COMPUTABILITY (02/12/2025)

* R.e. sets and reduction

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

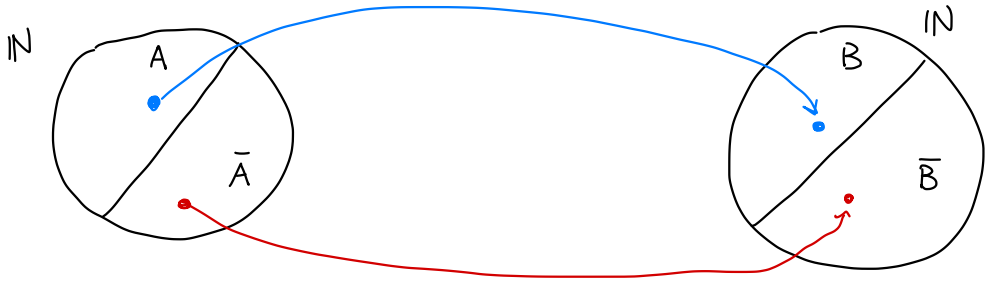
(1) if B is r.e. then A is r.e.

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

proof

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

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



(1) if B is r.e.

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

hence

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

computable (composition)

hence A is r.e.

(2) equivalent to (1).

□

* Why recursively enumerable?

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

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

$f(0)$ $f(1)$ $f(2)$ -----

enumeration of A

recursively enumerable : enumerable by a computable function

Proposition : Let $A \subseteq \mathbb{N}$ be a set

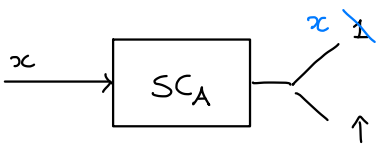
A is r.e. iff $A = \emptyset$ or

($A = \text{img}(f)$ with f total computable)

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

($\Rightarrow$) Let $A \subseteq \mathbb{N}$ be r.e. set

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



if $x \in A$

otherwise

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

→ computable

→ $\text{img}(f) = A$

not total

Assume $A \neq \emptyset$, fix $a_0 \in A$

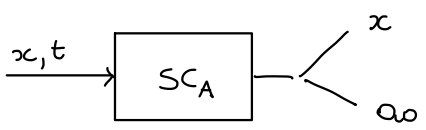
$$f(x) = \begin{cases} x & \text{if } sc_A(x) = 1 \quad (x \in A) \\ a_0 & \text{otherwise} \end{cases}$$

→ total

→ $\text{img}(f) = A$

not computable

Fix $e \in \mathbb{N}$ s.t. $\varphi_e = sc_A$



if $Pe(x) \downarrow 1$ in t steps

otherwise

$$f(\underbrace{x,t}_{z}) = \begin{cases} x & \text{if } H(e,x,t) \\ a_0 & \text{otherwise} \end{cases}$$

$(z)_1 = x$

$(z)_2 = t$

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

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

f is

- total
- computable (composition of computable functions)
- $\boxed{img(f) = A}$

($\subseteq$) let $x \in img(f)$ $\rightsquigarrow$ $x \in A$

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

- $x = f(z) = (z)_1$ with $H(e,(z)_1,(z)_2)$ true

  $\rightsquigarrow$ $Pe((z)_1) \downarrow$ thus $sc_A((z)_1) = 1$

  hence $x = (z)_1 \in A$
- $x = f(z) = a_0 \in A$

($\supseteq$) let $x \in A$ $\rightsquigarrow$ $x \in img(f)$

hence $sc_A(x) = 1 \downarrow$ and thus $Pe(x) \downarrow$ in some number t of steps

i.e. $H(e,x,t)$ is true.

If we take $z \in \mathbb{N}$ s.t. $(z)_1 = x$ and $(z)_2 = t$ $\quad [z = 2^x 3^t]$

then $H(e_1, (z)_1, (z)_2)$ and thus $f(z) = (z)_1 = x$

Thus $x \in img(f)$.

($\Leftarrow$)

- if $A = \emptyset$ then A is r.e. (since $\emptyset$ is finite hence recursive and thus r.e.)

- if $A = img(f)$ with total and computable

$$sc_A(x) = \begin{cases} 1 & \text{if } x \in A \iff \exists z.\ f(z) = x \\ \uparrow & \text{otherwise} \end{cases}$$

$$= \mathbb{1}\left(\underbrace{\mu z.\ |f(z) - x|}_{\substack{z \text{ s.t. } f(z) = x \text{, if it exists} \\ \uparrow \text{ otherwise}}} \right)$$

computable, hence A is r.e.

□

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

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

proof

($\Rightarrow$) let A be r.e.. Then $sc_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \uparrow & \text{otherwise} \end{cases}$ computable

and $A = dom(sc_A)$

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

$$sc_A(x) = \begin{cases} 1 & \text{if } x \in A \iff f(x)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

$$= \mathbb{1}(f(x))$$ computable by composition

□

(hence $W_0, W_1, W_2, \dots$ enumeration of r.e. sets).

