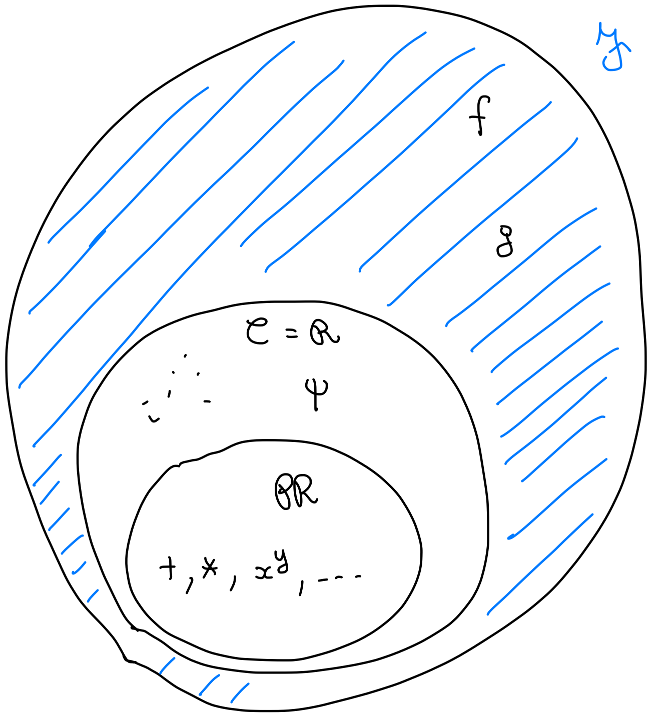
COMPUTABILITY (21/11/2023)

* Recursive and Recursively enumerable sets



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

$$g(x) = \begin{cases} 1 & \text{if } W_x = \mathbb{N} \\ 0 & \text{otherwise} \end{cases}$$

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

$\uparrow$ programs

$X = \{x \mid \varphi_x = \text{fact}\}$

$X = \{x \mid P_x \text{ has linear complexity}\}$

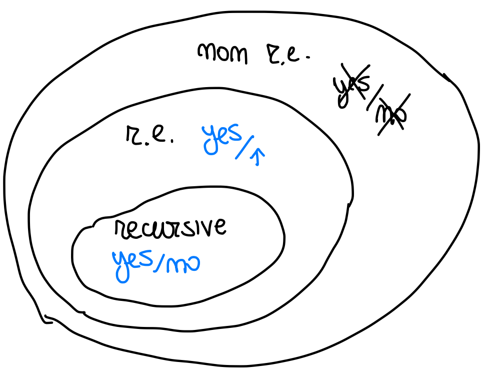
$X = \{x \mid P_x \text{ does not modify register } R_J\}$

$X = \{x \mid P_x \text{ executes each of its instructions for at least one input}\}$

$\vdots$

answer yes/no : <u>decidable</u> properties / <u>recursive</u> set

answer yes/$\uparrow$: <u>semidecidable</u> properties / <u>recursively enumerable</u> sets (r.e.)



* Recursive Sets

A set $A \subseteq \mathbb{N}$ is <u>recursive</u> 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)

<u>Examples</u>

$\mathbb{N}$	recursive	$\chi_{\mathbb{N}}(x) = 1 \quad \forall x$	computable
$\emptyset$	"	$\chi_{\emptyset}(x) = 0 \quad \forall x$	"
$\mathbb{P}$	"	$\chi_{\mathbb{P}}(x) = \overline{sg}(rm(2, x))$	
$\vdots$			

* OBSERVATION : All finite sets $A \subseteq \mathbb{N}$ are recursive

<u>proof</u>

let $A = \{x_0, x_1, \dots, x_k\}$

$$\chi_A(x) = \overline{sg}\left(\prod_{i=0}^{m} |x - x_i|\right)$$ computable

$$\boxed{\begin{aligned} K &= \{x \in \mathbb{N} \mid \varphi_x(x)\downarrow\} \\ &= \{x \in \mathbb{N} \mid x \in W_x\} \end{aligned}}$$

