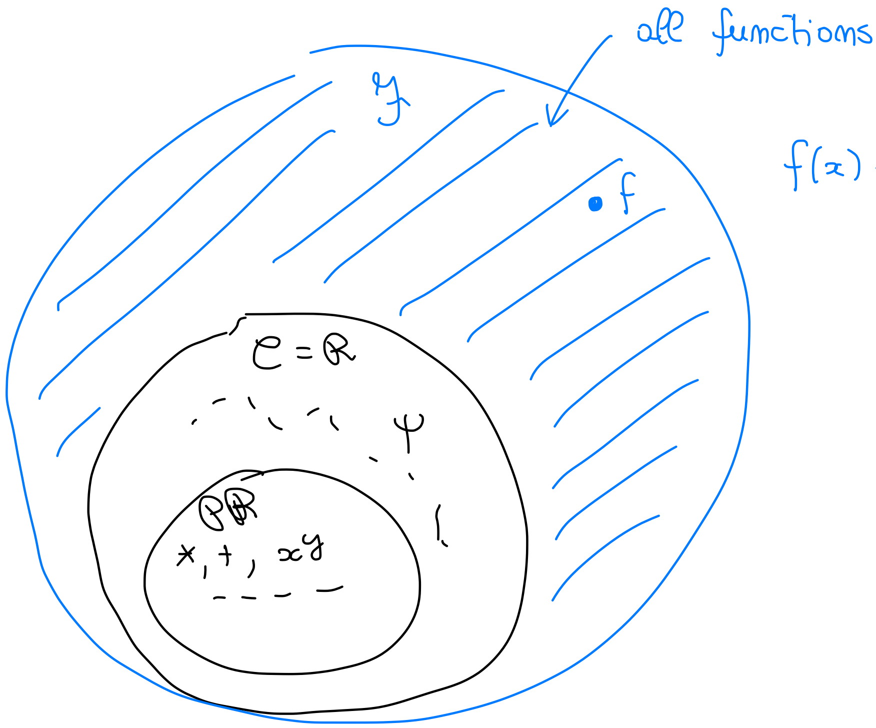
Computability (23/11/2021)

* **Recursive and Recursively enumerable sets**



$$f(x) = \begin{cases} 1 & \text{if } x \in W_x \ (\varphi_x(x)\downarrow) \\ 0 & \text{otherwise} \end{cases}$$

given $X \subseteq \mathbb{N}$ "$\underline{x \in X}$?"

↑ programs

$X = \{x \in \mathbb{N} \mid \varphi_x = fact\}$

$X' = \{x \in \mathbb{N} \mid P_x$ linear complexity$\}$

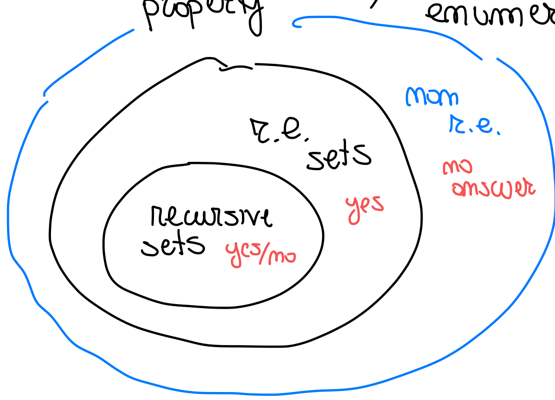
$X'' = \{x \in \mathbb{N} \mid P_x$ does not use $R_J\}$

$X''' = \{x \in \mathbb{N} \mid P_x$ executes at least once each instruct.$\}$

⋮

answer yes/no : decidable property / recursive set

answer yes/↑ : semidecidable property / recursively enumerable set



* Recursive sets

A set $A \subseteq \mathbb{N}$ is recursive if the characteristic function

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

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$ is computable

($\Leftrightarrow$ "$x \in A$" is decidable)

* $\mathbb{N}$ is recursive $\quad \chi_{\mathbb{N}}(x) = 1 \quad \forall x \in \mathbb{N}$ computable

  $\emptyset$ is recursive $\quad \chi_{\emptyset}(x) = 0 \quad \forall x \in \mathbb{N}$ "

