COMPUTABILITY (14/12/2023)

I RECURSION THEOREM

```
type T = func (int) int

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

* <u>functionals</u>

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

total

What is a functional Φ RECURSIVE (computable) ?

Example : successor

Given a function f, in order to compute the new function succ(f) over some x you need the value of f over a single point x.

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

$$f \mapsto succ(f)$$

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

Example : factorial

To compute the value of the transformed function over x you need no either information over the input (x=0) or just the value of the function f over just one point x.

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

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

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

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

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

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

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

in this case the fixpoint exists unique

<u>Example</u> :

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

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

$f(0) = 0$

$f(2) = ?$

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 fixed points for Φ

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

This is what a programmer means

$$f_K(n) = \begin{cases} 0 & \text{if } x=0 \\ K & \text{if } x>0 \end{cases} \quad \text{for } K \in \mathbb{N}$$

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

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

ψ akermann's function is some "special" fixpoint of Ψ.

* What is a recursive (computable) functional ?

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

we ask that for all $\vec{x} \in \mathbb{N}^h$

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

We are transforming the original function f (with k functions parameters) in a new one (with h parameters)

→ using a finite amount of information on f

i.e. values of f over a finite number of inputs

→ the finite amount of information is procened in an "effective way"

more precisely, in order to compute $^*\Phi(f)(\vec{x})$ (* transformed function over f)

→ we use a finite subfunction $\vartheta \subseteq f$

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

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

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

encoding of ϑ

NOTE : finite functions can be encoded as numbers

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

$$\vartheta(x) = \begin{cases} y_1 & \text{if } x = x_1 \\ y_2 & \text{if } x = x_2 \\ \vdots & \\ y_m & \text{if } x = x_m \\ \uparrow & \text{otherwise} \end{cases}$$

The product give us a corresponding prime number (depending on the exponent) if the function is valid on the input x and

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

+1 because the prime numbers that we use begin from 1 and not 0

given the above

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

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

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

$\varphi : \mathbb{N}^{h+1} \to \mathbb{N}$ * such that for all $f \in \mathcal{F}(\mathbb{N}^k)$ [for all possible original input functions]

* which takes the new parameters

for all $\vec{x} \in \mathbb{N}^h$ [for all possible inputs to the transformed function]

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

All the functionals that we considered above are recursive.

(IDEA on the successor and factorial examples)

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

and $f \in \mathcal{F}(\mathbb{N}^k)$ (f is the function took as input)

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

<u>OBSERVATION</u>: 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 $\Phi(f) : \mathbb{N} \to \mathbb{N}$ computable

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

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

hence Φ induces a function over programs

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

$$e \mapsto h_\Phi(e) = a \quad \text{s.t.} \quad \Phi(\varphi_e) = \varphi_{h_\Phi(e)}$$

<u>extensional</u>: $\forall e, e' \in \mathbb{N}$ s.t. $\varphi_e = \varphi_{e'}$

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

<u>Myhill - Shepherdson's theorem</u>

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

Then there exists a total computable function $h_\Phi : \mathbb{N} \to \mathbb{N}$ st.

$\forall e \in \mathbb{N} \qquad \Phi\left(\varphi_e^{(k)}\right) = \varphi^{(i)}_{h_\Phi(e)}$ and h_Φ is extensional

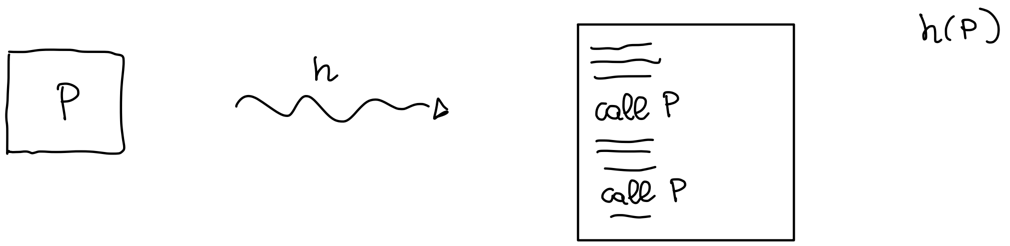
(2) Let $h : \mathbb{N} \to \mathbb{N}$ be a total computable function and h extensional. Then there is a unique recursive functional

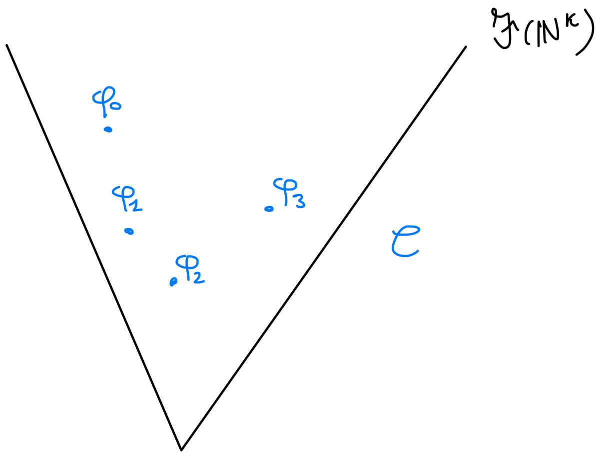
$$\Phi : \mathcal{F}(\mathbb{N}^k) \to \mathcal{F}(\mathbb{N}^i)$$

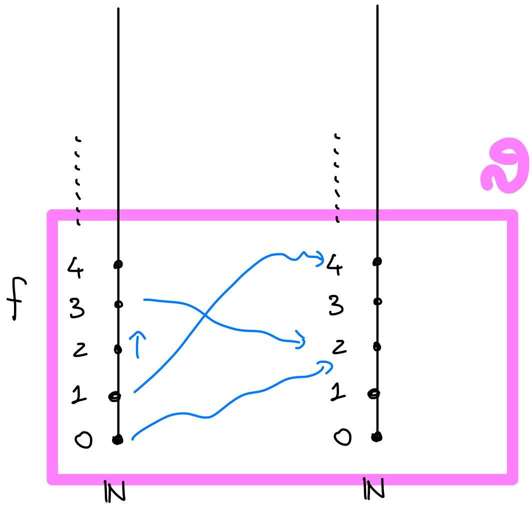
s.t. for all $e \in \mathbb{N}$

$$\Phi\left(\varphi_e^{(k)}\right) = \varphi^{(i)}_{h(e)}$$

- extensional program transformation h







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 $f_\Phi : \mathbb{N}^k \to \mathbb{N}$ which is computable

i.e.

(i) $\Phi(f_\Phi) = f_\Phi$ (definition of a fixed point)

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

(iii) f_Φ is computable

Example : Ackermann's function

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

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

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

recursive functional

the Ackermann function ψ is the least fixed point of Ψ

which exists and is computable by I Recursion Theorem.

(fixpoint is unique since it is total)

Example :

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

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

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

there are many fixed points for Φ

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

We want this because it is the least fix point !!!

$(f \subseteq f_K \quad \forall K)$

$$f_K(n) = \begin{cases} 0 & \text{if } x=0 \\ K & \text{if } x>0 \end{cases}$$

for $K \in \mathbb{N}$

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 fixed point

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

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

least fixed point is $m : \mathbb{N}^{K+1} \to \mathbb{N}$

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

computable by I Recursion Theorem

hence

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