COMPUTABILITY (16/12/2024)

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 (operators)

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

total

What is a ~~computable~~ recursive functional?

Example: successor

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

$$f \mapsto succ(f)$$

defined by $\Phi_{succ}(f)(x) = f(x) + 1 \quad \forall x$

Example: factorial

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

$$fact(x) = \begin{cases} 1 & \text{if } x = 0 \\ x * fact(x-1) & \text{if } x > 0 \end{cases}$$

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

$$f \mapsto \Phi_{fact}(f)$$

defined $\quad \Phi_{fact}(f)(x) = \begin{cases} 1 & \text{if } x = 0 \\ x * f(x-1) & \text{if } x > 0 \end{cases}$

then the factorial is a fixed point of Φ_{fact}

i.e. a function fact

$$\Phi_{fact}(fact) = fact$$

in this case
the fixpoint exists
and it is unique

Example :

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

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

$f(0) = 0$

$f(2) = ?$

functional $\quad \Phi : \mathcal{F}(\mathbb{N}^1) \to \mathcal{F}(\mathbb{N}^1)$

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

there are <u>many fixed points</u> for Φ

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

WE
MEAN
THIS

WHY ?

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

* Ackermann's function

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

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

functional

$$\Phi_{Ack} : \mathcal{F}(\mathbb{N}^2) \to \mathcal{F}(\mathbb{N}^2)$$

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

↳ Ackermann's function is a "special" fixed point ...

What is a recursive (computable) functional

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

we ask that for each $f \in \mathcal{F}(\mathbb{N}^k)$ $\forall \vec{x} \in \mathbb{N}^h$

$$\underbrace{\Phi(f)}_{\in\, \mathcal{F}(\mathbb{N}^h)}(\vec{x}) \quad \text{is computable}$$

1) by using a finite amount of information about f

i.e. we use the value of f over a finite set of inputs

2) and this finite amount of information is used in an "effective way"

we look only at a finite subfunction $\vartheta \subseteq f$

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

→ we can refer only to a finite subfunction $\vartheta \subseteq f$

in a computable way

i.e. there is a computable function φ such that

↑ in the old sense

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

Note : we can encode a finite function ϑ as a number

$$\vartheta \longrightarrow \tilde{\vartheta} \in \mathbb{N}$$

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

$$\tilde{\vartheta} = \prod_{i=1}^{n} p_{x_i+1}^{y_i+1}$$

given the encoding

$$\begin{cases} x \in dom(\vartheta) & \text{iff} \quad (\tilde{\vartheta})_{x+1} \neq 0 \\ \text{if } x \in dom(\vartheta) & \text{then} \quad \vartheta(x) = (\tilde{\vartheta})_{x+1} \dot{-} 1 \end{cases}$$

Def (Recursive functional)

A functional $\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$ is recursive if

there is a computable function $\varphi : \mathbb{N}^{h+1} \to \mathbb{N}$ such that

for all $f \in \mathcal{F}(\mathbb{N}^k)$ $\quad \forall \vec{x} \in \mathbb{N}^h \quad \forall y \in \mathbb{N}$

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

for some $\vartheta \subseteq f$

All functionals considered above are recursive.

OBSERVATION: Let $\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$ be a recursive functional and $f \in \mathcal{F}(\mathbb{N}^k)$ computable then $\Phi(f) \in \mathcal{F}(\mathbb{N}^h)$ is computable

OBSERVATION: Let $\Phi : \mathcal{F}(\mathbb{N}^1) \to \mathcal{F}(\mathbb{N}^1)$ be a recursive functional

if $f : \mathbb{N} \to \mathbb{N}$ computable then $\Phi(f)$ is computable

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

$= \varphi_{e'}$ $\qquad\qquad$ $= \varphi_{a'}$

hence Φ induces a function over programs

$$h_\Phi : \mathbb{N} \to \mathbb{N}$$
$$e \mapsto \overset{a}{\overset{\shortparallel}{h_\Phi(e)}} \qquad \text{s.t.} \qquad \Phi(\varphi_e) = \varphi_{h_\Phi(e)}$$

extensional: $\forall e, e'$ s.t. $\varphi_e = \varphi_{e}$ then

$$\varphi_{h_\Phi(e)} = \varphi_{h_\Phi(e')}$$

Myhill - Shepherdson's Theorem

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

Then there is a total computable extensional function

$$h_\Phi : \mathbb{N} \to \mathbb{N}$$

such that $\forall e \in \mathbb{N}$

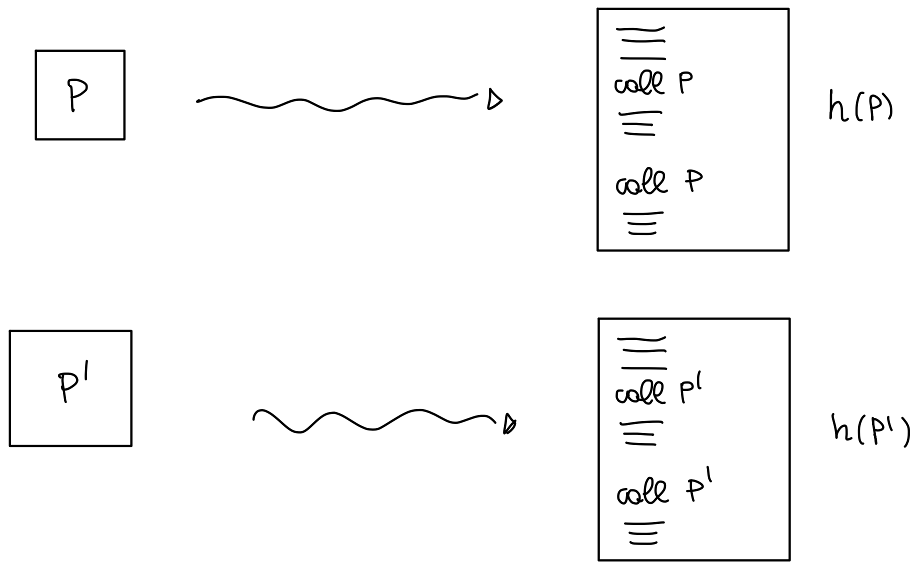
$$\Phi(\varphi_e^{(k)}) = \varphi^{(\ell)}_{h_\Phi(e)}$$