  $\mathbb{P}$ " " $\quad \chi_{\mathbb{P}}(x) = \overline{sg}(rm(2, x))$

  P_2 " "

  ⋮

* A finite $\Rightarrow$ A recursive

$$A = \{a_1, \dots, a_n\}$$

$$\chi_A(x) = \overline{sg}\left(\prod_{i=1}^{n} |x - a_i|\right)$$

⋮

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

NOT RECURSIVE

$$f_K(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{otherwise} \end{cases}$$ is not computable

* **OBSERVATION**: if $A, B \subseteq \mathbb{N}$ recursive then

(i) $\bar{A} = \mathbb{N} \setminus A$

(ii) $A \cap B$ are recursive

(iii) $A \cup B$

proof

(i) A is recursive $\Rightarrow$ χ_A computable

$$\chi_{\bar{A}}(x) = \begin{cases} 1 & \text{if } x \in \bar{A} \\ 0 & \text{otherwise} \end{cases} = \begin{cases} 1 & x \notin A \\ 0 & x \in A \end{cases}$$

$$= \overline{sg}\,(\chi_A(x))$$

(ii)
(iii) exercise

* **REDUCTION**

problems $\mathcal{A}$ and $\mathcal{B}$

$\mathcal{A}$ reduces to $\mathcal{B}$ if every instance of $\mathcal{A}$ can be transformed easily into an instance of $\mathcal{B}$

$\equiv$

$\mathcal{A}$ is easier than $\mathcal{B}$

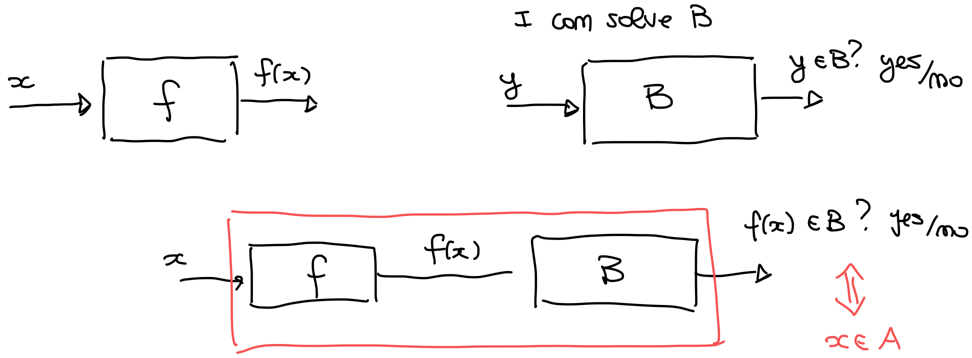
Def: Given $A, B \subseteq \mathbb{N}$

the problem $x \in A$ reduces to the problem $x \in B$

A reduces to B, written $A \leq_m B$

if there exists a total computable function $f: \mathbb{N} \to \mathbb{N}$

s.t. $\forall x \in \mathbb{N}$ $\quad x \in A$ iff $f(x) \in B$



OBSERVATION: Given $A, B \subseteq \mathbb{N}$ $\quad A \leq_m B$

(i) if B is recursive $\Rightarrow$ A is recursive

(ii) if A is not recursive $\Rightarrow$ B not recursive $\leftarrow$

proof

(i) let B recursive

$$\Downarrow \quad \chi_B(x) = \begin{cases} 1 & x \in B \\ 0 & \text{otherwise} \end{cases} \qquad \text{computable}$$

