COMPUTABILITY (09/12/2024)

* R.E. Sets and Reducibility

Given $A, B \subseteq \mathbb{N}$ and $A \leq_m B$, then

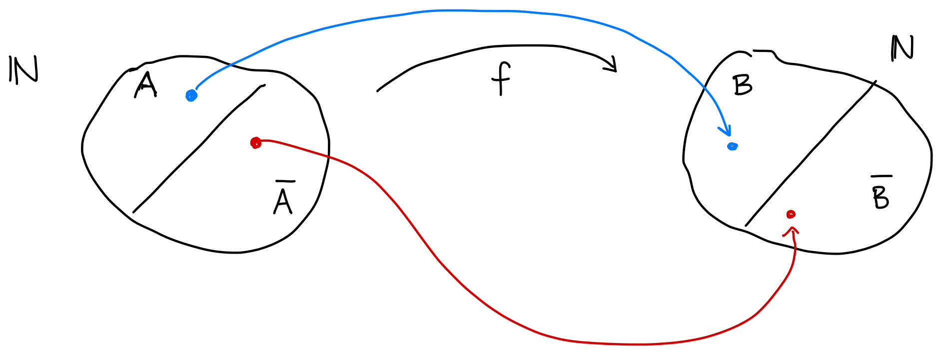
(1) if B is r.e. then A is r.e.

(2) if A is not r.e. then B is not r.e.

proof

let $A \leq_m B$ i.e. there is a total computable $f : \mathbb{N} \to \mathbb{N}$

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



(1) let B r.e. i.e.

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

then

$$sc_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \uparrow & \text{if } x \notin A \end{cases} = \underbrace{sc_B(f(x))}_{\text{computable by composition}}$$

computable

computable

hence sc_A is r.e.

(2) equivalent to (1)

$\square$

* Recursively Enumerable : WHY ?

enumerable / countable $\qquad |A| \leq |\mathbb{N}|$

i.e. there is a surjective (total) function $f : \mathbb{N} \to A$

$$\underbrace{f(0) \quad f(1) \quad f(2) \quad f(3) \quad \cdots\cdots}_{\text{enumeration of } A}$$

recursively enumerable $\equiv$ enumerable via a computable function f

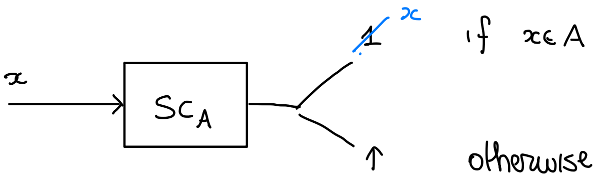
<u>Proposition</u> : Let $A \subseteq \mathbb{N}$ be a set

A r.e. $\quad$ iff $\quad$ $(A = \emptyset)$ or $(A = img(f)$ with $f : \mathbb{N} \to \mathbb{N}$ total computable $)$

<u>proof</u>

$(\Longrightarrow)$ let $A \subseteq \mathbb{N}$ be r.e., i.e.

$$sc_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \uparrow & \text{otherwise} \end{cases} \qquad \text{is computable}$$



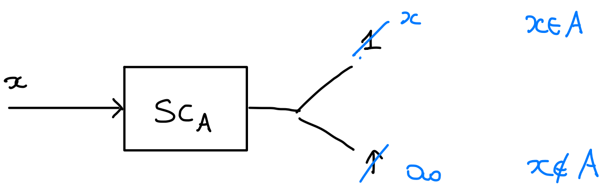
$f(x) = x \cdot sc_A(x)$ $\qquad$ computable

$img(f) = \{ f(x) \mid x \in \mathbb{N} \} = A$

~~TOTAL~~

assume $A \neq \emptyset$ (otherwise, if $A = \emptyset$ we conclude immediately)

and let $a_0 \in A$



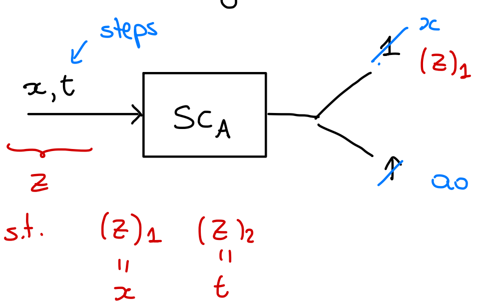
$$f(x) = \begin{cases} x & \text{if } x \in A \\ a_0 & \text{otherwise} \end{cases}$$

TOTAL

$img(f) = A$

NOT COMPUTABLE

We do the following: take $e \in \mathbb{N}$ s.t. $sc_A = \varphi_e$



if program for sc_A halts (producing 1) in t steps
$H(e, x, t)$
$H(e, (z)_1, (z)_2)$

a_0 otherwise

$$f(z) = \begin{cases} (z)_1 & \text{if } H(e, (z)_1, (z)_2) \\ a_0 & \text{otherwise} \end{cases}$$

$$= (z)_1 \cdot \chi_H(e, (z)_1, (z)_2) + a_0 \cdot \chi_{\neg H}(e, (z)_1, (z)_2)$$

f is computable

total

$$\boxed{img(f) \overset{?}{=} A}$$

$(\subseteq)$ let $x \in img(f) \overset{?}{\rightsquigarrow} x \in A$

i.e. there is $z \in \mathbb{N}$ s.t. $f(z) = x$, hence there are two possibilities

- $x = f(z) = (z)_1$ and $\underbrace{H(e, (z)_1, (z)_2)}_{sc_A((z)_1) = \varphi_e((z)_1) \downarrow 1}$

  hence $x = (z)_1 \in A$

- $x = f(z) = a_0 \in A$ ok.

