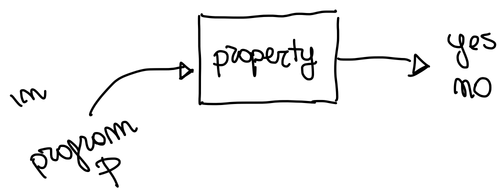
Computability (29/11/2021)

Rice's theorem



every program property
which concerns the behaviour (I/O)
of programs
is not decidable

" P is stopping on every input "

" program P on input 1 provides 2 as output "

" program P computes the function f "

⋮

undecidable

" the length of P is ≤ 10 "

decidable

What is a behavioural property of a program P ?

$A \subseteq \mathbb{N}$

set of programs

$A_1 = \{ n \mid P_n \text{ always stops} \}$

$A_2 = \{ n \mid \text{the function computed by } P_n \text{ is } \mathbb{1} \}$

$A \subseteq \mathbb{N}$ (program property) is behavioural property if for every program $n \in \mathbb{N}$ the fact that $n \in A$ or not only depends on φ_n

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

if $\forall n, m \in \mathbb{N}$ if $n \in A$ and $\varphi_n = \varphi_m$ then $m \in A$.

$\Updownarrow$

A saturated if $A = \{ n \mid \varphi_n \text{ satisfies} \ldots\ldots \}$

$\Updownarrow$ A saturated if $A = \{ n \mid \varphi_n \in \mathcal{A} \}$

where $\mathcal{A} \subseteq \mathcal{C}$

Examples :

* $T = \{ n \mid P_n$ always terminate (on every input) $\}$ SATURATED

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

$= \{ n \mid \varphi_n \in \mathcal{T} \}$ where $\mathcal{T} = \{ f \mid f \text{ is total} \}$

* ONE $= \{ n \mid P_n \text{ computes } \mathbb{1} \}$

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

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

* LEN10 $= \{ n \mid P_n$ length of $P_n \leq 10 \}$ NOT SATURATED

$n \in$ LEN10

and $\varphi_n = \varphi_m$

$m \notin$ LEN 10

$n = \gamma(Z(1))$ $\varphi_n = \lambda x.0$ $n \in$ LEN10

$$m = \gamma \left(\begin{array}{c} Z(1) \\ \vdots \\ Z(1) \end{array} \right) > 10 \qquad \varphi_m = \lambda x.0 \qquad m \notin \text{LEN } 10$$

* $K = \{ x \mid \varphi_x (x) \downarrow \}$

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

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

apparently K is not saturated

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

$m \in K \quad \underline{\varphi_m} (m) \downarrow$

and $\underline{\varphi_m = \varphi_n}$

$n \notin K \quad \underline{\varphi_n} (n) \uparrow$

> if we could find a program m such that
>
> $$\varphi_{\underline{m}} (x) = \begin{cases} 1 & x = m \\ \uparrow & \text{otherwise} \end{cases}$$

we can conclude

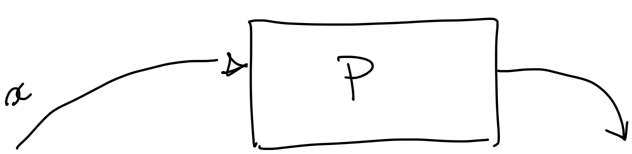
1. $\boxed{m \in K}$ $[\ \varphi_m (m) = 1 \downarrow\ ]$

2. there are infinitely many programs computing φ_m

   $\Rightarrow \exists\ \boxed{n \neq m}$ s.t. $\boxed{\varphi_n = \varphi_m}$

3. $\boxed{n \notin K}$ $\varphi_n (n) = \varphi_m (n) \uparrow$

$\Downarrow$ K not saturated



if $x = P$ then out 0

if $x \neq P$ then $\uparrow$

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

???

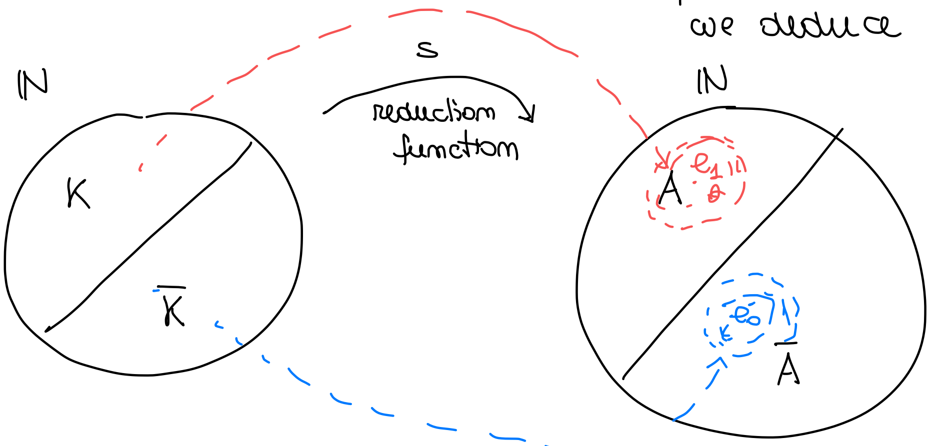
Rice's theorem

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

then A not recursive

proof

we show that $K \leq_m A$ (hence, since K is not recursive we deduce A not recursive)