$$\Downarrow$$

$$\chi_A(x) = \chi_B(f(x))$$

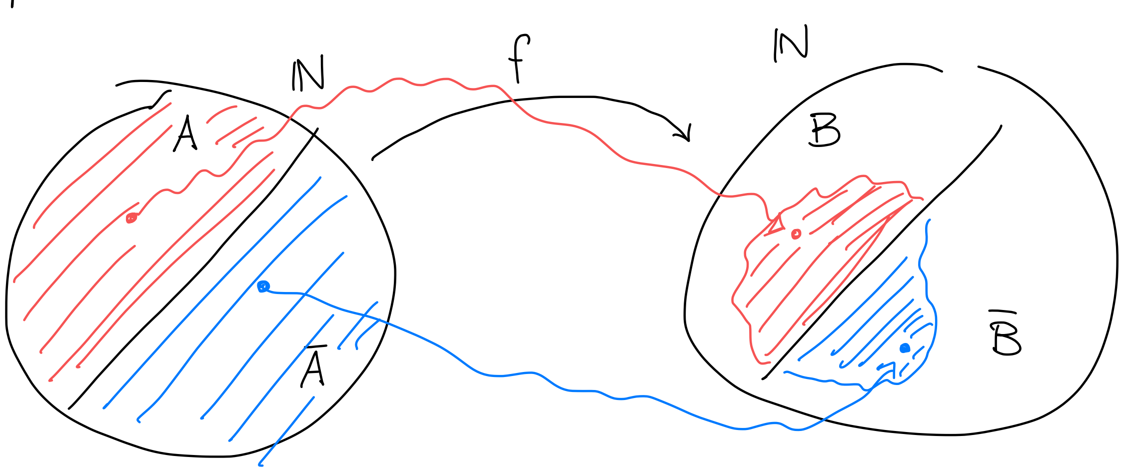
where $f : \mathbb{N} \to \mathbb{N}$ is the reduction function [total computable]

$x \in A$ iff $f(x) \in B$

$\Rightarrow$ χ_A is computable (by composition)

$\Rightarrow$ A is recursive

(ii) equivalent to (i)



Example : $K = \{x \mid x \in W_x\}$ not recursive

$T = \{x \mid \varphi_x \text{ total}\}$

$K \leq_m T$

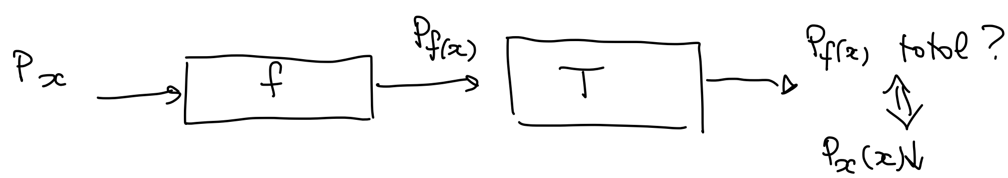
$\uparrow \qquad \uparrow$

assume that we have

if we can construct for every P_x a program $P_{f(x)}$

$P_x(x)\downarrow$ iff $P_{f(x)}$ is total

using T we can answer to the question $P_x(x)\downarrow$?



intuitively

def $P_{f(x)}(y)$:
$P_x(x)$
return 1

$P_x(x)\downarrow$ $P_{f(x)}(y)$ is the constant 1

$P_x(x)\uparrow$ $P_{f(x)}(y)\uparrow$ for all y

Formally $g(x, y) = 1(\varphi_x(x))$

$= 1(\Psi_U(x, x))$

computable

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

$\forall x, y \quad \varphi_{f(x)}(y) = g(x, y) = 1(\varphi_x(x))$

f is the function reducing K to T

i.e. $x \in K$ iff $f(x) \in T$ $\forall x$

we decompose it

① $\rhd$ $x \in K$ then $f(x) \in T$

② $\rhd$ $x \notin K$ then $f(x) \notin T$

① if $x \in K$ then $f(x) \in T$

let $\boxed{x \in K}$ $\rightsquigarrow$ $\varphi_x(x)\downarrow$ $\rightsquigarrow$ $\varphi_{f(x)}(y) = g(x,y) = \mathbb{1}(\varphi_x(x))$
$= 1$ $\forall y$

$\rightsquigarrow$ $\varphi_{f(x)}$ is total $\rightsquigarrow$ $\boxed{f(x) \in T}$

② if $x \notin K$ then $f(x) \notin T$

let $\boxed{x \notin K}$ $\rightsquigarrow$ $\varphi_x(x)\uparrow$ $\rightsquigarrow$ $\varphi_{f(x)}(y) = g(x,y) = \mathbb{1}(\varphi_x(x))\uparrow$ $\forall y$

