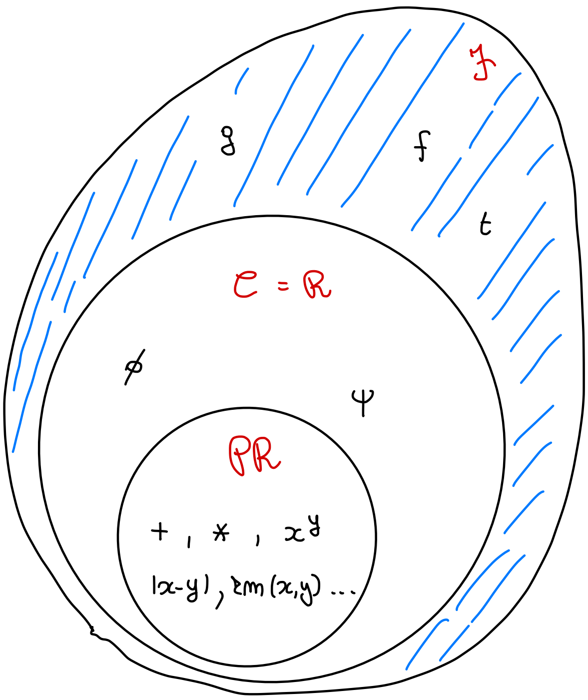
COMPUTABILITY (26/11/2024)

* RECURSIVE AND RECURSIVELY ENUMERABLE SETS



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

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

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

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

↑ set of programs

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

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

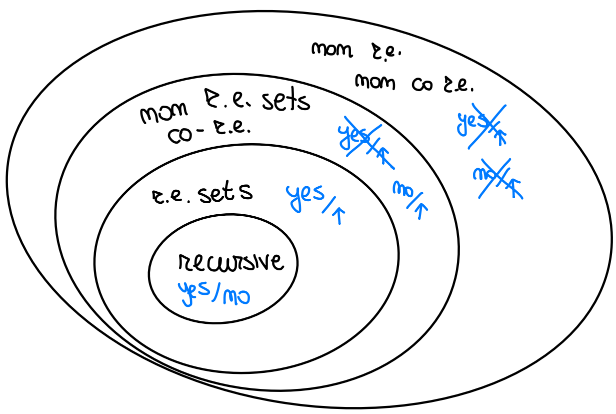
$X = \{ x \mid P_x$ access register $R_j \}$

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

⋮

answer yes / no : decidable properties / recursive sets

answer yes / ↑ : semi-decidable properties / recursively enumerable set (r.e. set)



* Recursive Sets

Let $A \subseteq \mathbb{N}$ a be set. It is recursive if the characteristic function

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

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

(i.e. "$x \in A$" decidable)

<u>Examples</u>:

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

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

<u>proof</u>

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

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

$\square$

OBSERVATION: Let $A, B \subseteq \mathbb{N}$ A, B recursive. Then

1. $\bar{A} = \mathbb{N} \setminus A = \{x \in \mathbb{N} \mid x \notin A\}$
2. $A \cup B$ are recursive
3. $A \cap B$

<u>proof</u> (same as for decidable properties)

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

2, 3 same idea (exercise)

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

NOT RECURSIVE

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

not computable

* REDUCTION

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

$\mathcal{A}$ reduces to $\mathcal{B}$

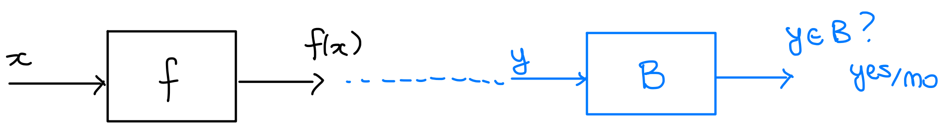
every instance of $\mathcal{A}$ can be transformed easily into an istance of $\mathcal{B}$

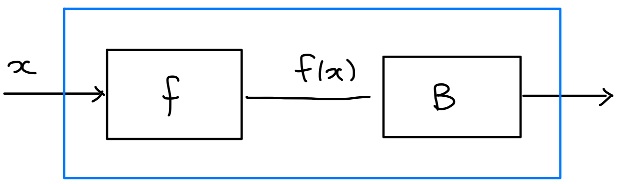
Def: Given $A, B \subseteq \mathbb{N}$

we say that the problem "$x \in A$" reduces to "$x \in B$" (or simply A reduces to B), written $A \leq_m B$ if there is a total computable function $f : \mathbb{N} \to \mathbb{N}$

