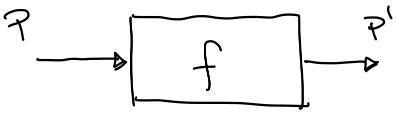
Computability (14/12/2021)

2nd Recursion Theorem

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

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

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



by Myhill - Shepherdson theorem $\exists$ (unique) recursive functional Φ such that $\forall e \in \mathbb{N}$

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

$\Rightarrow$ by 1st recursion theorem Φ has a least fixed point

$f_\Phi : \mathbb{N} \to \mathbb{N}$ computable

$$\Phi(f_\Phi) = f_\Phi \quad (*)$$

$$\exists e_0 \text{ s.t. } \varphi_{e_0} = f_\Phi \quad (*)$$

$$\varphi_{e_0} = f_\Phi = \Phi(f_\Phi) = \Phi(\varphi_{e_0}) = \varphi_{f(e_0)}$$

In summary:

$$\boxed{\begin{array}{l}\text{If } f : \mathbb{N} \to \mathbb{N} \text{ is computable total } \cancel{\text{extensional}} \\ \text{then there exists } e_0 \text{ s.t. } \varphi_{e_0} = \varphi_{f(e_0)}\end{array}}$$

2nd recursion theorem

II Recursion Theorem (Kleene)

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

Then there exists $e_0 \in \mathbb{N}$ such that $\varphi_{e_0} = \varphi_{f(e_0)}$

$$\text{programs} \longrightarrow \boxed{\;f\;} \longrightarrow \text{programs}$$

proof

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

observe $\quad x \mapsto \varphi_x(x) \quad$ computable
$$\varphi_x(x) = \Psi_U(x,x)$$

$$x \mapsto f(\varphi_x(x)) \quad \text{computable}$$

take

conventionally: $\varphi_\uparrow = \uparrow$

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

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

$$= \Psi_U(f(\Psi_U(x,x)), y)$$

computable

by smn theorem there exists $s: \mathbb{N} \to \mathbb{N}$ total computable

such that

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

Since s is computable there exists $m \in \mathbb{N}$ such that

$$\varphi_m = \underline{\underline{s}}$$

hence

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

For $x = m$

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

Set $e_0 = \varphi_m(m)\downarrow$ and replace in $(*)$

$$\varphi_{e_0}(y) = \varphi_{f(e_0)}(y) \qquad \forall y$$

i.e.

