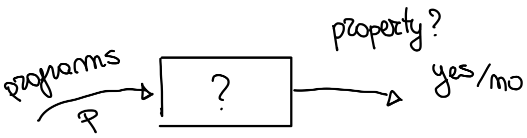
COMPUTABILITY (27/11/2023)

Rice's Theorem



> every property of programs which concerns the I/O behaviour is <u>undecidable</u>

"P is terminating on every input"
"P has some fixed $n \in \mathbb{N}$ as an output"
"P computes a function f"
⋮

} undecidable

"the length of program P is ≤ 10"
⋮

} decidable

> What is a behavioural property of a program?

$$A \subseteq \mathbb{N}$$

↑ set of programs (program property)

$$T = \{ n \mid P_n \text{ is terminating on every input} \}$$
$$= \{ n \mid \varphi_n \text{ is total} \}$$

$$ONE = \{ n \mid P_n \text{ is a sound implementation of } \mathbb{1} \}$$
$$= \{ n \mid \varphi_n \text{ is } \mathbb{1} \}$$

$A \subseteq \mathbb{N}$ (program property) is a behavioural property if for all programs $n \in \mathbb{N}$ the fact that $n \in A$ or $n \notin A$ <u>only depends</u> on φ_n

Def. (saturated / extensional set): $A \subseteq \mathbb{N}$ is <u>saturated</u> (<u>extensional</u>)

if for all $m, n \in \mathbb{N}$

if $m \in A$ and $\varphi_m = \varphi_n$ then $n \in A$

$\Updownarrow$

A saturated if $A = \{ n \mid \varphi_n \text{ satisfies a property of functions} \}$

$= \{ n \mid \varphi_n \in \mathcal{A} \}$

where $\mathcal{A} \subseteq \mathcal{F}$ ← set of all functions

($\mathcal{A}$: property of functions)

<u>Examples</u>

* $T = \{ n \mid P_n \text{ is terminating on every input} \}$

$= \{ n \mid \varphi_n \text{ is total} \}$

$= \{ n \mid \varphi_n \in \mathcal{T} \} \qquad \mathcal{T} = \{ f \in \mathcal{F} \mid f \text{ total} \}$

* $ONE = \{ n \mid P_n \text{ is a sound implementation of } \mathbb{1} \}$

$= \{ n \mid \varphi_n = \mathbb{1} \} = \{ n \mid \varphi_n \in \{ \mathbb{1} \} \}$

* $LEN10 = \{ n \mid P_n \text{ has length} \leq 10 \}$

$m \in LEN10$

and $\varphi_m = \varphi_n$

$n \notin LEN10$

e.g. $m = \gamma(Z(1)) \in LEN10$

$\varphi_m = \varphi_n = 0$ ← constant zero

$$n = \gamma \left(\begin{matrix} Z(1) \\ Z(1) \\ \vdots \\ Z(1) \end{matrix} \right) \Big\} \geq 11 \qquad \notin LEN10$$

$*\quad K = \{ n \mid \varphi_n(n)\downarrow \}$

$\quad\;\; = \{ n \mid \varphi_n \in \mathcal{K} \}$

$\mathcal{K} = \{ f \mid f(\,?\,)\downarrow \}$???

It seems that K is not saturated

formally I should find $m, n \in \mathbb{N}$

$m \in K \qquad \varphi_m(m)\downarrow$

$n \notin K \qquad \varphi_n(n)\uparrow$

and $\varphi_m = \varphi_n$

---

if we were able to show that there is program $m \in \mathbb{N}$ s.t.

$$\varphi_m(x) = \begin{cases} 1 & \text{if } x = m \\ \uparrow & \text{otherwise} \end{cases}$$

(*)

---

we can conclude

① $\boxed{m \in K} \qquad \varphi_m(m)\downarrow$

② for a computable function there are infinitely many programs

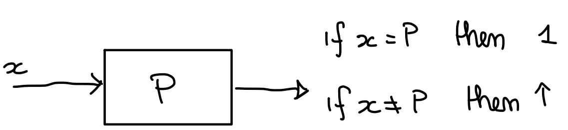
hence there is $n \neq m$ s.t. $\boxed{\varphi_m = \varphi_n}$

③ $\boxed{n \notin K}$

$$\varphi_n(n) \underset{\underset{\varphi_n = \varphi_m}{\uparrow}}{=} \varphi_m(n)\uparrow_{\underset{n \neq m}{\nwarrow}}$$

K is not saturated!