Let $e_0 \in \mathbb{N}$ be an index for the function always undefined

$$\varphi_{e_0}(x) \uparrow \quad \forall x$$

* $\boxed{\text{We assume that } e_0 \notin A}$

take $e_1 \in A$ (we can because $A \neq \emptyset$)

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

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

$$= \varphi_{e_1}(y) \cdot \underbrace{\mathbb{1}(\varphi_x(x))}_{\substack{1 \quad \text{if } \varphi_x(x)\downarrow \\ \uparrow \quad \text{otherwise}}}$$

$\downarrow$

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

computable

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

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

Now observe that s is a reduction function for $\boxed{K \leq_m A}$

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

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

hence $\varphi_{s(x)} = \varphi_{e_1}$ since $e_1 \in A$ and A saturated hence $s(x) \in A$

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

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

hence $\varphi_{s(x)} = \varphi_{e_0}$ $\quad e_0 \notin A \quad A$ saturated $\Rightarrow s(x) \notin A$

Hence $K \leq_m A$, K not recursive, thus A not recursive

* If instead $e_0 \in A$

$$\left.\begin{array}{l} e_0 \notin \bar{A}, \\ \bar{A} \neq \emptyset \quad (\text{since } A \neq \mathbb{N}) \\ \bar{A} \neq \mathbb{N} \quad (\text{since } A \neq \emptyset) \\ \bar{A} \text{ saturated } (\text{since } A \text{ is so}) \end{array}\right\} \begin{array}{c} \bar{A} \text{ is not recursive} \\ \Downarrow \\ A \text{ is not recursive} \end{array}$$

$\square$

* $B_n = \{x \in \mathbb{N} \mid n \in E_x\}$ is not recursive

we showed that $K \leq_m B_n$

We can conclude the same by observing

1. B_m is saturated

$$B_m = \{ x \mid \varphi_x \in \mathcal{B}_m \} \qquad \mathcal{B}_m \subseteq \mathcal{C}$$

$$\mathcal{B}_m = \{ f \mid m \in \text{cod}(f) \}$$

2. $B_m \neq \emptyset$

e.g. if e_1 is such that $\varphi_{e_1} = \lambda x. x$ then $e_1 \in B_m$ (since $m \in E_{e_1} = \mathbb{N}$)

3. $B_m \neq \mathbb{N}$

e.g. if e_2 such that $\varphi_{e_2} = \lambda x. n$ $\quad n \neq m$ then $e_2 \in B_m$ since $m \notin E_{e_2} = \{n\}$

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

<u>Example</u>

$$I = \{ n \in \mathbb{N} \mid E_n \text{ is infinite} \}$$

saturated ? yes

$$= \{ n \in \mathbb{N} \mid \varphi_n \in \mathcal{Y} \}$$

$$\mathcal{Y} = \{ f \mid \text{cod}(f) \text{ is infinite} \}$$

$I \neq \emptyset$ $\quad$ e_1 (defined above) $\quad$ $e_1 \in I$ $\quad$ (since $E_{e_1} = \mathbb{N}$)

$I \neq \mathbb{N}$ $\quad$ e_2 (" ") $\quad$ $e_2 \notin I$ $\quad$ (since $E_{e_2} = \{n\}$)

Hence I is not recursive (by Rice's theorem)

<u>Exampl</u> : $A = \{ x \mid x \in W_x \cap E_x \}$

* saturated ? probably not

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

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

* we show $K \le_m A$

reduction function $s : \mathbb{N} \to \mathbb{N}$ total computable such that

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

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

$$= y \cdot \underbrace{\mathbb{1}(\varphi_x(x))}_{\substack{\uparrow \text{ if } x \notin K \\ 1 \text{ if } x \in K}} = y \cdot \mathbb{1}(\Psi_U(x,x))$$

computable

By smn theorem $\exists\, s : \mathbb{N} \to \mathbb{N}$ total computable such that

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

s is the reduction function for $K \le_m A$ (so A not recursive)

* $x \in K \Rightarrow \varphi_{s(x)}(y) = g(x,y) = y \ \forall y \Rightarrow W_{s(x)} = \mathbb{N}$, $E_{s(x)} = \mathbb{N}$

$\Rightarrow s(x) \in W_{s(x)} \cap E_{s(x)} = \mathbb{N} \Rightarrow s(x) \in A$

* $x \notin K \Rightarrow \varphi_{s(x)}(y) = g(x,y)\uparrow \ \forall y \Rightarrow W_{s(x)} = E_{s(x)} = \emptyset$

$\Rightarrow s(x) \notin W_{s(x)} \cap E_{s(x)} = \emptyset \Rightarrow s(x) \notin A$