NOT RECURSIVE

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

<u>OBSERVATION</u> : Let $A, B \subseteq \mathbb{N}$ recursive sets. Then

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

(ii) $A \cap B$ are recursive

(iii) $A \cup B$

proof

(i) $\chi_{\bar{A}}(x) = \begin{cases} 1 & \text{if } \underbrace{x \in \bar{A}}_{x \notin A} \\ 0 & \text{otherwise } (x \in A) \end{cases} = \overline{sg}(\chi_A(x))$ computable

(ii), (iii) (see decidable predicates)

* REDUCTION

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

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

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

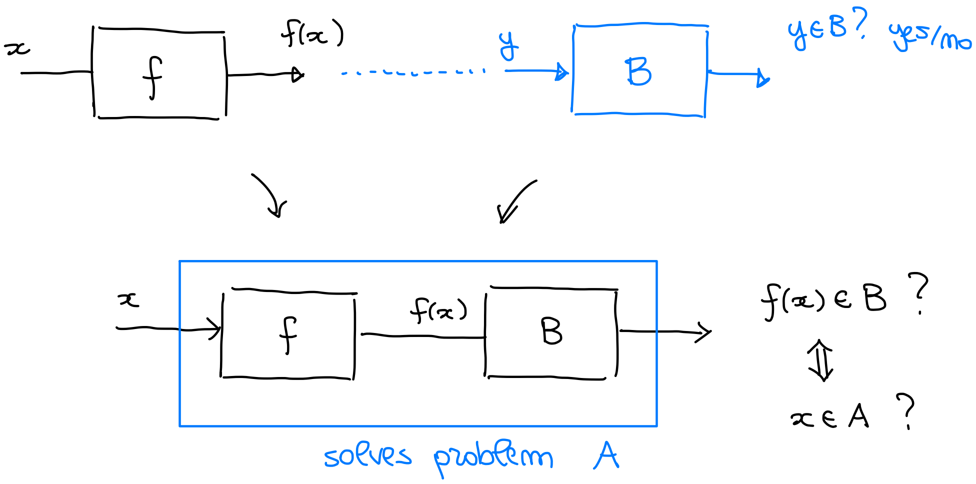
we say the problem $x \in A$ reduces to "$x \in B$"

(A reduces to B)

if there is a total computable function $f : \mathbb{N} \to \mathbb{N}$ s.t.

$\forall x \in \mathbb{N}$

$$x \in A \quad \text{iff} \quad f(x) \in B$$



$f(x) \in B$?

$\Updownarrow$

$x \in A$?

In this case $\quad A \leq_m B$

OBSERVATION : Let $A, B \subseteq \mathbb{N}$ $\quad A \le_m B$

(i) if B is recursive then A is recursive

(ii) if A not recursive then B not recursive

proof

(i) let B recursive

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

since $A \le_m B$ there is a total computable $f : \mathbb{N} \to \mathbb{N}$ s.t.

