COMPUTABILITY (04/11/2024)

* Class of <u>partial recursive functions</u> $\mathcal{R}$

least <u>rich</u> class of functions i.e. least class of functions

→ including the BASIC FUNCTIONS

→ closed under

1. COMPOSITION
2. PRIMITIVE RECURSION
3. UNBOUNDED MINIMALISATION

<u>Theorem</u> : $\mathcal{R} = \mathcal{C}$

<u>proof</u>

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

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

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

Then

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

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

program P (in std form)

$\rightarrow I_1$

$\rightarrow I_2$

$\vdots$

I_s

memory

r_1	r_2	r_3	- - - -	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		
2	0	1	0	0	- - -

$$c = p_1^2 \cdot p_2^0 \cdot p_3^1 \cdot \overbrace{p_4^0 \cdot p_5^0 \cdots}^{1}$$

$$= 2^2 \cdot 3^0 \cdot 5^1 = 20$$

$$(20)_3 = 1$$

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

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

we define C_P, J_P by primitive recursion

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

1	- - -	k	
x_1	- - -	x_k	0 ..

recursion cases

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

by using $C_P(\vec{x}, t) \rightsquigarrow c$
$J_P(\vec{x}, t) \rightsquigarrow j$

$$C_P(\vec{x}, t+1) = \begin{cases} qt\left(p_m^{(c)_m}, c\right) & \text{if } 1 \le j \le \ell(P) \\ & \text{and } I_j = Z(m) \\[2ex] p_m \cdot c & \text{if } 1 \le j \le \ell(P) \\ & \text{and } I_j = S(m) \\[2ex] p_m^{(c)_m} \cdot qt\left(p_m^{(c)_m}, c\right) & \text{if } 1 \le j \le \ell(P) \\ & \text{and } I_j = T(m, m) \\[2ex] c & \text{otherwise} \\ & (j = 0 \text{ or } 1 \le j \le \ell(P) \\ & \text{and } I_j = J(m, m, u)) \end{cases}$$

$$\begin{array}{cc} & \begin{array}{cccc} 1 & & m \xleftarrow{\ 0\ } & \\ \end{array} \\ c & \begin{array}{|c|c|c|c} \hline r_1 & \cdots\cdots & r_m & \qquad\qquad \\ \hline \end{array} \end{array}$$

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

$$J_P(\vec{x}, t+1) = \begin{cases} j+1 & \text{if } 1 \le j < \ell(P) \\ & \text{and } I_j = S(m), Z(m), T(m, m) \\ & \text{or } I_j = J(m, m, u) \ \& \ (c)_m \ne (c)_m \\[2ex] u & \text{if } 1 \le j \le \ell(P) \text{ and} \\ & I_j = J(m, m, u) \text{ and } (c)_m = (c)_m \\ & \text{and } u \le \ell(P) \\[2ex] 0 & \text{otherwise} \end{cases}$$

Hence $C_P, J_P \in \mathcal{R}$ (obtained by composition / primitive recursion from functions in $\mathcal{R}$)

and

$$f(\vec{x}) = \left(C_P\left(\vec{x}, \mu t.\ J_P(\vec{x}, t)\right)\right)_1$$

hence $f \in \mathcal{R}$ $\square$

Primitive Recursive Functions

PR = least class of functions which

→ includes basic functions

→ closed under

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

$$\begin{array}{ccc} PR & \subsetneq & \underline{\mathcal{R}} \cap Tot \\ \| & ? & \| \\ \mathcal{C}_{for} & & \mathcal{C}_{for,\ while} \end{array}$$

Ackermann's Function ψ

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

(1) $\psi \in Tot$
(2) $\psi \in \mathcal{R}$
(3) $\psi \notin PR$

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

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

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

$>_{lex} (x, w)$

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

$(1000\,000,\ 1000\,000\,000) <_{lex} (1000\,001,\ 0)$

$(1000\,000,\ 1000\,000\,000) >_{lex} (1000\,000,\ 0)$

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

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

$f(-1) = f(-2) = f(-3) = \ldots$

* <u>partially ordered set</u> (poset)

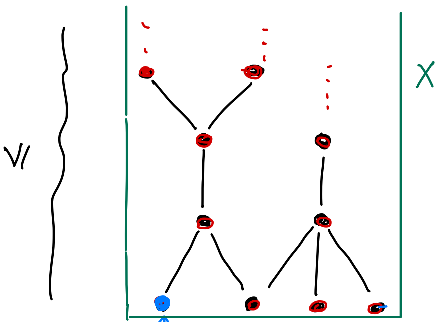
$(D, \leq)$ $\quad \leq$ reflexive $\quad x \leq x$

antisymmetric $\quad x \leq y \ \& \ y \leq x \Rightarrow x = y$

transitive $\quad x \leq y \ \& \ y \leq z \Rightarrow x \leq z$

* <u>Well-founded poset</u>

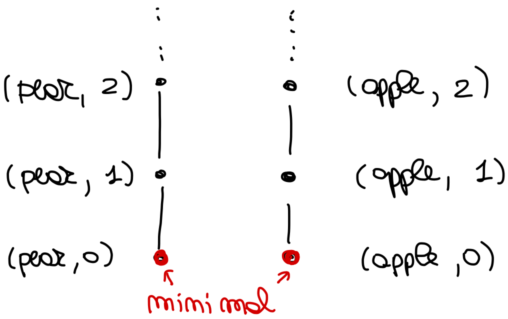
$(D, \leq)$ is well-founded if $\forall X \subseteq D \quad X \neq \emptyset \quad X$ has a minimal element