$(\supseteq)$ let $x \in A \overset{?}{\rightsquigarrow} x \in img(f)$

i.e. $sc_A(x) = 1$ and thus there is $t \in \mathbb{N}$ s.t. $H(e, x, t)$

Thus, if we take z s.t. $(z)_1 = x$ and $(z)_2 = t$, [e.g. $z = 2^x \cdot 3^t$]

therefore $f(z) = (z)_1 = x$. Thus $x \in img(f)$

$(\Leftarrow)$ • if $A = \emptyset$ then A r.e. (since $SC_A = \emptyset$ always undefined $\Rightarrow$ computable)

• if $A = img(f)$ f total computable

$x \in A$ iff $\underbrace{\text{there exists}}$ $z \in \mathbb{N}$ s.t. $f(z) = x$

then

$$SC_A(x) = \mathbb{1}(\underbrace{\mu z. \ |f(z) - x|}_{\substack{1 \ \cancel{\downarrow} \quad \text{if } x \in img(f) = A \\ \uparrow \quad \text{otherwise}}}) \qquad \text{computable}$$

Thus A is r.e. $\square$

<u>OBSERVATION</u> : let $A \subseteq \mathbb{N}$ be a set.

A r.e. iff $A = dom(f)$ f computable

(i.e.

$W_0 \quad W_1 \quad W_2 \quad \ldots\ldots$ enumeration of r.e. sets)

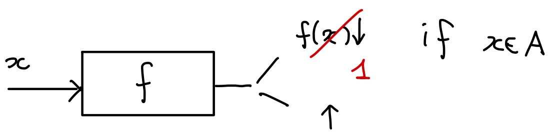
<u>proof</u>

$(\Rightarrow)$ let $A \subseteq \mathbb{N}$ be r.e., i.e.

$$SC_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \uparrow & \text{otherwise} \end{cases} \qquad \text{is computable}$$

hence $dom(SC_A) = A$, done.

$(\Leftarrow)$ let $A = dom(f)$ f computable



hence

$$SC_A(x) = \mathbb{1}(f(x)) \qquad \text{computable}$$

Therefore A is r.e. $\square$

OBSERVATION : let $A \subseteq \mathbb{N}$

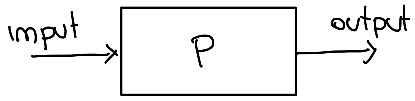
A r.e. iff $A = img(f)$ f computable

[EXERCISE]

* Rice - Shapiro theorem

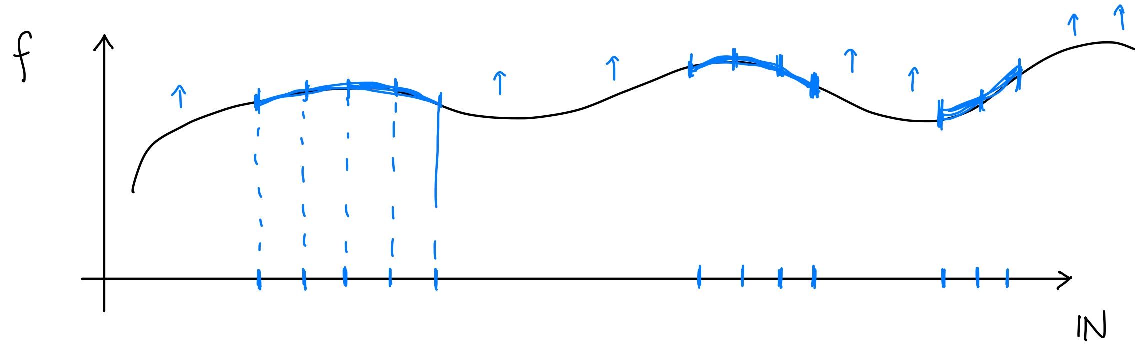
The only program properties which have a hope of being semi decidable are those which are "finitary"

depends only on the behaviour on a finite amount of inputs



Examples

- the program P on input 3 outputs value 2 — finitary
- the program P is defined on at least two inputs — finitary
- the program P is defined on every input — not finitary
- the program P produces infinitely many values — not finitary
- the program P computes the factorial — not finitary