$$\varphi_{e_0} = \varphi_{f(e_0)} \qquad \blacksquare$$

{idea}

if $\boxed{h \text{ is computable}}$ $\qquad h : \mathbb{N} \to \mathbb{N}$

$$\varphi_0 \quad \varphi_1 \quad \varphi_2 \quad \varphi_3 \quad \cdots$$

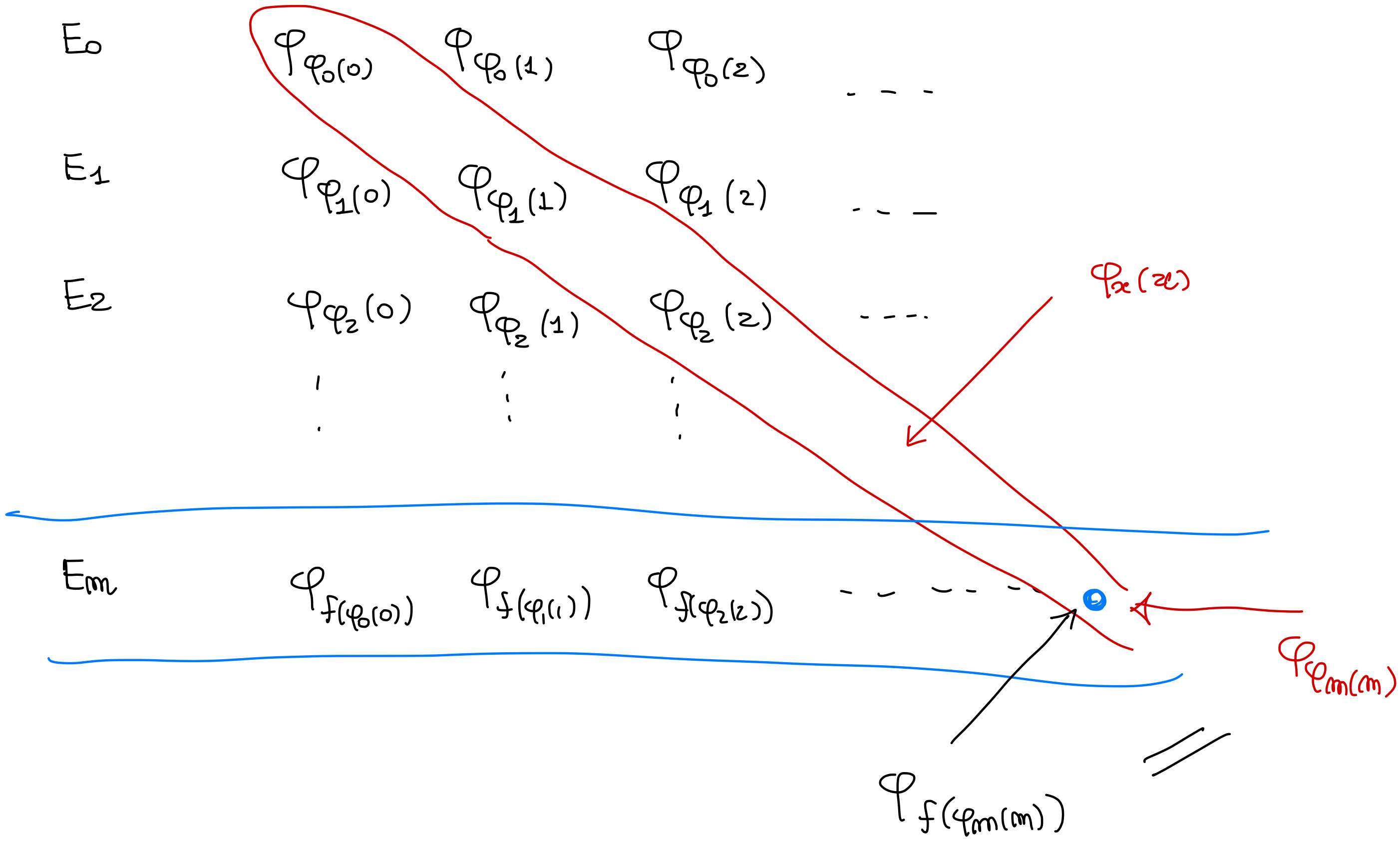
$$\Big\downarrow h$$

$$\boxed{\varphi_{h(0)} \quad \varphi_{h(1)} \quad \varphi_{h(2)} \quad \cdots}$$

E_0	$\varphi_{\varphi_0(0)}$	$\varphi_{\varphi_0(1)}$	$\varphi_{\varphi_0(2)}$	$\cdots$	induced φ_0
E_1	$\varphi_{\varphi_1(0)}$	$\varphi_{\varphi_1(1)}$	$\varphi_{\varphi_1(2)}$	$\cdots$	
E_2	$\varphi_{\varphi_2(0)}$	$\varphi_{\varphi_2(1)}$	$\varphi_{\varphi_2(2)}$	$\cdots$	
	$\vdots$	$\vdots$	$\vdots$		

in the proof we took

$$h(x) = f(\varphi_x(x)) = f(\psi_U(x,x)) = \varphi_m(x)$$

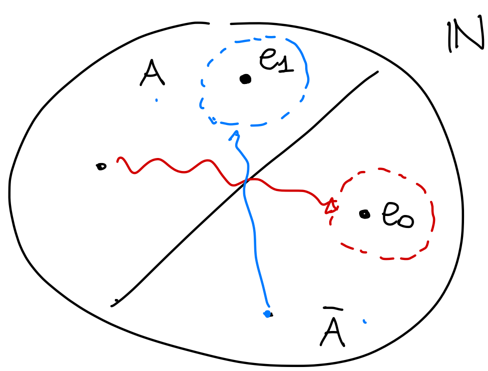


Rice's Theorem

Let $A \subseteq \mathbb{N}$ $\quad A \neq \emptyset$, $A \neq \mathbb{N}$ and A saturated $\Rightarrow$ A not recursive

<u>proof</u> (alternative, using II rec. theorem)

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



$e_1 \in A$, $\quad e_0 \notin A$

Assume by contradiction that $\underset{\sim\sim\sim\sim\sim}{A \text{ is recursive}}$

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

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

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

$$\begin{cases} x \in A & e_0 \cdot 1 + e_1 \cdot 0 = e_0 \\ x \notin A & e_0 \cdot 0 + e_1 \cdot 1 = e_1 \end{cases}$$

since A is recursive $\qquad \chi_A$ computable

$\bar{A}$ " "" $\qquad \chi_{\bar{A}}$ "

$\Rightarrow$ f computable

and $\qquad f$ is total

$\Rightarrow$ by II Recursion theorem $\exists e \in \mathbb{N}$ such that $\varphi_e = \varphi_{f(e)}$

two possibilities

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

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

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

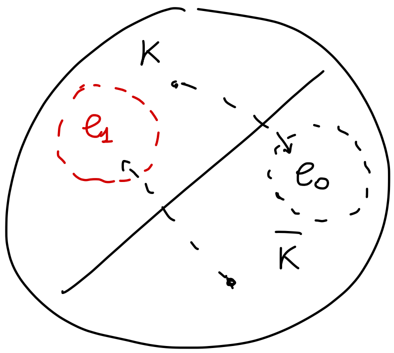
$$\varphi_e \neq \varphi_{e_1} = \varphi_{f(e)}$$

absurd.

