta(T(2,1)) &= 4 * \pi(2-1, 1-1) + 2 \\ &= 4 * \pi(1,0) + 2 \\ &= 4 * (2^1(2\cdot 0+1) - 1) + 2 \\ &= 6 \end{aligned}$$

$$\gamma(P) = \tau(6) = 2^{6+1} \dot{-} 2 = 128 - 2 = 126$$

$$\varphi_{126}(x) = \varphi^{(1)}_{126}(x) = 0 \quad \forall x$$

1	2	
x	0	

$$\varphi^{(2)}_{126}(x, y) = y$$

1	2			
x	y	0	0	- -

Back to the enumerations.

$$\underbrace{\varphi_0 \quad \varphi_1 \quad \underset{\text{successor}}{\overset{}{\varphi_2}} \quad \varphi_3 \quad \cdots\cdots \quad \varphi_{19.439.999.998}}_{}$$

successor = φ_2 ; successor = $\varphi_{19.439.999.998}$

enumeration of unary computable functions
with repetitions (not injective) INFINITE

$$|\mathcal{C}^{(1)}| \leq |\mathbb{N}|$$

$$|\mathcal{C}^{(k)}| \leq |\mathbb{N}|$$

$\mathcal{C} = \bigcup_{k \geq 1} \mathcal{C}^{(k)}$ denumerable $|\mathcal{C}| \leq |\mathbb{N}|$

$=$ ↑ infinitely many constants

__Exercise__ : $\mathcal{R}$ partial recursive function

least rich class

→ includes the basic functions

→ closed under
- composition
- primitive recursion
- minimalisation

Originally defined by Gödel - Kleene $\mathcal{R}_0$

least class which

→ includes the basic functions

→ closed under
- composition
- primitive recursion
- minimalisation usable only when the result is total

$$\mathcal{R}_0 \subseteq \mathcal{R} \cap Tot$$

$$? \supseteq$$

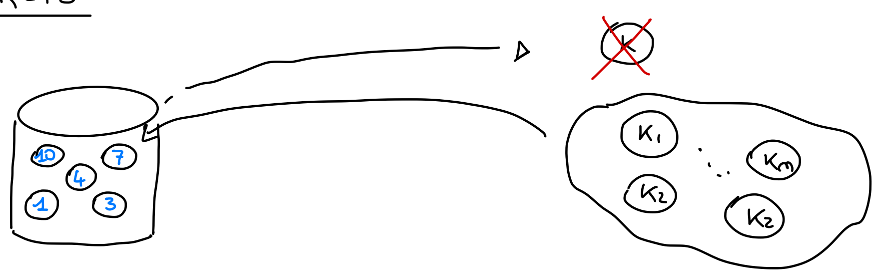
not obvious since total functions arise minimising partial functions

$f : \mathbb{N}^2 \to \mathbb{N}$ $\in \mathcal{R}$

$$f(x,y) = \begin{cases} 1 & y < x \\ 0 & y = x \\ \uparrow & \text{otherwise} \end{cases} = sg(x \dot- y) + \mu z.(y \dot- x)$$

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

__EXERCISE__ :



$K_1, \dots, K_m < K$

Does the process terminate? Why?