EXERCISE : Let $A \subseteq \mathbb{N}$

A r.e. $\Leftrightarrow$ $A = \text{img}(f)$ with $f : \mathbb{N} \to \mathbb{N}$ computable

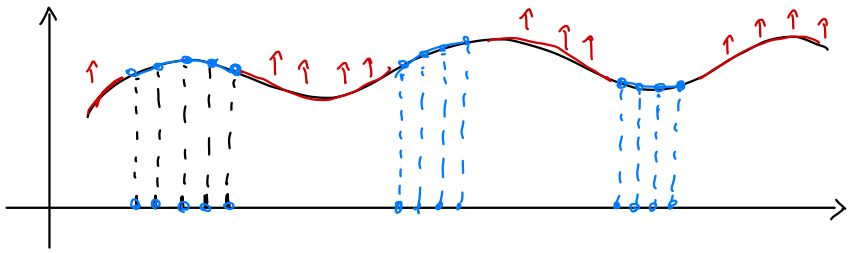
* RICE - SHAPIRO'S THEOREM

The only properties of the behaviour of programs which can be semi-decidable are the finitary properties

↑ depends only on the behaviour on a finite amount of inputs

Examples :

- program P on input 1 produces output 9 — finitary
- " " produces output 9 on some input — finitary
- " " " at least two outputs — finitary
- program P is defined on every input — not finitary
- program P computes the factorial — not finitary



→ finite function

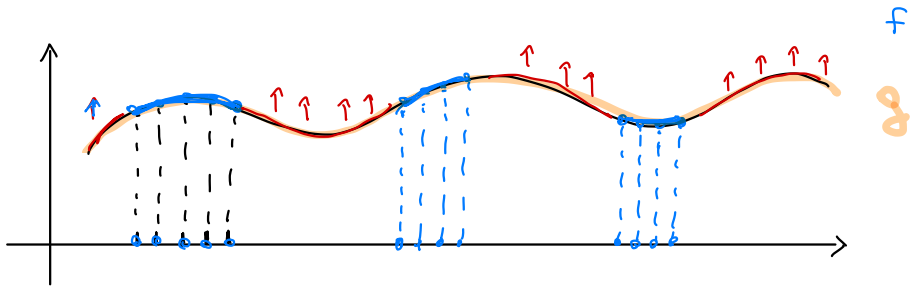
$\vartheta : \mathbb{N} \to \mathbb{N}$ is a finite function if $\text{dom}(\vartheta)$ is finite

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

subfunction

given two functions $f, g : \mathbb{N} \to \mathbb{N}$ we say that f is a subfunction of g, written $f \subseteq g$, if

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



Theorem (Rice - Shapiro)

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

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

if A r.e. then

$$\forall f \quad (f \in \mathcal{A} \iff \exists \vartheta \subseteq f, \vartheta \text{ finite } \vartheta \in \mathcal{A})$$

property $\mathcal{A}$ is finitary

proof (next lesson)

EXERCISE: Let $f: \mathbb{N} \to \mathbb{N}$ be computable, let $g = f$ almost every where ($\{ x \mid f(x) \neq g(x) \}$ is finite).

Then g is computable.

proof

Assume f computable

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

(1) if $g(x_0)\uparrow$ and $f(x_0)\downarrow$

$$g(x) = f(x) + \mu w. \overline{sg}|x - x_0|$$

computable

$\overline{sg}|x-x_0|$: 1 if $x = x_0$; 0 otherwise

$\mu w. \overline{sg}|x-x_0|$: $\uparrow$ if $x = x_0$; 0 otherwise

(1) $g(x_0) = y_0 \in \mathbb{N}$

let $e \in \mathbb{N}$ be such that $f = \varphi_e$

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

$$= \left(\mu w. \begin{array}{lll} (S(e,x,(w)_1,(w)_2) & \wedge & (x \neq x_0)) \vee \\ ((w)_1 = y_0) & \wedge & (x = x_0)) \end{array} \right)_1$$

computable

An inductive reasoning allows one to conclude for a finite number of points

$D = \{x \mid f(x) \neq g(x)\}$ finite $|D| = d$

by induction on D

$(d = 0)$ $f = g$ hence g computable

$(d \to d+1)$ take $x_0 \in D$

$$g'(x) = \begin{cases} g(x) & \text{if } x \neq x_0 \\ f(x) & \text{if } x = x_0 \end{cases}$$

then $\{x \mid g'(x) \neq f(x)\} = D \setminus \{x_0\}$

$|$ ——— $| = |$ —— $| = d$

hence, by inductive hypothesis, g' is computable.

But g' differs from g only on x_0, hence, for the first part also g is computable. □

Alternatively,

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

$$\theta(x) = \begin{cases} g(x) & x \in D \\ \uparrow & \text{otherwise} \end{cases}$$

finite hence computable

observe that

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

$\vartheta(x)$: computable

$f(x)$: computable

$x \in D$: decidable (D is finite, hence recursive)

$x \notin D$: decidable (D is finite, hence recursive)

computable since it is defined by cases using

- decidable predicates
- computable functions.