What about (*)?




```
def P(x) :
  if x = "def P(x) :
              ...
                       "
```

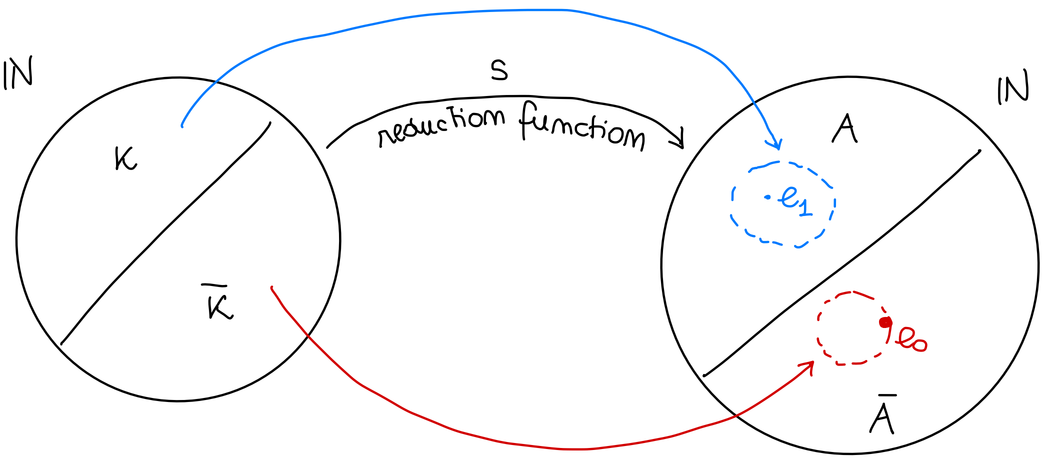
Rice's Theorem :

Let $A \subseteq \mathbb{N}$ if A is <u>saturated</u> $A \neq \emptyset$, $A \neq \mathbb{N}$

then A is <u>not recursive</u>

proof

we show $K \leq_m A$ (since K is not recursive so A not recursive)



Let $e_0 \in \mathbb{N}$ be s.t. $\varphi_{e_0}(x)\uparrow \ \forall x$ (program for the function always undefined)

① Assume $e_0 \notin A$

let $e_1 \in A$ (it exists since $A \neq \emptyset$)

define

$$g(x,y) = \begin{cases} \varphi_{e_1}(y) & \text{if } x \in K \\ \varphi_{e_0}(y) & \text{if } x \in \bar{K} \end{cases}$$

$$= \begin{cases} \varphi_{e_1}(y) & \text{if } x \in K & [\ \varphi_x(x)\downarrow\ ] \\ \uparrow & \text{if } x \in \bar{K} & [\ \varphi_x(x)\uparrow\ ] \end{cases}$$

$$= \varphi_{e_1}(y) \cdot \mathbb{1}(\varphi_x(x))$$

$\mathbb{1}$: 1 if $\varphi_x(x)\downarrow$, $\uparrow$ otherwise

$$= \varphi_{e_1}(y) \cdot \mathbb{1}(\psi_U(x,x))$$

computable !

By smn theorem there is $s : \mathbb{N} \to \mathbb{N}$ total and computable s.t. $\forall x, y$

$$\varphi_{s(x)}(y) = g(x,y) = \begin{cases} \varphi_{e_1}(y) & \text{if } x \in K \\ \varphi_{e_0}(y) & \text{if } x \in \overline{K} \end{cases}$$

s is the reduction function for $K \leq_m A$

* $x \in K \rightsquigarrow s(x) \in A$

if $x \in K$ then $\varphi_{s(x)}(y) = g(x,y) = \varphi_{e_1}(y) \quad \forall y$

i.e. $\varphi_{s(x)} = \varphi_{e_1}$. Since $e_1 \in A$ and A saturated $\rightsquigarrow s(x) \in A$

* $x \notin K \rightsquigarrow s(x) \notin A$

if $x \notin K$ then $\varphi_{s(x)}(y) = g(x,y) = \varphi_{e_0}(y) \quad \forall y$

i.e. $\varphi_{s(x)} = \varphi_{e_0}$. Since $e_0 \notin A$ and A saturated $\rightsquigarrow s(x) \notin A$

Hence s is the reduction function for $K \leq_m A$ and since K not recursive, we deduce A not recursive.