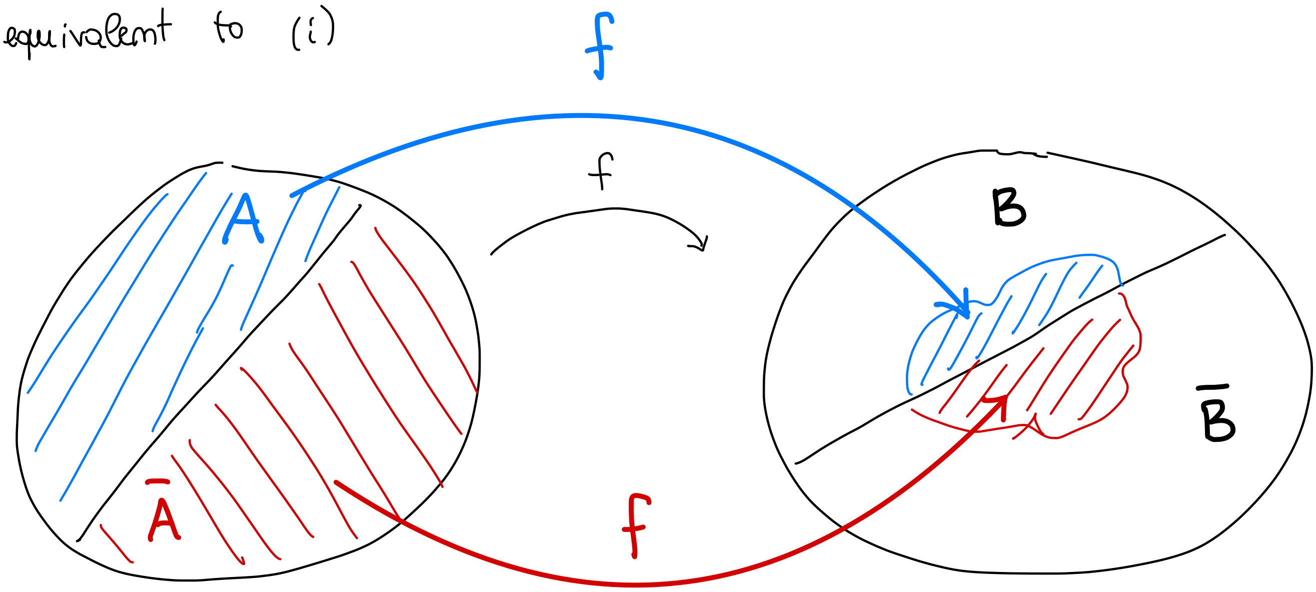
$\forall x \qquad x \in A$ iff $f(x) \in B$

Then
$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} = \chi_B(f(x))$$

computable by composition

$\leadsto$ A is recursive

(ii) equivalent to (i)



EXAMPLE : $K = \{x \mid x \in W_x\} = \{x \mid \varphi_x(x)\downarrow\}$ not recursive

$T = \{x \mid W_x = \mathbb{N}\} = \{x \mid \varphi_x \text{ total}\}$

$\boxed{K \leq_m T}$?

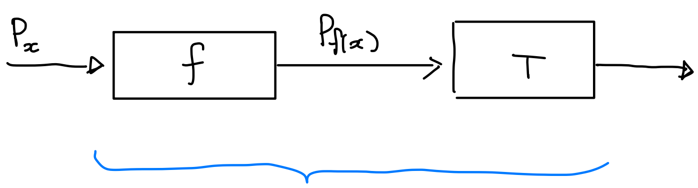
assume that we have

P_x always defined? yes/no

given P_x we construct $P_{f(x)}$ s.t.

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

then we could construct



this solves K

$P_{f(x)}$ is total? yes/no

$\Updownarrow$

$P_x(x)\downarrow$

The idea for defining f ....

$P_x \rightsquigarrow$

```
def P_f(x) (y):
    P_x (x)
    return 1
```

$P_x(x)\downarrow$

$\rightsquigarrow P_{f(x)}(y)\downarrow 1 \quad \forall y$

total

$P_x(x)\uparrow$

$\rightsquigarrow P_{f(x)}(y)\uparrow$

non total

Formally

$$g(x,y) = \mathbb{1}(\varphi_x(x)) \qquad\qquad \mathbb{1}(x) = 1 \quad \forall x$$

$$= \mathbb{1}(\Psi_U(x,x))$$

computable

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

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

We claim that $f: \mathbb{N} \to \mathbb{N}$ is the reduction function for $K \leq_m T$

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

$*$ if $x \in K \rightsquigarrow f(x) \in T$

if $x \in K \rightsquigarrow \varphi_x(x)\downarrow \rightsquigarrow \varphi_{f(x)}(y) = 1 \quad \forall y \rightsquigarrow$

$\rightsquigarrow \varphi_{f(x)}$ total i.e. $f(x) \in T$

$*$ if $x \notin K \rightsquigarrow f(x) \notin T$

if $x \notin K \rightsquigarrow \varphi_x(x)\uparrow \rightsquigarrow \varphi_{f(x)}(y)\uparrow \quad \forall y$

$\rightsquigarrow \varphi_{f(x)}$ not total i.e. $f(x) \notin T$

Therefore f is the reduction function for $K \leq_m T$

hence, since K not recursive then T is not recursive.

