COMPUTABILITY (28/10/2025)

* Class of partial recursive functions $\mathcal{R}$

least rich class of functions i.e. least class of functions

→ including the BASIC FUNCTIONS

→ closed under

1. COMPOSITION
2. PRIMITIVE RECURSION
3. UNBOUNDED MINIMALISATION

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

proof

$(\mathcal{R} \subseteq \mathcal{C})$ $\mathcal{C}$ is rich

$(\mathcal{C} \subseteq \mathcal{R})$

let $f : \mathbb{N}^k \to \mathbb{N}$ be a function in $\mathcal{C}$

and let P a URM - program for f

Define

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

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

Then

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

We conclude by proving $\boxed{C_P^1, J_P \in \mathcal{R}}$

program P (in std form)

I_1
I_2
$\vdots$

I_s

memory

r_1	r_2		r_m	0	0	...

$$c = \prod_{i \geq 1} p_i^{r_i} = \prod_{i=1}^{m} p_i^{r_i}$$

$$r_i = (c)_i$$

1	2	3	4	5	...
2	0	1	0	0	...

$$c = p_1^2 \, p_2^0 \, p_3^1 \cdot p_4^0 \cdot p_5^0 \cdots$$
$$= 2^2 \cdot 3^0 \cdot 5^1 \cdots$$
$$= 20$$

$$(20)_3 = 1$$

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

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

define C_P, J_P by primitive recursion

$$\begin{cases} C_P(\vec{x}, 0) = \prod_{i=1}^{k} p_i^{x_i} \\ J_P(\vec{x}, 0) = 1 \end{cases}$$

x_1	...	x_k	0	...	

iteration cases

we define $C_P(\vec{x}, t+1)$, $J_P(\vec{x}, t+1)$

by using $\underbrace{C_P(\vec{x}, t)}_{c}$ and $\underbrace{J_P(\vec{x}, t)}_{J}$

$$
c_p(\vec{x}, t+1) = \begin{cases}
qt\left(p_m^{(c)_m}, c\right) & \text{if } 1 \leq J \leq \ell(P) \\ & \text{and } I_J = Z(m) \\[1em]
c \cdot p_m & \text{if } 1 \leq J \leq \ell(P) \\ & \text{and } I_J = S(m) \\[1em]
p_m^{(c)_m} \cdot qt\left(p_m^{(c)_m}, c\right) & \text{if } 1 \leq J \leq \ell(P) \\ & \text{and } I_J = T(m, n) \\[1em]
c & \text{otherwise} \\ & \left(J = 0 \text{ or } \begin{array}{l} 1 \leq J \leq \ell(P) \\ \text{and } I_J = J(m, n, u) \end{array} \right)
\end{cases}
$$

$$
c \quad \begin{array}{|c|c|c|c}
\hline
r_1 & \cdots & \cancel{r_m}\ 0 & \\
\hline
\end{array}
$$

$$
= p_1^{r_1} \cdot \ldots \cdot p_m^{\cancel{r_m}\ 0} \cdots
$$

$$
r_m = (c)_m
$$

$$
J_p(\vec{x}, t+1) = \begin{cases}
J+1 & \text{if } 1 \leq J < \ell(P) \\ & \text{and } I_J \neq \text{jump} \\ & \text{or } \left(\begin{array}{l} I_J = J(m, n, u) \\ \text{and } (c)_m \neq (c)_n \end{array} \right) \\[1em]
u & \text{if } 1 \leq J \leq \ell(P) \\ & \text{and } I_J = J(m, n, u), \ (c)_m = (c)_n \\ & \text{and } u \leq \ell(P) \\[1em]
0 & \text{otherwise}
\end{cases}
$$

Hence $c_p, J_p \in \mathcal{R}$

thus $$f(\vec{x}) = \left(c_p(\vec{x}, \mu t.\, J_p(\vec{x}, t)) \right)_1$$

therefore $f \in \mathcal{R}$

$\square$

* Primitive Recursive Functions

PR least class of functions

→ including the basic functions

→ closed under

1. composition
2. primitive recursion ← for loop
3. ~~minimalisation~~ ← while loop

$$\mathcal{PR} \subsetneq \mathcal{R} \cap Tot$$

?