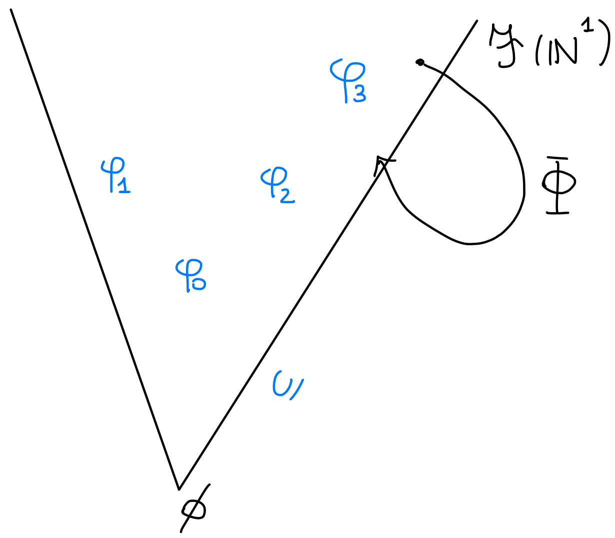
(2) Let $h : \mathbb{N} \to \mathbb{N}$ be a total computable extensional function.

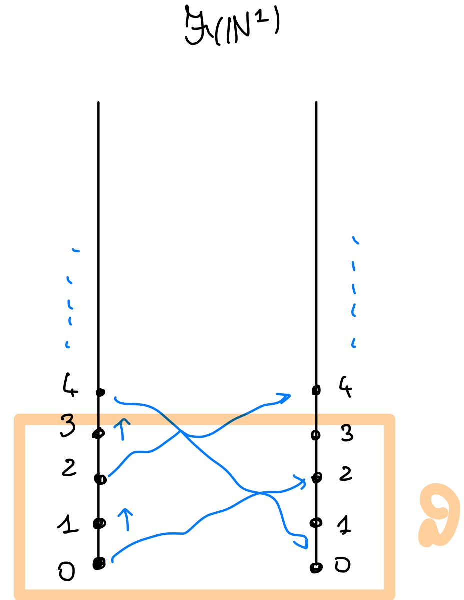
Then there is a unique recursive functional $\Phi_h : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^\ell)$

s.t. $\forall e \in \mathbb{N}$ $\qquad \Phi_h(\varphi_e^{(k)}) = \varphi^{(\ell)}_{h(e)}$

* extensional program transformation $h : \mathbb{N} \to \mathbb{N}$







I RECURSION THEOREM

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

Then Φ has a least fixed point $f_\Phi : \mathbb{N}^k \to \mathbb{N}$ is computable

i.e.

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

(ii) $\forall g : \mathbb{N}^k \to \mathbb{N}$ s.t. $\Phi(g) = g$ then $f_\Phi \subseteq g$

(iii) f_Φ is computable.

× <u>Ackermann's function</u>

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

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

functional

$$\Phi_{Ack} : \mathcal{F}(\mathbb{N}^2) \to \mathcal{F}(\mathbb{N}^2)$$

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

recursive functional

the Ackermann function is the least fixpoint of Φ_{Ack}, which exists and is computable since Φ_{Ack} is recursive. (BY I[st] recursion theorem)
[unique fixpoint since it is total]

<u>Example</u> :

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

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

$$f(0) = 0$$

$$f(2) = ?$$

RECURSIVE FUNCTIONAL

functional $\Phi : \mathcal{F}(\mathbb{N}^1) \to \mathcal{F}(\mathbb{N}^1)$

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

there are many fixed points for Φ

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

WE MEAN THIS

WHY ?

$$f_K(x) = \begin{cases} 0 & \text{if } x=0 \\ K & \text{otherwise} \end{cases} \qquad \forall K$$

because it is the least fix point

$(f \subsetneq f_K \quad \forall K \in \mathbb{N})$

EXERCISE : Find a recursive functional with non-computable fix points

Example : minimalisation

$f : \mathbb{N}^{K+1} \to \mathbb{N}$

$\mu y. f(\vec{x},y) : \mathbb{N}^K \to \mathbb{N}$

can be seen as a least fixpoint

$\Phi : \mathcal{F}(\mathbb{N}^{K+1}) \to \mathcal{F}(\mathbb{N}^{K+1})$

$$\Phi(g)(\vec{x},y) = \begin{cases} y & \text{if } f(\vec{x},y) = 0 \\ g(\vec{x},y+1) & \text{if } f(\vec{x},y)\downarrow \neq 0 \\ \uparrow & \text{if } f(\vec{x},y)\uparrow \end{cases}$$

Φ is recursive functional, hence its least fix point is computable

(by I recursion theorem)

$$m(\vec{x},y) = \mu z \geq y \,.\, f(\vec{x},z)$$

hence

$$m(\vec{x},0) = \mu z.\, f(\vec{x},0)$$