COMPUTABILITY (16/12/2025)

2nd RECURSION THEOREM

let $f : \mathbb{N} \to \mathbb{N}$ computable total extensional

$P \to \boxed{f} \to P'$

$\forall e, e' \in \mathbb{N}$

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

by Mihill - Shepherdson's theorem there is a (unique)

recursive functional s.t. $\Phi : \mathcal{F}(\mathbb{N}) \to \mathcal{F}(\mathbb{N})$

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

By 1st recursion theorem Φ has a least fixpoint $f_\Phi : \mathbb{N} \to \mathbb{N}$ computable

$$\begin{cases} \Phi(f_\Phi) = f_\Phi \\ \exists e \in \mathbb{N} \text{ s.t. } f_\Phi = \varphi_e \end{cases}$$

$$\varphi_e = f_\Phi = \Phi(f_\Phi) = \Phi(\varphi_e) = \varphi_{f(e)}$$

In summary

Given $f : \mathbb{N} \to \mathbb{N}$ total computable ~~extensional~~

there is $e \in \mathbb{N}$ s.t. $\varphi_e = \varphi_{f(e)}$

program transformer

$P \to \boxed{f} \to P'$

for every transformation, possibly brutal, there is a program P which before and after the transformation, computes the same function.

2^{nd} RECURSION THEOREM

Let $f: \mathbb{N} \to \mathbb{N}$ be a total computable function.

Then there exists $e \in \mathbb{N}$ s.t. $\varphi_e = \varphi_{f(e)}$

proof

let $f: \mathbb{N} \to \mathbb{N}$ be total and computable

consider

$$x \longmapsto f(\varphi_x(x)) \qquad \text{computable}$$

$$ \overset{\shortparallel}{f(\psi_U(x,x))}$$

define $g: \mathbb{N}^2 \to \mathbb{N}$

$$g(x,y) = \varphi_{f(\varphi_x(x))}(y) \qquad \text{convention } \varphi_\uparrow = \uparrow$$

$$= \psi_U(f(\varphi_x(x)), y)$$

$$= \psi_U(f(\psi_U(x,x)), y) \qquad \text{computable}$$

By the smn theorem there is $s: \mathbb{N} \to \mathbb{N}$ total computable s.t.

$$\varphi_{s(x)}(y) = g(x,y) = \varphi_{f(\varphi_x(x))}(y) \qquad \forall x, y$$

Since s is computable, there is $m \in \mathbb{N}$ s.t. $s = \varphi_m$

$$\varphi_{\varphi_m(x)}(y) = \varphi_{f(\varphi_x(x))}(y) \qquad \forall x, y$$

In particular, for $x = m$

$$\varphi_{\varphi_m(m)}(y) = \varphi_{f(\varphi_m(m))}(y) \qquad \forall y$$

Hence

$$\varphi_{\varphi_m(m)} = \varphi_{f(\varphi_m(m))}$$

If we let $e = \varphi_m(m)$

$$\varphi_e = \varphi_{f(e)} \qquad \text{as desired}$$

(note that $\varphi_m(m)\downarrow$ since $\varphi_m = s$ total) $\square$

Idea :

for each $h : \mathbb{N} \to \mathbb{N}$ computable one can "transform" the enumeration

$$\begin{array}{llllll} & \varphi_0 & \varphi_1 & \varphi_2 & \varphi_3 & \cdots\cdots \\ h & \rightsquigarrow & & & & \\ & \varphi_{h(0)} & \varphi_{h(1)} & \varphi_{h(2)} & \varphi_{h(3)} & \cdots\cdots \end{array}$$

we do the above for every $\varphi_i \quad i = 0, 1, 2, \ldots$

$$\begin{array}{llllll} E_0 & \varphi_{\varphi_0(0)} & \varphi_{\varphi_0(1)} & \varphi_{\varphi_0(2)} & \varphi_{\varphi_0(3)} & \cdots \\ E_1 & \varphi_{\varphi_1(0)} & \varphi_{\varphi_1(1)} & \varphi_{\varphi_1(2)} & \varphi_{\varphi_1(3)} & \cdots \\ E_2 & \varphi_{\varphi_2(0)} & \varphi_{\varphi_2(1)} & \varphi_{\varphi_2(2)} & \varphi_{\varphi_2(3)} & \cdots\cdots \\ & & & & \ddots & \end{array}$$