$\mathcal{C}_{for}$ $\not\ni$ ψ $\qquad$ $\mathcal{C}_{for,\,while}$ $\ni$ ψ

Ackermann's Function

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

(1) ψ total
(2) $\psi \in \mathcal{C} = \mathcal{R}$
(3) $\psi \notin \mathcal{PR}$

$$\begin{cases} \psi(0,y) = y+1 \\ \psi(x+1,0) = \psi(x,1) \\ \psi(x+1,y+1) = \underbrace{\psi(x, \underbrace{\psi(x+1,y)}_{u})} \end{cases}$$

$(x+1,0) >_{lex} (x,1)$

$(x+1,y+1) >_{lex} (x+1,y)$

$(x+1,y+1) >_{lex} (x,u)$

$(\mathbb{N}^2, \leq_{lex})$ $\qquad (x,y) \leq_{lex} (x',y')$ if $(x < x')$ or
$(x = x'$ and $y \leq y')$

$(9, 10000000) <_{lex} (10, 0)$

$(10, 10000000) >_{lex} (10, 0)$

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

$$f(z) = \begin{cases} 0 & \text{if } z \geq 0 \\ f(z-1) & \text{if } z < 0 \end{cases}$$

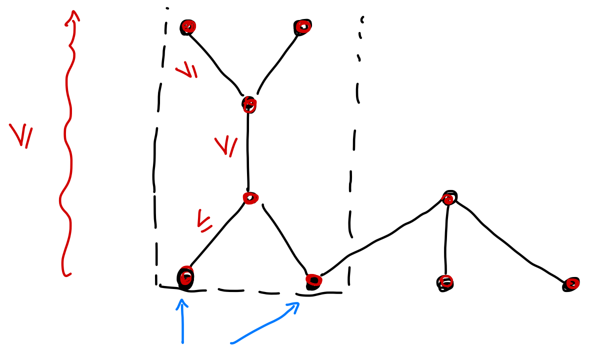
$f(-1)$
|
$f(-2)$
|
$f(-3)$

* partial order

$(D, \leq)$ $\quad$ $\leq$ reflexive $\quad x \leq x$

antisymmetric $\quad x \leq y$ and $y \leq x$ then $x = y$

transitive $\quad x \leq y$ and $y \leq z$ then $x \leq z$



x minimal $\quad \forall y \quad y \leq x \Rightarrow y = x$

$D = \{ (\text{pear}, n), (\text{apple}, n) \mid n \in \mathbb{N} \}$

⋮	
(pear, 2)	(apple, 2)
|	|
(pear, 1)	(apple, 1)
|	|
$\boxed{(\text{pear}, 0)}$	$\boxed{(\text{apple}, 0)}$

minimal

$(x, y) \leq (x', y')$

if $x = x'$ and $y \leq y'$

* Well founded partial order

$(D, \leq)$ is well founded if every $X \subseteq D$ $X \neq \emptyset$ has a minimal element

Es. $\quad \mathbb{N}$ well - founded

$\mathbb{Z}$ not well - founded

$(D, \leq)$ well-founded iff there is no infinite descending chain

up to foundational "details"

$$x_0 > x_1 > x_2 > x_2 \ \ldots\ldots$$

$*$ $(\mathbb{N}^2, \leq_{lex})$ well founded

let $X \subseteq \mathbb{N}^2$ $X \neq \emptyset$

$$x_0 = \min \{x \mid \exists y \ (x, y) \in X\}$$

$$y_0 = \min \{y \mid (x_0, y) \in X\}$$

$$(x_0, y_0) = \min X$$

<u>Induction</u> : $P(n)$ $\forall n \in \mathbb{N}$ $P(n) =$ " $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ "

$\boxed{P(0) \text{ and for all } n \in \mathbb{N} \text{ if } P(n) \text{ then } P(n+1)}$