<u>EXAMPLE</u> (input problem)

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

$K \leq_m A_n$

$P_x \rightsquigarrow$

```
def P_f(x) (y):
    P_x(x)
    return 1
```

defined on n iff $P_x(x)\downarrow$

- $P_x(x)\downarrow \rightsquigarrow P_{f(x)}(y)\downarrow \ \forall y$ in particular $P_{f(x)}(n)\downarrow$
- $P_x(x)\uparrow \rightsquigarrow P_{f(x)}(y)\uparrow \ \forall y$ in particular $P_{f(x)}(n)\uparrow$

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

$$g(x,y) = 1\!\!1(\varphi_x(x)) = 1\!\!1(\psi_U(x,x)) \quad \text{computable}$$

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

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

The function f is the reduction function for $K \leq_m A_n$

* $x \in K \rightsquigarrow f(x) \in A_n$

  if $x \in K$ then $\varphi_x(x)\downarrow$ Therefore $\varphi_{f(x)}(y) = \mathbb{1}(\varphi_x(x)) = 1 \quad \forall y$

  In particular $\varphi_{f(x)}(n)\downarrow$ thus $f(x) \in A_n$

* $x \notin K \rightsquigarrow f(x) \notin A_n$

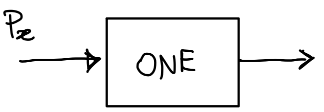
  if $x \notin K$ then $\varphi_x(x)\uparrow$. Therefore $\varphi_{f(x)}(y) = \mathbb{1}(\varphi_x(x))\uparrow \quad \forall y$

  In particular $\varphi_{f(x)}(n)\uparrow$. Thus $f(x) \notin A_n$

$\Rightarrow$ $K \leq_m A_n$ since K not recursive, A_n is not recursive

* EXERCISE : $A_n \leq_m K$ (home)

<u>EXAMPLE</u> : $ONE = \{x \mid \varphi_x = \mathbb{1}\}$

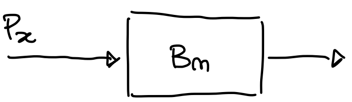


P_x is a correct implementation of $\mathbb{1}$

$K \leq_m ONE$ same reduction function as before

<u>EXAMPLE</u> : (OUTPUT PROBLEM)

Let $n \in \mathbb{N}$. Consider $B_n = \{x \mid n \in E_x\}$ not recursive



does P_x provide n as output for some input ?

Show

$K \leq_m B_n$

$P_x \longrightarrow \boxed{f} \longrightarrow P_{f(x)}$ outputs n iff $P_x(x)\downarrow$

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

Define

$$g(x,y) = n * \mathbb{1}(\varphi_x(x)) = n * \mathbb{1}(\psi(x,x)) \quad \text{computable}$$

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

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

f is the reduction function for $K \le_m B_n$

* if $x \in K$ then $\varphi_x(x)\downarrow$. Thus

$$\varphi_{f(x)}(y) = n * \mathbb{1}(\varphi_x(x)) = n \quad \forall y$$

Thus

$$n \in E_{f(x)} = \{n\}$$

hence $f(x) \in B_n$

* if $x \notin K$ then $\varphi_x(x)\uparrow$. Thus

$$\varphi_{f(x)}(y) = n * \mathbb{1}(\varphi_x(x)) \uparrow \quad \forall y$$

Thus

$$n \notin E_{f(x)} = \emptyset$$

hence $f(x) \notin B_n$

We conclude $K \le_m B_n$, hence B_n not recursive

$\square$

EXERCISE

① there exists $f: \mathbb{N} \to \mathbb{N}$ total computable s.t.

$$|W_{f(x)}| = 2x \qquad \forall x$$
$$|E_{f(x)}| = x$$

② Functions computed by programs which can only jump forward

$$I_i : \quad J(m, n, t) \qquad t > i$$

are all total.

(what if we allow only for backward steps ?)