$d \in X$ minimal $\quad \forall d' \leq d$ then $d' = d$

$D = \{ (pear, n), (apple, n) \mid n \in \mathbb{N} \}$ $\qquad (x, y) \leq (x', y')$

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



$\mathbb{Z}$ not well founded

$\mathbb{N}$ well founded

<u>NOTE</u>: $(D, \leq)$ well-founded iff there is no infinite descending chain

up to foundational "details"

$d_0 > d_1 > d_2 > \cdots$

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

let $X \subseteq \mathbb{N}^2 \quad X \neq \emptyset$ $\qquad x_0 = \min \{ x \mid \exists y \ (x, y) \in X \}$

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

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

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

$P(0)$ and assuming $P(n)$ you prove $P(n+1)$

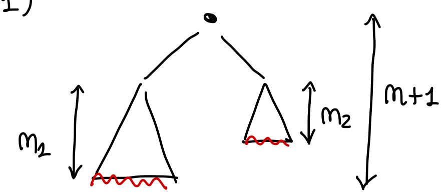
$\Downarrow$

$P(n)$ for all n

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

$(n=0)$ • number of leaves $= 1 \leq 2^0$

$(n \to n+1)$



$n_1, n_2 \leq n$

one needs the inductive hypothesis on n_1 & n_2

* <u>Complete induction</u>

$P(n)$ $n \in \mathbb{N}$

if for all $n \in \mathbb{N}$, assuming $P(n')$ for all $n' < n$
I can deduce $P(n)$

$\Downarrow$

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

• <u>Well-founded induction</u>

$(D, \leq)$ well-founded order

$P(x)$ property of $x \in D$

if for all $d \in D$, assuming $\forall d' < d$ $P(d')$
I can conclude $P(d)$

$\Downarrow$

$\forall d \in D$ $P(d)$

① $\boxed{\psi \text{ is total}}$

$\forall (x,y) \in \mathbb{N}^2 \quad \psi(x,y)\downarrow$

by well-founded induction on $(\mathbb{N}^2, \leq_{lex})$

proof

let $(x,y) \in \mathbb{N}^2$, assume $\forall (x',y') <_{lex} (x,y) \quad \psi(x',y')\downarrow$

we want to show $\psi(x,y)\downarrow$

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

3 cases

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

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

$(x,1) <_{lex} (x+1,0)$ by ind. hyp $\Rightarrow \psi(x,1)\downarrow$

$(x,y>0) \quad \psi(x,y) = \psi(x-1, \psi(\underbrace{x,y-1}_{<_{lex}(x,y)})) = \psi(\underbrace{x-1,u}_{<_{lex}(x,y)}) \downarrow$ by ind. hyp

$\Rightarrow \psi(x,y-1)\downarrow = u$ by ind. hyp

$\square$

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

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

$\leq$

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

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

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

valid set of triples S : informally

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

$\rightarrow \quad S$ includes all triples needed to compute φ on (x, y)

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

formally : $\quad S \subseteq \mathbb{N}^3$ valid if

(1) $(0, y, z) \in S \quad \Rightarrow \quad z = y+1$

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

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

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

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

then

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

$S = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_m, y_m, z_m)\}$

$\{\pi(\pi(x_1, y_1), z_1), \quad \ldots \quad \pi(\pi(x_m, y_m), z_m)\}$

$k_1 \quad \ldots \quad k_m$

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

$\leadsto \quad \varphi \in \mathcal{R} = \mathcal{C}$

③ $\boxed{\psi \in PR}$

successor

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

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

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

$\vdots$

idea: ψ brings the above to the "limit"

$$\begin{cases} \psi(0,y) = y+1 \\ \psi(x+1,0) = \psi(x,1) \\ \psi(x+1,y+1) = \psi(x,\psi(x+1,y)) \end{cases} \qquad \begin{cases} \psi_0(y) = y+1 \\ \psi_{x+1}(0) = \psi_x(1) \\ \psi_{x+1}(y+1) = \psi_x(\psi_{x+1}(y)) \end{cases}$$

consider x as fixed parameter

$$\psi_x(y) = \psi(x,y)$$

$$\begin{aligned} \psi_{x+1}(y+1) &= \psi_x(\psi_{x+1}(y)) \\ &= \psi_x\,\psi_x\,(\psi_{x+1}(y-1)) \\ &= \psi_x\,\psi_x\,\psi_x\,(\psi_{x+1}(y-2))\dots) \\ &\vdots \\ &= \underbrace{\psi_x \cdots \psi_x}_{y+1 \text{ times}}\ \underbrace{\psi_{x+1}(0)}_{\psi_x(1)} \\ &= \psi_x^{y+2}(0) \end{aligned}$$

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

$\rightarrow$ increases the number of nested primitive recursion

→ the full function would require infinitely many nested primitive recursions

Some more ideas....

concretely:

$\varphi_0(y) = y+1$

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

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

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

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

es: $\varphi_0(1) = 2$

$\varphi_2(1) = 5$

$\varphi_3(1) = 13$

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

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

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 $x = y = J+1$

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

contradiction

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