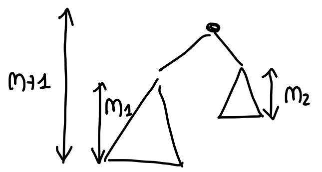
$\Downarrow$

$\boxed{\text{for all } n \in \mathbb{N} \ P(n)}$

<u>Strong Induction</u>

A binary tree with heigth n has at most 2^n leaves

$n \to n+1$



$n_1, n_2 \leq n$

need inductive hypothesis for all $n' \leq n$

$\boxed{\text{for all } n, \text{ assuming } P(n') \text{ for } n' < n \text{ I can deduce } P(n)}$

$\Downarrow$

$\boxed{\text{for all } n \in \mathbb{N} \ P(n)}$

<u>Well-founded induction</u>

$(D, \leq)$ well founded order

$P(x)$ property of elements of D

$$\boxed{\text{if for all } d \in D \text{, assuming } \forall d' < d \;\; P(d') \text{ I can conclude } P(d)}$$

$$\Downarrow$$

$$\boxed{\forall d \in D \quad P(d)}$$

① $\boxed{\psi \text{ is total}}$

$$\forall (x,y) \in \mathbb{N} \qquad \psi(x,y)\downarrow$$

by well-founded induction on $(\mathbb{N}, \leq_{lex})$

<u>proof</u>

let $(x,y) \in \mathbb{N}^2$

assume $\forall \; (x',y') <_{lex} (x,y) \qquad \psi(x',y')\downarrow$

and we show $\psi(x,y)$

$$\begin{cases} \psi(0,y) = y+1 \\ \psi(x+1,0) = \psi(x,1) \\ \psi(x+1,y+1) = \psi(x, \psi(x+1,y)) \end{cases}$$

we distinguish 3 cases

$(x=0) \qquad \psi(x,y) = \psi(0,y) = y+1 \downarrow$

$(x>0,\, y=0) \qquad \psi(x,y) = \psi(\underbrace{x-1,\, 1}_{<_{lex}(x,y)})\downarrow$

hence $\psi(x-1,1)\downarrow$

$(x>0,\, y>0) \qquad \psi(x,y) = \psi(x-1,\, \psi(\underbrace{x,\, y-1}_{<_{lex}(x,y)})) = \psi(\underbrace{x-1,u}_{<_{lex}(x,y)})\downarrow$

$\rightsquigarrow \psi(x,y-1) = u\downarrow \qquad \rightsquigarrow \psi(x-1,u)\downarrow$

$\square$

$(\mathbb{N}^2, \leq_{lex})$

$$\underbrace{(0,0)\;(0,1)\;(0,2)\;\ldots\ldots}_{\mathbb{N}} \;\; \underbrace{(1,0)\;(1,1)\;(1,2)\;\ldots\ldots}_{\mathbb{N}} \;\; \underbrace{(2,0)\;(2,1)\;(2,2)\;\ldots}_{\mathbb{N}}$$

② $\boxed{\varphi \in \mathcal{R} = \mathcal{C}}$

$$\varphi(1,1) = \varphi(0, \underbrace{\varphi(1,0)}_{\varphi(0,1)=2}) = \varphi(0,2) = 3$$

$(1,1,3)$ $(0,2,3)$ $(1,0,2)$ $(0,1,2)$

<u>valid set of triples S</u>: informally

$(x,y,z) \in \mathbb{N}^3$ → $z = \varphi(x,y)$

→ S includes all triples needed to compute φ on (x,y)

formally: $S \subseteq \mathbb{N}^3$ <u>valid</u> if

$$\begin{cases} \varphi(0,y) = y+1 \\ \varphi(x+1,0) = \varphi(x,1) \\ \varphi(x+1,y+1) = \varphi(x, \underbrace{\varphi(x+1,y)}_{u}) \end{cases}$$

① $(0,y,z) \in S \Rightarrow z = y+1$

(2) $(x+1,0,z) \in S \Rightarrow (x,1,z) \in S$

(3) $(x+1,y+1,z) \in S \Rightarrow \exists u \in \mathbb{N}$ s.t. $(x+1,y,u) \in S$, $(x,u,z) \in S$

you can prove $\forall (x,y,z) \in \mathbb{N}^3$

$\varphi(x,y) = z$ iff $\exists S \subseteq \mathbb{N}^3$ a <u>finite</u> valid set of triples s.t. $(x,y,z) \in S$