2 if instead $e_0 \in A$

$e_0 \notin \bar{A}$

$\bar{A}$ saturated (since A is saturated)

$\bar{A} \neq \emptyset$ (since $A \neq \mathbb{N}$)

$\bar{A} \neq \mathbb{N}$ (" $A \neq \emptyset$)

$\rightsquigarrow$ by (1) applied to $\bar{A}$ we deduce $\bar{A}$ not recursive

$\rightsquigarrow$ A not recursive (since A recursive $\rightsquigarrow$ $\bar{A}$ recursive)

$\square$

* Output problem $\quad B_n = \{ x \mid n \in E_x \}$

we observed $\quad K \leq_m B_n$

- B_n saturated, in fact

$B_n = \{ x \mid \varphi_x \in \mathcal{B}_n \}$

$\mathcal{B}_n = \{ f \mid n \in \mathrm{cod}(f) \}$

- $B_n \neq \emptyset$

e.g. let $e_1 \in \mathbb{N}$ be s.t. $\varphi_{e_1}(y) = y \quad \forall y \quad \rightsquigarrow \quad n \in E_{e_1} = \mathbb{N}$

$\rightarrow \quad e_1 \in B_n \neq \emptyset$

- $B_n \neq \mathbb{N}$

e.g. let $e_2 \in \mathbb{N}$ s.t. $\varphi_{e_2}(y) = m \; (\neq n) \quad \forall y$

$e_2 \in B_n \quad$ (since $\quad n \notin E_{e_2} = \{m\}$)

$\Rightarrow$ By Rice's theorem B_n is not recursive.

EXAMPLE :

$I = \{ x \in \mathbb{N} \mid P_x$ has infinitely many possible outputs $\}$

$= \{ x \in \mathbb{N} \mid E_x$ is infinite $\}$

* saturated

$I = \{ x \mid \varphi_x \in \mathcal{Y} \}$

with $\quad \mathcal{Y} = \{ f \mid \mathrm{cod}(f)$ infinite $\}$

* $I \neq \emptyset$

if e_1 is as in previous exercise $\Rightarrow E_{e_1} = \mathbb{N}$ infinite $\Rightarrow e_1 \in I$

* $I \neq \mathbb{N}$

if e_2 is as before $\rightsquigarrow E_{e_2} = \{m\} \rightsquigarrow e_2 \notin I$

$\Rightarrow$ I not recursive, by Rice's theorem.

Example

$$A = \{ x \mid x \in W_x \cap E_x \}$$

saturated ?

$$A = \{ x \mid \varphi_x \in \mathcal{A} \}$$

$$\mathcal{A} = \{ f \mid ? \in dom(f) \cap cod(f) \}$$

we do not know what to put here

probably not saturated

we do not use Rice

We $K \leq_m A$, i.e. that there is a total computable function $s: \mathbb{N} \to \mathbb{N}$

s.t.

$$x \in K \quad \text{iff} \quad s(x) \in A$$

$$\Updownarrow$$

$s(x) \in W_{s(x)}$ .... $\varphi_{s(x)}(s(x)) \downarrow$

and

$s(x) \in E_{s(x)}$ ... $\varphi_{s(x)}(y) = s(x)$ for some y

we define

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

$$= y \cdot 1\!\!1(\varphi_x(x))$$

$$= y \cdot 1\!\!1(\varphi_U(x,x)) \quad \text{computable}$$

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

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

s is the reduction function

$\to$ if $x \in K$ then $\varphi_{s(x)}(y) = g(x,y) = y \quad \forall y$

Hence

$$s(x) \in \underset{\mathbb{N}}{\underset{\shortparallel}{W_{s(x)}}} \cap \underset{\mathbb{N}}{\underset{\shortparallel}{E_{s(x)}}} = \mathbb{N} \quad . \text{ Thus } s(x) \in A$$

→ if $x \notin K$ then $\varphi_{s(x)}(y) = g(x,y) \uparrow \quad \forall y$

Hence $s(x) \notin \underbrace{W_{s(x)}}_{= \emptyset} \cap \underbrace{E_{s(x)}}_{= \emptyset} = \emptyset$

Thus $s(x) \notin A$

Thus $K \leq_m A$, and, since K not recursive, also A is not recursive. •