Computability (13/12/2021)

1st RECURSION THEOREM

```
type T = func (int) int

func succ (f : T) T
    res = func (x int) int
            return f(x) + 1

    return res
```

* functionals

$\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$ where $\mathcal{F}(\mathbb{N}^k) = \{ f \mid f : \mathbb{N}^k \to \mathbb{N} \}$

total

When is Φ RECURSIVE? (to be read as "computable")

Example: $succ : \mathcal{F}(\mathbb{N}^1) \to \mathcal{F}(\mathbb{N}^1)$

$f \mapsto succ(f)$

$succ(f)(x) = f(x) + 1$

Example

factorial

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

$$fact(n) = \begin{cases} 1 & \text{if } n = 0 \\ fac(n-1) \cdot n & \text{if } n > 0 \end{cases}$$

$\Phi_{fac} : \mathcal{F}(\mathbb{N}^1) \to \mathcal{F}(\mathbb{N}^1)$

$$\Phi_{fac}(f)(n) = \begin{cases} 1 & \text{if } n = 0 \\ f(n) \cdot (n-1) & \text{if } n > 0 \end{cases}$$

What is the factorial function? It is a function

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

$$\Phi_{fac}(f) = f$$

fixed point of Φ_{fact} (unique in this case)

<u>Ex.</u>

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ f(n+1) & \text{if } n > 0 \end{cases}$$

$f(2) = ?$

$$f_k(n) = \begin{cases} 0 & \text{if } x = 0 \\ k & \text{otherwise} \end{cases}$$

is a "solution" for any $k \in \mathbb{N}$ (*)

$$\boxed{f(n) = \begin{cases} 0 & \text{if } x = 0 \\ \uparrow & \text{otherwise} \end{cases}}$$

We like this (**)

$\Phi : \mathcal{F}(\mathbb{N}') \to \mathcal{F}(\mathbb{N}')$

$$\Phi(f)(n) = \begin{cases} 0 & \text{if } n = 0 \\ f(n+1) & \text{if } n > 0 \end{cases}$$

All the functions (*), (**) are fixed points, but we want (**).

* <u>Ackermann</u>

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

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

define $\Psi : \mathcal{F}(\mathbb{N}^2) \to \mathcal{F}(\mathbb{N}^2)$

$$\begin{cases} \boxed{\Psi(f)}(0,y) = y+1 \\ \boxed{\Psi(f)}(x+1,0) = \underline{f}(x,1) \\ \boxed{\Psi(f)}(x+1,y+1) = \underline{f(x, \underline{f(x+1,y)})} \end{cases}$$

ψ is a "   " fixed point of Ψ

⋮

* What is a RECURSIVE (computable) FUNCTIONAL

$$\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$$

→ input / output are infinite objects

→ idea : We want that

$\Phi(f)(\vec{x})$ is computable

→ using a "finite amount of information" about f

↳ value of f over a finite number of inputs

→ the above finite amount of information is used in a computable way

in order to compute $\Phi(f)(\vec{x})$

we use $\vartheta \subseteq f$ finite subfuction

in a computable way

i.e. $\Phi(f)(\vec{x}) = \varphi(\tilde{\vartheta}, \vec{x})$

computable

$\boxed{\text{NOTE}}$

finite functions can be encoded as numbers

$$\vartheta(x) = \begin{cases} y_1 & \text{if } x = x_1 \\ \vdots & \\ y_n & \text{if } x = x_n \\ \uparrow & \text{otherwise} \end{cases}$$

$y_i = 0$

encoding $\tilde{\vartheta} \in \mathbb{N}$ $\qquad \tilde{\vartheta} = \prod_{i=1}^{n} p_{x_i+1}^{y_i+1}$ $\qquad p_{x_i+1}$

$\downarrow$

$x \in dom(\vartheta)$ iff $(\tilde{\vartheta})_{x+1} \neq 0$

if $x \in dom(\vartheta)$ $\qquad \vartheta(x) = (\tilde{\vartheta})_{x+1} \dot{-} 1$

<u>Def</u> (Recursive Functional): A functional $\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$ is RECURSIVE if there exists a <u>computable function</u> $\varphi : \mathbb{N}^{h+1} \to \mathbb{N}$ such that $\forall f \in \mathcal{F}(\mathbb{N}^k)$ $\forall \vec{x} \in \mathbb{N}$ $\forall y \in \mathbb{N}$

$$\Phi(f)(\vec{x}) = y \quad \text{iff} \quad \varphi(\tilde{\vartheta}, \vec{x}) = y$$

for some $\vartheta \subseteq f$ finite

OBSERVATION: Let $\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$ be a recursive functional.