then $\varphi(x,y) =$ " $\mu \underbrace{(S,z)}_{}. \left(S \subseteq \mathbb{N}^3 \text{ valid finite set of tuples} \wedge (x,y,z) \in S \right)$ "

encode as a number

$S = \{(x_1,y_1,z_1), (x_2,y_2,z_2), \dots, (x_n,y_n,z_n)\}$

$\{\pi(\pi(x_1,y_2), z_1), \quad - - - \quad \pi(\pi(x_n,y_n), z_n)\}$

$k_1 \quad - - - - \quad k_n$

$$\prod_{i=1}^{n} p_i^{k_i}$$

⇝ $\varphi \in \mathcal{R} = \mathcal{C}$

③ $\boxed{\varphi \in \mathcal{PR}}$

successor

$$\begin{cases} x+0 = x \\ x+(y+1) = (x+y)+1 \end{cases}$$ ← successor iterated y times

$$\begin{cases} x * 0 = 0 \\ x*(y+1) = (x*y)+x \end{cases}$$ ← "$+x$" iterated y times

$$\begin{cases} x^0 = 1 \\ x^{y+1} = x^y * x \end{cases}$$ ← "$*x$" iterated y times

⋮

idea: φ brings the above to the "limit"

$$\begin{cases} \varphi(0,y) = y+1 \\ \varphi(x+1,0) = \varphi(x,1) \\ \varphi(x+1,y+1) = \varphi(x,\varphi(x+1,y)) \end{cases}$$

$$\begin{cases} \varphi_0(y) = y+1 \\ \varphi_{x+1}(0) = \varphi_x(1) \\ \varphi_{x+1}(y+1) = \varphi_x(\varphi_{x+1}(y)) \end{cases}$$

consider x as fixed parameter

$$\varphi_x(y) = \varphi(x,y)$$

$$\begin{aligned}
\varphi_{x+1}(y+1) &= \varphi_x(\varphi_{x+1}(y)) \\
&= \varphi_x\,\varphi_x\,(\varphi_{x+1}(y-1)) \\
&= \varphi_x\,\varphi_x\,\varphi_x\,(\varphi_{x+1}(y-2))\ldots) \\
&\;\;\vdots \\
&= \underbrace{\varphi_x\,\cdots\,\varphi_x}_{y+1\text{ times}}\;\underbrace{\varphi_{x+1}(0)}_{\varphi_x(1)} \\
&= \varphi_x^{\,y+2}(0)
\end{aligned}$$

roughly: increasing x to $x+1$ requires iterating the function φ_x

→ increases the number of nested primitive recursion

$\rightarrow$ the full function would require infinitely many nested primitive recursions

Some more ideas....

concretely:

$\varphi_0(y) = y+1$

$\varphi_1(y) = \varphi_0^{y+1}(1) = y+2$

$\varphi_2(y) = \varphi_1^{y+1}(1) = 2(y+1)+1 = 2y+3 \approx 2y$

$\varphi_3(y) = \varphi_2^{y+1}(1) \cong 2^y$

$\varphi_4(y) = \varphi_3^{y+1}(1) \cong 2^{2^{2^{2^{\cdot^{\cdot}}}}}\}y$

eg: $\varphi_0(1) = 2$

$\varphi_2(1) = 5$

$\varphi_3(1) = 13$

$\varphi_4(1) \cong 2^{16}$

$\varphi_4(2) \cong 2^{2^{16}} \cong 10^{6400}$

ONE CAN PROVE: Given a function $f: \mathbb{N}^m \to \mathbb{N} \in PR$ and a program P computing f using only "for-loops" (primitive recursion) if J is the maximum level of nesting of for-loops

$$f(\vec{x}) < \varphi_{J+1}(\max\{x_i\})$$

Now, assume $\varphi \in PR$, let J be the level of nesting of for-loops (of primitive recursive defs) for computing φ

$\forall (x, y)$

$$\varphi(x,y) < \varphi_{J+1}(\max\{x,y\})$$

let $x = y = J+1$

$$\varphi(J+1, J+1) < \varphi_{J+1}(J+1) \qquad = \varphi(J+1, J+1)$$

contradiction

$\Rightarrow \quad \varphi \notin \mathcal{PR}$