→ finite function

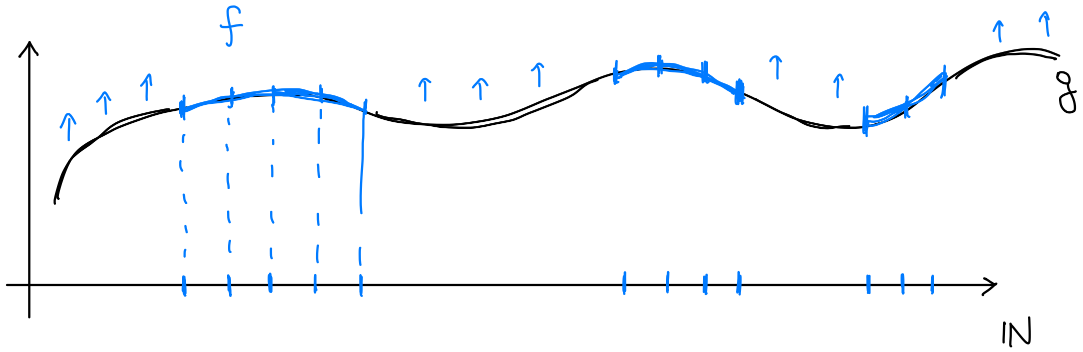
$\vartheta : \mathbb{N} \to \mathbb{N}$ is a finite function if $dom(\vartheta)$ is finite

$$\vartheta(x) = \begin{cases} y_1 & \text{if } x = x_1 \\ y_2 & \text{if } x = x_2 \\ \vdots & \vdots \\ y_n & \text{if } x = x_n \\ \uparrow & \text{otherwise} \end{cases}$$

→ <u>sub function</u>

we say that f is a subfunction of g, written $f \leq g$

if $\forall x$ if $f(x)\downarrow$ then $g(x)\downarrow$ and $f(x) = g(x)$



<u>Theorem</u> (RICE - SHAPIRO)

Let $\mathcal{A} \subseteq \mathcal{C}$ be a set of computable functions.

and $A = \{ x \in \mathbb{N} \mid \varphi_x \in \mathcal{A} \}$

if $\boxed{A \text{ is r.e.}}$ then

$$\boxed{\forall f \ (f \in \mathcal{A} \iff \exists \theta \leq f \ \ \theta \text{ finite s.t. } \theta \in \mathcal{A})}$$

↑ $\mathcal{A}$ is a finitary property

<u>proof</u> (next lesson)

<u>EXERCISE</u> : Let $f : \mathbb{N} \to \mathbb{N}$ be computable

Let $g = f$ almost everywhere (except for a finite set $\{x \mid f(x) \neq g(x)\}$ finite)

Then g is computable.

<u>proof</u>

Assume f computable

and $g(x) = f(x) \ \ \forall x \neq x_0$ while $f(x_0) \neq g(x_0)$

We distinguish two cases

(1) if $g(x_0)\uparrow$

(hence $f(x_0)\downarrow$)

then $$g(x) = f(x) + \underbrace{\mu w.\ \underbrace{\overline{sg}\ |x - x_0|}_{\substack{1 \text{ if } x = x_0 \\ 0 \text{ otherwise}}}}_{\substack{\uparrow \text{ if } x = x_0 \\ 0 \text{ otherwise}}}$$

computable by composition and minimalisation.

(2) if $g(x_0)\downarrow$ $\quad g(x_0) = y_0$

let $e \in \mathbb{N}$ s.t. $f = \varphi_e$ (program for f)

$$g(x) = \left(\mu (y,t).\ \begin{array}{l}(S(e,x,y,t) \quad \wedge \quad (x \neq x_0)) \ \vee \\ (y = y_0) \quad \wedge \quad (x = x_0)\end{array}\right)_y$$

$$= \left(\underbrace{\mu w}_{\substack{(w)_1 = y \\ (w)_2 = t}} .\ \begin{array}{l}(S(e,x,(w)_1,(w)_2) \wedge (x \neq x_0)) \ \vee \\ ((w)_1 = y_0) \wedge (x = x_0)\end{array}\right)_1$$

computable

Inductive argument for the general case.

Alternatively,

$$D = \{x \in \mathbb{N} \mid f(x) \neq g(x)\}$$

$$\underbrace{\vartheta(x)}_{\text{computable (since it is a finite function)}} = \begin{cases} g(x) & \text{if } x \in D \\ \uparrow & \text{otherwise} \end{cases}$$

Then observe

$$g(x) = \begin{cases} \vartheta(x) & \text{if } \boxed{x \in D} \\ f(x) & \text{if } \boxed{x \notin D} \end{cases}$$

decidable (since D finite)

g is computable since f, ϑ computable; $x \in D$, $x \notin D$ decidable } definition by cases.

<u>Exercise</u> :

Define a total non-computable $f : \mathbb{N} \to \mathbb{N}$ such that

$$img(f) = \{2^n \mid n \in \mathbb{N}\}$$