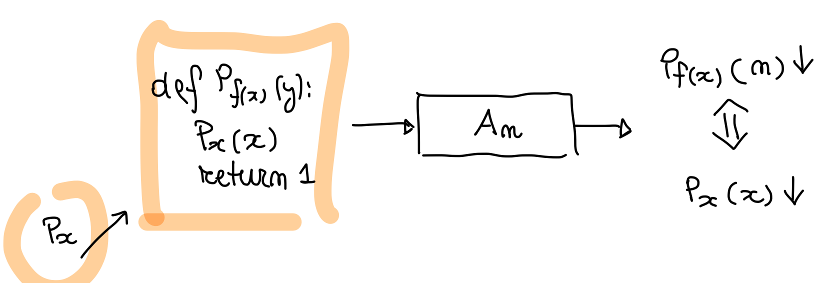
$\rightsquigarrow$ $\varphi_{f(x)}$ is not total $\rightsquigarrow$ $\boxed{f(x) \notin T}$

$\Rightarrow$ $K \leq_m T$
$\uparrow$
not recursive
$\Big\}$ $\Rightarrow$ T not recursive

Example: (input problem)

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

$K \leq_m A_n$



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

$$g(x,y) = \mathbb{1}(\varphi_x(x)) = \mathbb{1}(\psi_U(x,x))$$

computable

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

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

s is the reduction function of $K \leq_m A_n$

* If $x \in K$ then $s(x) \in A_n$

let $x \in K \implies \varphi_x(x)\downarrow \implies \forall y \quad \varphi_{s(x)}(y) = g(x,y) =$

$= \mathbb{1}(\varphi_x(x)) = 1$

$\implies$ in particular $\varphi_{s(x)}(n)\downarrow \implies f(x) \in A_n$

* if $x \notin K$ then $s(x) \notin A_n$

let $x \notin K$ then $\varphi_x(x)\uparrow \implies \forall y \quad \varphi_{s(x)}(y) = \mathbb{1}(\varphi_x(x))\uparrow$

$\implies \varphi_{s(x)}(n)\uparrow \implies s(x) \notin A_n$

$\Downarrow$

$\left.\begin{array}{l} K \leq_m A_n \\ \uparrow \\ \text{not recursive} \end{array}\right\} \implies A_n$ is not recursive

Example : $\quad ONE = \{x \mid \varphi_x = \mathbb{1}\}$

$P_x \to \boxed{ONE} \to$ P_x is a correct implementation of $\lambda x.1$

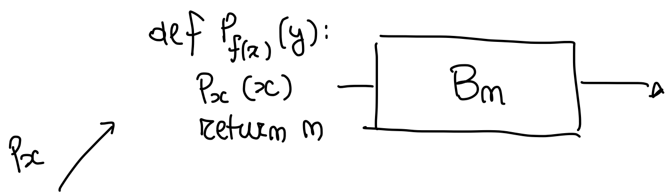
$K \leq_m ONE \qquad$ same reduction function as before

$\hookrightarrow$ ONE not recursive

Example (OUTPUT PROBLEM) : $n \in \mathbb{N}$

$$B_n = \{ x \in \mathbb{N} \mid n \in E_x \}$$ (programs which output n for some input)

We show $K \leq_m B_n$



$P_{f(x)}(y) = n$ for some y

$\Updownarrow$

$x \in W_x$ i.e. $P_x(x)\downarrow$

We define

$$g(x,y) = \mathbb{1}(\varphi_x(x)) \cdot n = \begin{cases} n & \text{if } \varphi_x(x)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

$$= \mathbb{1}(\varphi_U(x,x)) \cdot n$$

computable

$\hookrightarrow$ by smn theorem we get $s: \mathbb{N} \to \mathbb{N}$ total computable such that $\forall x,y$ $\varphi_{s(x)}(y) = g(x,y) = \begin{cases} n & \varphi_x(x)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$

$\boxed{s \text{ reduces } K \leq_m B_n}$ $\Rightarrow$ B_n not recursive

* if $x \in K$ then $\varphi_x(x)\downarrow$ hence $\varphi_{s(x)}(y) = n$ $\forall y \in \mathbb{N}$
  $\Rightarrow n \in E_{s(x)} \Rightarrow s(x) \in B_n$

* if $x \notin K$ then $\varphi_x(x)\uparrow$ hence $\varphi_{s(x)}(y)\uparrow$ $\forall y \in \mathbb{N}$
  $\Rightarrow n \notin E_{s(x)} = \emptyset \Rightarrow s(x) \notin B_n$