$\Rightarrow$ A can't be recursive $\blacksquare$

OBSERVATION: The halting set $K = \{ x \mid \varphi_x(x)\downarrow \}$ is not recursive

proof (alternative, using 2nd recursion theorem)



if $e_0 \in \mathbb{N}$ s.t. $\varphi_{e_0}(x)\uparrow \ \forall x \Rightarrow e_0 \in \overline{K}$

if $e_1 \in \mathbb{N}$ s.t. $\varphi_{e_1}(x) = x \ \forall x \Rightarrow e_1 \in K$

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

$$f(x) = \begin{cases} e_0 & \text{if } x \in K \\ e_1 & \text{if } x \notin K \end{cases} = e_0 \, \chi_K(x) + e_1 \cdot \chi_{\overline{K}}(x)$$

if K were recursive then $\chi_K, \chi_{\overline{K}}$ computable $\Rightarrow$ f total computable

then, by II recursion theorem, there is $e \in \mathbb{N}$ s.t. $\varphi_e = \varphi_{f(e)}$

$\rightarrow e \in K \Rightarrow f(e) = e_0 \Rightarrow \varphi_e(e)\downarrow \neq \varphi_{f(e)}(e) = \varphi_{e_0}(e)\uparrow$

$\rightarrow e \in \overline{K} \Rightarrow f(e) = e_1 \quad \varphi_e(e)\uparrow \neq \varphi_{f(e)}(e) = \varphi_{e_1}(e) = e\downarrow$

absurdum.

$\Rightarrow$ K can't be recursive $\square$

* $K = \{x \mid \varphi_x(x)\downarrow\}$ is not saturated

$\boxed{\begin{array}{l} \text{We want to show that there exist} \\ \quad e, e' \in \mathbb{N} \\ \quad \varphi_e = \varphi_{e'} \qquad\qquad e \in K \\ \qquad\qquad\qquad\qquad\quad e' \notin K \end{array}}$

* Assume that there is $e \in \mathbb{N}$ such that

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

then

- $e \in K$ $\qquad \varphi_e(e) = 0\downarrow$
- there exists $e' \neq e$ s.t. $\varphi_{e'} = \varphi_e$

and $e' \notin K$

(because $\varphi_{e'}(e') = \underset{e \neq e'}{\varphi_e(e')}\uparrow$)

* We need to prove that there is $e \in \mathbb{N}$ s.t. $\varphi_e(x) = \begin{cases} 0 & \text{if } x = e \\ \uparrow & \text{otherwise} \end{cases}$

INFORMALLY

weird program.py

```
def P(x)
    if x == readline('weird program.py')
        return 0
    else
        loop
```

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

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

$$= \mu z.\ |x - e|$$

computable. So by smn theorem $\exists\ \underline{s: \mathbb{N} \to \mathbb{N} \text{ total computable}}$ such that

$$\varphi_{s(e)}(x) = g(e, x) \qquad \forall e, x$$

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

Since s is total and computable there exists $e_0 \in \mathbb{N}$ st.

$$\varphi_{s(e_0)} = \varphi_{e_0}$$

then

$$\varphi_{e_0}(x) = \varphi_{s(e_0)}(x) = \begin{cases} 0 & \text{if } x = e_0 \\ \uparrow & \text{otherwise} \end{cases}$$

$\square$

$\boxed{\text{EXERCISE}}$:

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

and consider $\qquad B_f = \{ x \mid \varphi_x = f \}$

Are $B_f, \overline{B_f}$ recursive / recursively enumerable ?

1. f is not computable

   $B_f = \emptyset \qquad \overline{B_f} = \mathbb{N} \qquad$ recursive

2. f is computable

   B_f is saturated, $B_f \neq \emptyset$, $B_f \neq \mathbb{N} \overset{\text{Rice}}{\Rightarrow} B_f, \overline{B_f}$ are not recursive

maybe $B_f, \overline{B_f}$ are not r.e. ?

wrong in general

$$\text{if} \quad f = \emptyset \qquad f(x)\uparrow \quad \forall x$$

$$\overline{B_\emptyset} = \{ x \mid \varphi_x \neq \emptyset \} = \{ x \mid \exists y.\ \varphi_x(y)\downarrow \}$$

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

Complete the exercise.