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

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





$f(x) \in B$? yes/no

$\Updownarrow$

$x \in A$

solves A

OBSERVATION: Let $A, B \subseteq \mathbb{N}$ and $A \leq_m B$

(i) if B is recursive then A is recursive

(ii) if A is not recursive then B is not recursive

proof

(i) if B is recursive

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

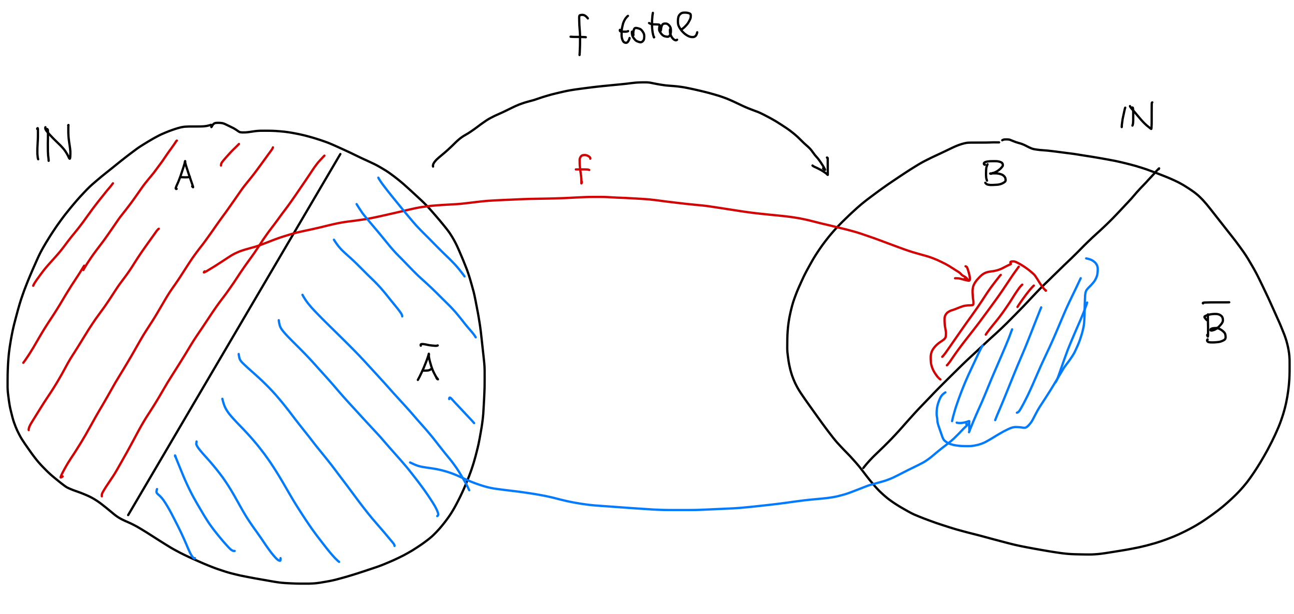
since $A \leq_m B$ there is a reduction function $f : \mathbb{N} \to \mathbb{N}$ total computable

s.t. $\forall x \in \mathbb{N}$ $x \in A$ 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.

(ii) equivalent to (i).



$A \leq_m B$

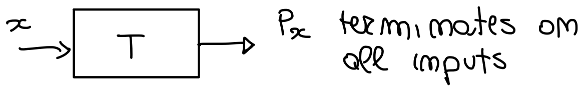
EXAMPLE :

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

$$T = \{ x \mid W_x = \mathbb{N} \} = \{ x \mid \forall y.\ \varphi_x(y)\downarrow \}$$

$\boxed{K \leq_m T}$

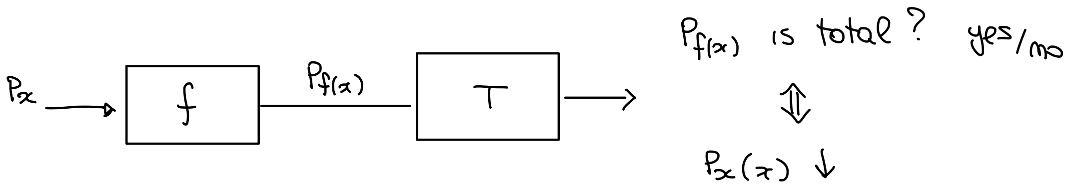
assume that we have



given P_x I construct a program $P_{f(x)}$ s.t.

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

the we can solve " $P_x(x)\downarrow$ "



We can define f

$P_x \rightsquigarrow$

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

$P_x(x)\downarrow$

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

$P_x(x)\uparrow$

$\rightsquigarrow P_{f(x)}(y)\uparrow \ \forall y$
(not total)

Formally

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

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

computable.

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

$\forall x, y$

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

The function f is the reduction function for $K \leq_m T$, i.e.

