COMPUTABILITY (15/12/2025)

1st RECURSION THEOREM

```
type T = funct(int) int

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

* functionals (operators)

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

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

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

total

What is a ~~computable~~ recursive functional ?

Example : successors

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

$$f \mapsto \text{succ}(f) \qquad \text{defined as}$$

$$\text{succ}(f)(x) = f(x) + 1$$

Example : factorial

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

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

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

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

where

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

then $fact : \mathbb{N} \to \mathbb{N}$ is a fixed point, i.e. a function $f : \mathbb{N} \to \mathbb{N}$ s.t.

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

unique in this case

<u>Example</u> : $f : \mathbb{N} \to \mathbb{N}$

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

what are

$$f(0) = 0$$

$$f(1) = ?$$

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 fix points of Φ

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

This is what we want!

WHY?

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

$k \in \mathbb{N}$ fixed

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

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

ψ (Ackermann's function) is a "special" fixpoint of Φ_{Ack}.

* What is a computable/recursive functional?

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

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

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

$\mathcal{F}(\mathbb{N}^h)$

(1) using only a finite amount of information on f

i.e. using the value of f on finitely many inputs

we use a finite subfunction of f

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

more precisely, for computing $\Phi(f)(\vec{x})$

we use a finite subfunction $\vartheta \subseteq f$ in computable way

i.e. there is a computable function φ (in the old sense)

$$\Phi(f)(\vec{x}) = \text{“}\varphi(\vartheta, \vec{x})\text{''}$$

encoding of ϑ as a number

$$= \varphi(\tilde{\vartheta}, \vec{x})$$

<u>NOTE</u>: finite functions can be encoded as numbers

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

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

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

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

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

<u>Def</u>. (Recursive functional): A functional $\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$ is <u>recursive</u> if there is a total computable function

$$\varphi : \mathbb{N}^{h+1} \to \mathbb{N}$$

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

$\Phi(f)(\vec{x}) = y$ iff there exists $\vartheta \subseteq f$, ϑ finite s.t. $\varphi(\tilde{\vartheta}, \vec{x}) = y$

<u>Examples</u>: All the functionals introduced above are recursive

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

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

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

$f = \varphi_a$ $\quad a \in \mathbb{N}$ $\qquad\qquad \Phi(f) = \varphi_b$ $\quad b \in \mathbb{N}$

$\;\;\; = \varphi_{a'}$ $\quad a' \in \mathbb{N}$ $\qquad\qquad\qquad\; = \varphi_{b'}$

hence Φ induces a function from programs to programs

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

$$a \mapsto h_\Phi(a) = b \qquad \text{s.t.} \qquad \Phi(f) = \varphi_{h_\Phi(a)}$$

extensional : $\forall a, a' \in \mathbb{N}$ s.t. $\varphi_a = \varphi_{a'}$ then $\varphi_{h_\Phi(a)} = \varphi_{h_\Phi(a')}$

Myhill - Shepherdson's Theorem

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

Then there is a total computable function $h_\Phi : \mathbb{N} \to \mathbb{N}$ s.t.

$\forall a \in \mathbb{N}$ $\qquad \Phi(\varphi_a^{(k)}) = \varphi_{h_\Phi(a)}^{(h)}$ $\qquad$ and h_Φ is extensional.

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

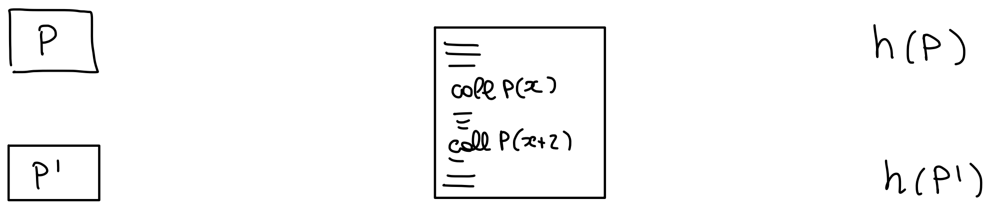
Then there is a unique recursive functional

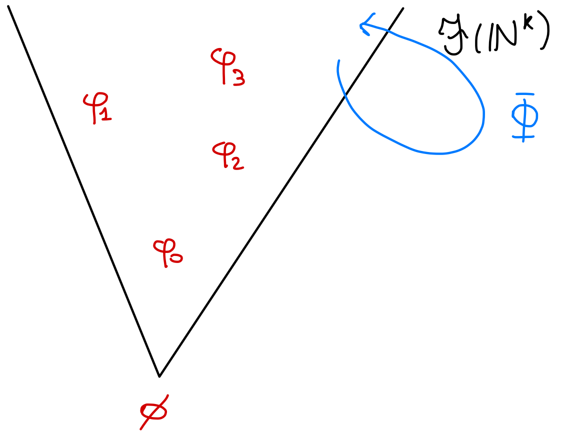
$$\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^h)$$

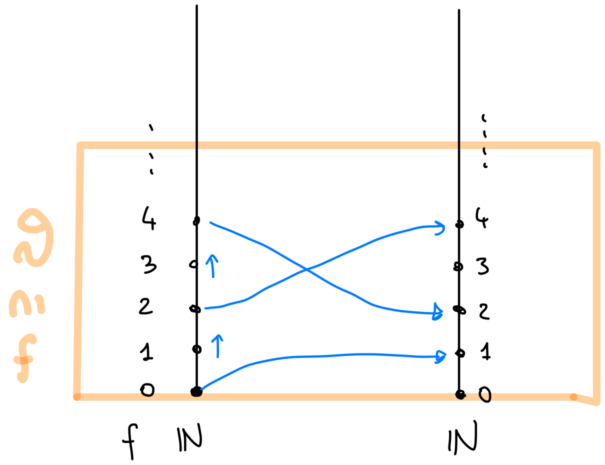
s.t. for all $a \in \mathbb{N}$

$$\Phi(\varphi_a^{(k)}) = \varphi_{h(a)}^{(h)}$$

- extensional program transformation







1^{st} RECURSION THEOREM

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

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

i.e.

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

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

(iii) f_Φ is computable

Example : Ackermann's function

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

the Ackermann function is the least fixpoint of Φ_{Ack}

this exists and it is computable by the 1st recursion theorem

unique (since it is total)

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

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

what are

$f(0) = 0$

$f(1) = ?$

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 many fixpoints of Φ

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

This is what we want!

WHY?

because it is the least fixpoint

$$f_k(m) = \begin{cases} 0 & \text{if } m = 0 \\ k & \text{if } m > 0 \end{cases}$$

EXERCISE : Example of a recursive functional with non-computable fixpoints

SOLUTION :

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

$\Phi(f) = f$ identity $\quad \Phi(f)(x) = f(x) \quad$ recursive

- least fixpoint $lfp(\Phi) = \emptyset$ (always undefined function) computable
- every f is a fixpoint, e.g. $f = \chi_K$ not computable.

Example : mimimalisation

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

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

we can obtain it as a least fixpoin of $\Phi : \mathcal{F}(\mathbb{N}^{k+1}) \to \mathcal{F}(\mathbb{N}^{k+1})$

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

depends on f

recursive

the least fixpoint $m : \mathbb{N}^{k+1} \to \mathbb{N}$

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

computable by 1st recursion theorem

in particular

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