For all $f \in \mathcal{F}(\mathbb{N}^k)$ if f is computable

the $\Phi(f)$ is computable

$\Downarrow$

Note : let $\Phi : \mathcal{F}(\mathbb{N}^1) \to \mathcal{F}(\mathbb{N}^1)$ be recursive

if f computable $\rightsquigarrow$ $\Phi(f)$ is computable

$f = \varphi_e = \varphi_{e'}$, $e \in \mathbb{N}$ $\qquad \Phi(f) = \varphi_a = \varphi_{a'}$, $a \in \mathbb{N}$

it induces a function over programs

$$h : \mathbb{N} \longrightarrow \mathbb{N}$$

$$e \longmapsto h(e) = a$$

$\boxed{\text{extensional}}$: $\forall e, e' \in \mathbb{N}$

if $\varphi_e = \varphi_{e'}$ then $\varphi_{h(e)} = \varphi_{h(e')}$

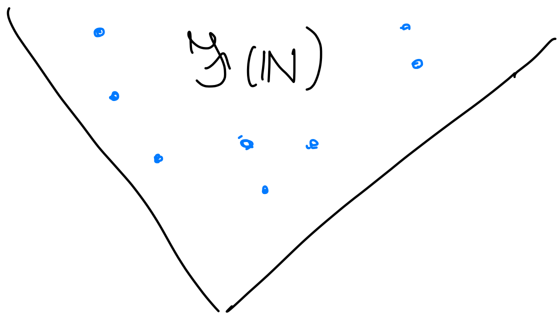
Myhill - Shepherdson Theorem

1. Let $\Phi : \mathcal{F}(\mathbb{N}) \to \mathcal{F}(\mathbb{N})$ be a recursive functional then there exists a total computable function $h : \mathbb{N} \to \mathbb{N}$ extensional s.t. $\forall e \in \mathbb{N}$

$$\Phi(\varphi_e) = \varphi_{h(e)}$$

2. Let $h : \mathbb{N} \to \mathbb{N}$ be a total computable extensional function Then there is a unique $\Phi : \mathcal{F}(\mathbb{N}) \to \mathcal{F}(\mathbb{N})$ recursive functional such that

$$\Phi(\varphi_e) = \varphi_{h(e)} \qquad \forall e \in \mathbb{N}$$



$\mathcal{C}$ computable

I Recursion theorem (Kleene)

Let $\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^k)$ be a recursive functional

Then there exists a least fixed point of Φ, call it $f_\Phi : \mathbb{N}^k \to \mathbb{N}$

and f_Φ is computable

i.e.

(i) $\Phi(f_\Phi) = f_\Phi$

(ii) $\forall g \in \mathcal{F}(\mathbb{N}^k)$ if $\Phi(g) = g$ then $f_\Phi \subseteq g$

(iii) f_Φ is computable

Example: Ackermann

define $\Psi : \mathcal{F}(\mathbb{N}^2) \to \mathcal{F}(\mathbb{N}^2)$

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

recursive functional

ψ is the least fixed point of Ψ

$\uparrow$ (unique since it is total)

$\Rightarrow$ computable by first recursion theorem.

<u>Exercise</u> : Let $A = \{ x \mid \varphi_x(x) = x^2 \}$

is it recursive / r.e. ?
NO yes

what about $\bar{A}$?

conjecture $\left.\begin{array}{l} A \text{ is r.e.} \\ \text{not recursive} \end{array}\right\} \Rightarrow \begin{array}{l} \bar{A} \text{ not r.e.} \\ \Downarrow \\ \text{not recursive} \end{array}$

$\boxed{A \text{ (probably) not saturated} \qquad \begin{array}{l} A = \{ x \mid \varphi_x \in \mathcal{A} \} \\ \mathcal{A} = \{ f \mid f(\ ) = (\ )^2 \} \end{array}}$

* A r.e.

$$sc_A(x) = \begin{cases} 1 & x \in A \\ \uparrow & \text{otherwise} \end{cases} = \mathbb{1}\left(\underbrace{\mu z. \, |\varphi_x(x) - x^2|}_{0 \text{ if } \varphi_x(x) = x^2, \ \uparrow \text{ otherwise}} \right)$$

$$= \mathbb{1}\left(\mu z. \ |\psi_U(x, x) - x^2| \right)$$

computable

* A not recursive

$K \leq_m A$

$$g(x, y) = \begin{cases} y^2 & x \in K \\ \uparrow & x \notin K \end{cases}$$

$$= y^2 \cdot sc_K(x)$$

computable
$\Downarrow$

By smn $\exists s : \mathbb{N}^2 \to \mathbb{N}$ total computable such that

$$\varphi_{s(x)}(y) = g(x, y) = \begin{cases} y^2 & x \in K \\ \uparrow & x \notin K \end{cases}$$

$\boxed{s \text{ reduces } \quad K \leq_m A}$

* if $x \in K$ then $s(x) \in A$

  if $x \in K$ then $\varphi_{s(x)}(y) = y^2 \ \forall y$. In particular $\varphi_{s(x)}(s(x)) = s(x)^2$

  $\Rightarrow s(x) \in A$

* if $x \notin K$ then $s(x) \notin A$

  if $x \notin K$ then $\varphi_{s(x)}(y) \uparrow \ \forall y$ Hence $\varphi_{s(x)}(s(x)) \uparrow \neq s(x)^2$

  $\Rightarrow s(x) \notin A$

Hence A is not recursive. $\square$