

I RECURSION THEOREM

type  $T = \text{func } (\text{int}) \text{ int}$

func succ ( $f: T$ )  $T$

res = func ( $x: \text{int}$ ) int

return  $f(x) + 1$

return res

\* functionals

$\Phi : \mathcal{F}(\mathbb{N}^k) \rightarrow \mathcal{F}(\mathbb{N}^n)$

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

total

What is a functional  $\Phi$  RECURSIVE (computable) ?

Example : successor

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

$f \mapsto \text{succ}(f)$

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

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

Example : factorial

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

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

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  $x$ .

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

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

where

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

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

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

in this case the fixpoint exists unique

Example :

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

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

$$f(0) = 0$$

$$f(2) = ?$$

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

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

there are many fixed points for  $\Phi$

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

← This is what a programmer means

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

for  $k \in \mathbb{N}$

\* Ackermann's function

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

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

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

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

$\psi$  Ackermann's function is some "special" fixpoint of  $\Psi$ .

\* What is a recursive (computable) functional?

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

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

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

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

→ 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 processed 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  $\varphi$  computable (in the old sense)

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

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

← encoding of  $\vartheta$

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 gives 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 start from 1 and not 0

given the above

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

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

Def (Recursive functional) : A functional  $\Phi : \mathcal{F}(\mathbb{N}^k) \rightarrow \mathcal{F}(\mathbb{N}^n)$

is recursive if there is a total computable function

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

\* which takes the new parameters  $\text{for all } \vec{x} \in \mathbb{N}^n \quad \text{[for all possible inputs to the transformed function]}$

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

All the functionals that we considered above are recursive.

(IDEA on the successor and factorial examples)

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

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

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

OBSERVATION: Let  $\Phi : \mathcal{F}(IN^1) \rightarrow \mathcal{F}(IN^1)$  be a recursive functional

if  $f: IN \rightarrow IN$  is computable then  $\Phi(f): IN \rightarrow IN$  computable

$\downarrow$   $\downarrow$   
 $f = \varphi_e \quad e \in IN$   $\Phi(f) = \varphi_a \quad a \in IN$   
 $\quad \varphi_{e'} \quad e' \in IN$   $\quad \varphi_{a'}$

hence  $\Phi$  induces a function over programs

$$h_\Phi : IN \rightarrow IN$$

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

extensional:  $\forall e, e' \in IN \quad \text{s.t.} \quad \varphi_e = \varphi_{e'}$

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

Myhill - Shepherson's theorem

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

Then there exists a total computable function  $h_\Phi : IN \rightarrow IN$  s.t.

$$\forall e \in IN \quad \Phi(\varphi_e^{(k)}) = \varphi_{h_\Phi(e)}^{(i)} \quad \text{and } h_\Phi \text{ is extensional}$$

(2) Let  $h: IN \rightarrow IN$  be a total computable function and

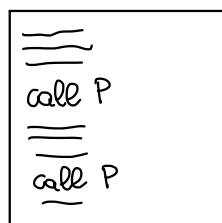
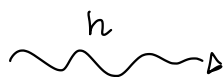
$h$  extensional. Then there is a unique recursive functional

$$\Phi : \mathcal{F}(IN^k) \rightarrow \mathcal{F}(IN^i)$$

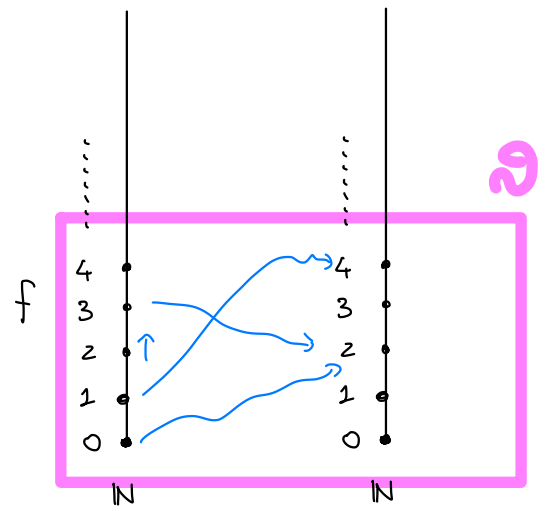
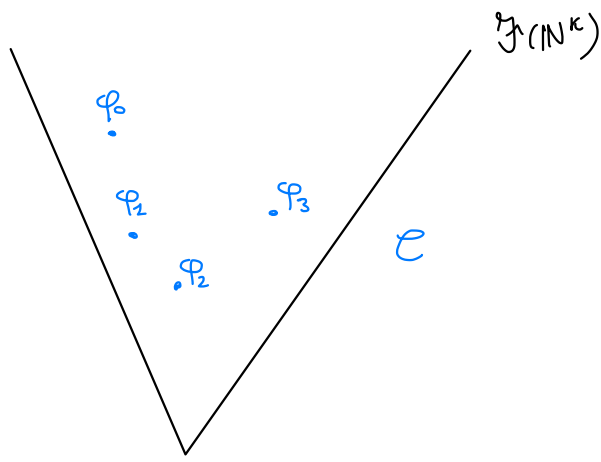
s.t. for all  $e \in IN$

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

• extensional program transformation  $h$



$h(P)$



## I Recursion Theorem :

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

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

Example : Ackermann's function

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

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

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

recursive functional

the Ackermann function  $\psi$  is the least fixed point of  $\Psi$  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  $\Phi : \mathcal{H}(\mathbb{N}^1) \rightarrow \mathcal{H}(\mathbb{N}^1)$

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

there are many fixed points for  $\Phi$

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

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

$$(f \leq f_k \quad \forall k)$$

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

for  $k \in \mathbb{N}$

Example : minimisation

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

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

can be seen as a least fixed point

$$\Phi : \mathcal{H}(\mathbb{N}^{k+1}) \rightarrow \mathcal{H}(\mathbb{N}^{k+1})$$

$$\Phi(g)(\vec{x}, y) = \begin{cases} y \\ g(\vec{x}, y+1) \\ \uparrow \end{cases}$$

$$\text{if } f(\vec{x}, y) = 0$$

$$\text{if } f(\vec{x}, y) \downarrow \text{ and } \neq 0$$

otherwise

least fixed point is  $m : \mathbb{N}^{k+1} \rightarrow \mathbb{N}$

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

computable  
by I Recursion Theorem

hence

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