$h(x) = \varphi_x(x)$

in particular one can consider the enumeration produced by

$$h(x) = f(\varphi_x(x)) = \varphi_m(x) \qquad \text{for some } m \in \mathbb{N}$$

$$\begin{array}{llllll} E_0 & \varphi_{\varphi_0(0)} & \varphi_{\varphi_0(1)} & \varphi_{\varphi_0(2)} & \varphi_{\varphi_0(3)} & \cdots \\ E_1 & \varphi_{\varphi_1(0)} & \varphi_{\varphi_1(1)} & \varphi_{\varphi_1(2)} & \varphi_{\varphi_1(3)} & \cdots \\ E_2 & \varphi_{\varphi_2(0)} & \varphi_{\varphi_2(1)} & \varphi_{\varphi_2(2)} & \varphi_{\varphi_2(3)} & \cdots\cdots \\ E_m & \varphi_{f(\varphi_0(0))} & \varphi_{f(\varphi_1(1))} & \varphi_{f(\varphi_2(2))} & \cdots & \bullet \end{array}$$

$\varphi_{\varphi_m(m)}$

$\varphi_{f(\varphi_m(m))}$

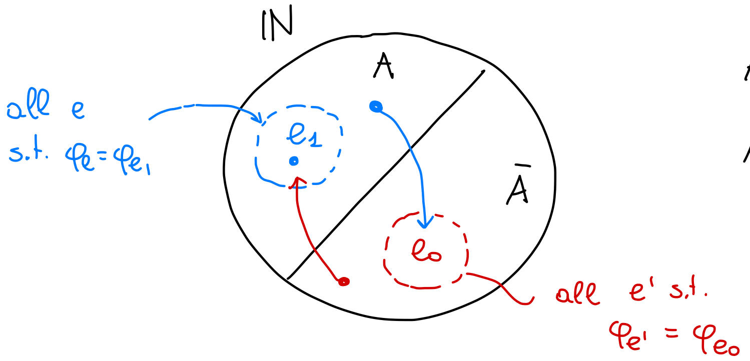
they must coincide

Rice's Theorem

Let $A \subseteq \mathbb{N}$ saturated $A \neq \emptyset$, $A \neq \mathbb{N}$ then A is not recursive

proof (alternative using 2nd recursion theorem)

Let $A \subseteq \mathbb{N}$ saturated, $A \neq \emptyset$, $A \neq \mathbb{N}$



$A \neq \emptyset \leadsto \exists e_1 \in A$

$A \neq \mathbb{N} \leadsto \exists e_0 \notin A$

Assume by contradiction that A is recursive and define

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

$$f(x) = \begin{cases} e_0 & \text{if } x \in A \\ e_1 & \text{if } x \notin A \end{cases}$$

$$= e_0 \cdot \chi_A(x) + e_1 \cdot \chi_{\bar{A}}(x)$$

$x \in A \qquad e_0 \cdot 1 + e_1 \cdot 0 = e_0$

$x \notin A \qquad e_0 \cdot 0 + e_1 \cdot 1 = e_1$

since A is assumed recursive, f is computable, it is total but for all $e \in \mathbb{N}$ $\varphi_e \neq \varphi_{f(e)}$. In fact

- if $e \in A \Rightarrow f(e) = e_0 \notin A$ and since A saturated

$$\varphi_e \neq \varphi_{e_0} = \varphi_{f(e)}$$

- if $e \notin A \Rightarrow f(e) = e_1 \in A$ and since A saturated

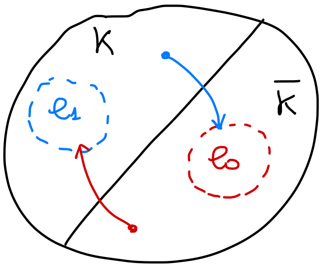
$$\varphi_e \neq \varphi_{e_1} = \varphi_{f(e)}$$

This contradicts the 2nd recursion theorem. Absurd.

$\leadsto$ A is not recursive $\square$

Proposition: The halting set $K = \{x \in \mathbb{N} \mid \varphi_x(x)\downarrow\}$ is not recursive.

proof (alternative, using 2nd recursion theorem)



- if $e_0 \in \mathbb{N}$ is s.t. $\varphi_{e_0} = \emptyset$ $(\varphi_{e_0}(x)\uparrow \forall x)$
  then $e_0 \in \overline{K}$
- if $e_1 \in \mathbb{N}$ is s.t. $\varphi_{e_1} = \mathbb{1}$ $(\varphi_{e_1}(x) = 1 \ \forall x)$
  then $e_1 \in K$

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

$$f(x) = \begin{cases} e_0 & \text{if } x \in K \\ e_1 & \text{if } x \notin K \end{cases}$$

$$= e_0 \cdot \chi_K(x) + e_1 \cdot \chi_{\overline{K}}(x)$$

if K, by contradiction, were recursive the f would be computable total and such that $\forall e \in \mathbb{N}$ $\varphi_e \neq \varphi_{f(e)}$

$\rightarrow$ $e \in K$ $\leadsto$ $f(e) = e_0$ hence $\varphi_e(e)\downarrow$ and $\varphi_{f(e)}(e) = \varphi_{e_0}(e)\uparrow$

$\leadsto$ $\varphi_e \neq \varphi_{f(e)}$

$\rightarrow$ $e \notin K$ $\leadsto$ $f(e) = e_1$ hence $\varphi_e(e)\uparrow$ and $\varphi_{f(e)}(e) = \varphi_{e_1}(e) = 1$

$\leadsto$ $\varphi_e \neq \varphi_{f(e)}$

contradicts 2nd recursion theorem.