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

* $x \in K \Rightarrow f(x) \in T$

if $x \in K$ then $\varphi_x(x)\downarrow$ and thus $\varphi_{f(x)}(y) = \varphi_x(x)\downarrow \ \forall y$

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

* $x \notin K \Rightarrow f(x) \notin T$

if $x \notin K$ then $\varphi_x(x)\uparrow$ and thus $\varphi_{f(x)}(y) = \varphi_x(x)\uparrow \ \forall y$

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

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

and since K is not recursive we deduce T is not recursive.

<u>EXAMPLE</u> (INPUT PROBLEM)

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

$K \leq_m A_n$ ($\rightsquigarrow A_n$ is not recursive)

$P_x \rightsquigarrow$ $\boxed{\begin{array}{l}\text{def } P_{f(x)}(y): \\ \quad \text{return } P_x(x)\end{array}}$

$(P_x(x)\downarrow ?)$

$P_x(x)\downarrow$

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

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

$P_x(x)\uparrow$

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

in particular $\varphi_{f(x)}(n)\uparrow$

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

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

$$= \varphi_{f(x)}(y)$$

for $f: \mathbb{N} \to \mathbb{N}$ total and computable given by smn theorem and f is the reduction function for $K \leq_m A_m$

* if $x \in K$ $\overset{?}{\rightsquigarrow}$ $f(x) \in A_m$

  then $\varphi_{f(x)}(y) = \varphi_x(x)\downarrow \quad \forall y$

  in particular $\varphi_{f(x)}(m)\downarrow$ hence $f(x) \in A_m$

* if $x \in K$ $\overset{?}{\rightsquigarrow}$ $f(x) \notin A_m$

  then $\varphi_{f(x)}(y) = \varphi_x(x)\uparrow \quad \forall y$

  in particular $\varphi_{f(x)}(m)\uparrow$ hence $f(x) \notin A_m$

$\square$

It also holds

$$A_m \leq_m K \qquad \text{(exercise)}$$

while

$$T \not\leq_m K$$

<u>EXAMPLE</u> :

$$ONE = \{ x \mid \varphi_x = \mathbb{1} \}$$

$P_x \longrightarrow \boxed{ONE} \longrightarrow$ yes/no

is P_x a sound implementation of the constant $\mathbb{1}$

$$K \leq_m ONE$$

<u>idea</u> :

$P_x \rightsquigarrow$

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

↗ constant 1 if $P_x(x)\downarrow$

↘ always undefined hence $\neq \mathbb{1}$ otherwise

⇝ formalise this into a proper reduction (EXERCISE) home

[solution]

Define

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

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

computable

Hence by smn-theorem there is $f: \mathbb{N} \to \mathbb{N}$ total computable such that $\forall x, y$

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

We claim that f is a reduction function for $K \leq_m ONE$

In fact

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

  Thus $\forall y$
  $$\varphi_{f(x)}(y) = \mathbb{1}(\varphi_x(x)) = 1$$

  therefore $\varphi_{f(x)} = \mathbb{1}$ and hence $f(x) \in ONE$

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

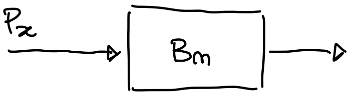
  Thus $\forall y$
  $$\varphi_{f(x)}(y) = \mathbb{1}(\varphi_x(x))\uparrow$$

  therefore
  $$\varphi_{f(x)} = \emptyset \neq \mathbb{1}$$

  and hence $f(x) \notin ONE$

EXERCISE : (OUTPUT PROBLEM)

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



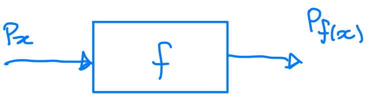
does P_x provides m as output for some input ?

show

$K \leq_m B_m$ [home]

[solution]

Idea :



outputs m iff $P_x(x)\downarrow$

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

Define

$$g(x,y) = m * 1\!\!1(\varphi_x(x)) = m * 1\!\!1(\psi(x,x)) \qquad \text{computable}$$

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

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

f is the reduction function for $K \leq_m$ ONE , in fact

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

$$\varphi_{f(x)}(y) = m * 1\!\!1(\varphi_x(x)) = m$$

Therefore $m \in E_{f(x)} = \{m\}$ and hence $f(x) \in B_m$

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

$$\varphi_{f(x)}(y) = m * 1\!\!1(\varphi_x(x))\uparrow$$

Therefore $m \notin E_{f(x)} = \emptyset$ and hence $f(x) \notin B_m$

We conclude $K \leq_m B_m$, hence B_m not recursive.

□

EXERCISE : URM with programs where only forward jumps are allowed

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

Show that all functions are total, hence $\mathcal{C}' \subsetneq \mathcal{C}$

What if I can only jump backward

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

[home]