Hence K is not recursive $\square$

* K is saturated? NO

$$K = \{ x \in \mathbb{N} \mid \varphi_x(x)\downarrow \}$$

We want to show that there are $e, e' \in \mathbb{N}$ s.t.

$$\boxed{\begin{array}{l} \varphi_e = \varphi_{e'} \\ e \in K \quad , \quad e' \notin K \end{array}}$$

Assume that there is $e \in \mathbb{N}$ s.t.

$$\varphi_e(y) = \begin{cases} 0 & \text{if } y = e \\ \uparrow & \text{otherwise} \end{cases} \qquad (*)$$

then
- $e \in K$ since $\varphi_e(e) = 0 \downarrow$
- there is $e' \neq e$ s.t. $\varphi_{e'} = \varphi_e$
- $e' \notin K$ since $\varphi_{e'}(e') = \varphi_e(e') \uparrow$ ↖ since $e \neq e'$

We need to show that there exists $e \in \mathbb{N}$

$$\varphi_e(y) = \begin{cases} 0 & \text{if } y = e \\ \uparrow & \text{otherwise} \end{cases} \qquad (*)$$

kleene.py

```
def P(y):
    if y = "------"        read("kleene.py")
        then return 0
        else loop
```

formally

$$g(x,y) = \begin{cases} 0 & \text{if } y = x \\ \uparrow & \text{otherwise} \end{cases}$$

$= \mu w. \, |y - x|$

computable

by smn theorem, there is $s: \mathbb{N} \to \mathbb{N}$ total computable s.t.

$\forall x, y$

$$\varphi_{s(x)}(y) = g(x,y) = \begin{cases} 0 & \text{if } y = x \\ \uparrow & \text{otherwise} \end{cases}$$

Since s total computable, by the 2nd recursion theorem, there is $e \in \mathbb{N}$ s.t. $\varphi_e = \varphi_{s(e)}$. Thus

$$\varphi_e(y) = \varphi_{s(e)}(y) = g(e, y) = \begin{cases} 0 & \text{if } y = e \\ \uparrow & \text{otherwise} \end{cases}$$

as desired.

Hence $(*)$ is true, hence K is not saturated. $\square$

EXERCISE: RANDOM NUMBERS (from 1st lesson)

→ $m \in \mathbb{N}$ is random if all programs producing m in output are "larger" than m

two questions:

(1) there are infinitely many random numbers

(2) the property of being random is not decidable

Try again:

→ size of a program? $\quad |P_e| = e$

→ define a number to be random if

for all $e \in \mathbb{N}$ s.t. $\varphi_e(0) = m$ it holds $e > m$

EXERCISE :

Let $f: \mathbb{N} \to \mathbb{N}$ be a function

and consider

$$B_f = \{ e \in \mathbb{N} \mid \varphi_e = f \}$$

Are B_f, $\overline{B_f}$ recursive / r.e. ?

(1) f not computable

$B_f = \emptyset$ $\qquad$ $\overline{B_f} = \mathbb{N}$ $\qquad$ recursive and thus r.e.

(2) f computable

B_f is saturated

f computable means there is $e \in \mathbb{N}$ $\varphi_e = f$ and $e \in B_f \neq \emptyset$

$g \neq f$ is computable and $e' \in \mathbb{N}$ is st. $\varphi_{e'} = g$ then $e' \notin B_f \neq \mathbb{N}$

$\Rightarrow$ by Rice's theorem $\qquad$ B_f not recursive, hence

$\qquad\qquad\qquad\qquad\qquad\qquad\quad$ $\overline{B_f}$ " "

can B_f, $\overline{B_f}$ be r.e. ?

if $f = \emptyset$ $\qquad$ $(f(x)\uparrow \ \forall x)$

then $\overline{B_f} = \{ e \mid \varphi_e \neq \emptyset \}$

$$= \{ e \mid \underbrace{\exists y.\ \underbrace{\varphi_e(y)\downarrow}_{\text{semidec}}}_{\text{semidec.}} \}$$

$$sc_{\overline{B_f}}(x) = \mathbb{1}\left(\mu w.\ H(x, (w)_1, (w)_2)\right)$$

complete the exercise!