COMPUTABILITY (04/12/2023)

* Recursively enumerable sets and reducibility

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

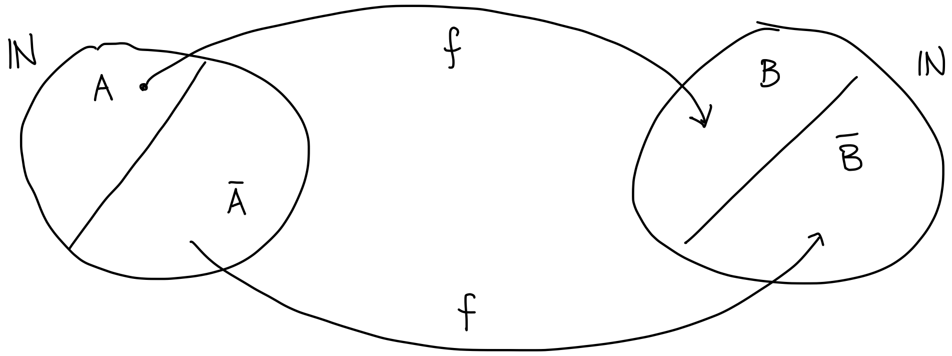
(1) if B is r.e. then A is r.e.

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

proof

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

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



(1) let B is r.e.

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

then

$$SC_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \uparrow & \text{if } x \notin A \end{cases} = \underbrace{\underset{\text{computable}}{SC_B}\,(\underset{\text{computable}}{f(x)})}_{\text{composition computable}}$$

hence SC_A computable

↳ A is r.e.

(2) equivalent to (1)

□

* Why recursively enumerable ?

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

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

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

recursively enumerable $\rightsquigarrow$ enumerable by a computable function

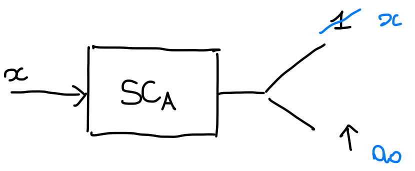
<u>Proposition</u>: Let $A \subseteq \mathbb{N}$ be a set

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

<u>proof</u>

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

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



if $x \in A$

if $x \notin A$

$f(x) = x * SC_A(x) \qquad$ computable

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

NOT total

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

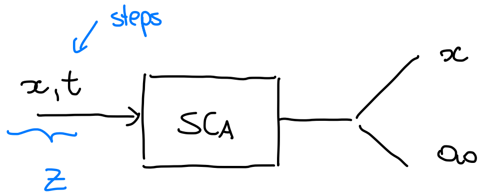
$$f(x) = \begin{cases} x & \text{if } x \in A \\ a_0 & \text{otherwise} \end{cases}$$

total

NOT COMPUTABLE

$img(f) = A$

We proceed as follows : fix $e \in \mathbb{N}$ s.t. $\varphi_e = SC_A$



if $P_e(x)\downarrow$ in t steps

otherwise

$(z)_1 = x \qquad (z)_2 = t$

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

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

f is

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

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

$\downarrow$ there is z s.t. $x = f(z)$, hence there are two possibilities

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

  hence $P_e((z)_1)\downarrow$, thus $SC_A((z)_1)\downarrow 1$

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

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

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

$\downarrow$ $SC_A(x) = 1\downarrow$ and thus $P_e(x)\downarrow$ for a suitable number of steps t

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

Therefore if we take $z \in \mathbb{N}$ st. $(z)_1 = x$, $(z)_2 = t$

$f(z) = (z)_1 = x$ (e.g. $z = 2^x \cdot 3^t \cdot \ldots$)

thus $x \in img(f)$

(⇐)

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

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

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

then

$$SC_A(x) = \mathbb{1}\Big(\underbrace{\mu z.\ |f(z) - x|}_{\begin{cases} 1 \ \cancel{\downarrow} & \text{if } x \in img(f) = A \\ \uparrow & \text{otherwise} \end{cases}}\Big)$$

computable

$\Downarrow$

A is r.e.

$\square$

<u>OBSERVATION</u>: let $A \subseteq \mathbb{N}$

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

(hence

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

<u>proof</u>

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

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

then $A = dom(SC_A)$, as desired.

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

$$x \longrightarrow \boxed{f} \longrightarrow \begin{cases} \overset{1}{f(x)} \in \mathbb{N} & \text{if } x \in A \\ \uparrow & \text{otherwise} \end{cases}$$

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

hence A r.e.

$\square$

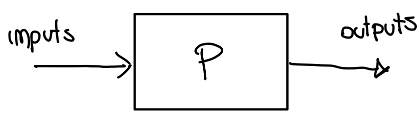
EXERCISE: Let $A \subseteq \mathbb{N}$

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

* Rice - Shapiro's theorem

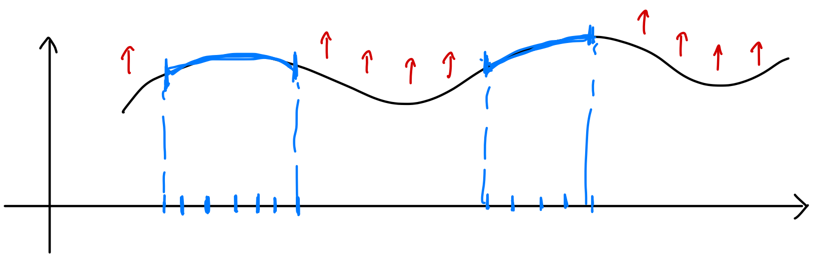
The only properties of the behaviour of programs which can be semi-decidable are the "finitary properties"

↑ properties which depends on the behaviour on a finite number of inputs



Examples:

- the program P on input $\emptyset$ outputs value 1 — finitary
- program P is defined on at least two inputs — finitary
- program P is defined on every input — not finitary
- program P produces infinitely many values as outputs — not finitary
- the program P computes the factorial — not finitary



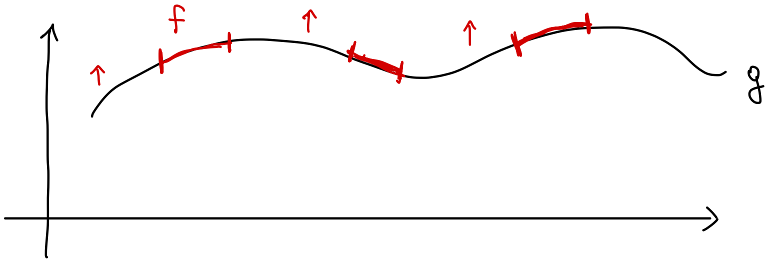
→ finite function

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

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

→ <u>sub function</u>

we say that f is a <u>subfunction</u> of g, written $f \subseteq g$,
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 let $A = \{x \mid \varphi_x \in \mathcal{A}\}$

Then if $\boxed{A \text{ is r.e.}}$ then $\Downarrow$ $\not\Uparrow$

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

↑ property is finitary

<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 $g(x) = f(x) \quad \forall x \neq x_0 \qquad f(x_0) \neq g(x_0)$

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

then $g(x) = f(x) + \underbrace{\mu w.\ \overline{sg}\,|x - x_0|}_{}$ $\quad \begin{cases} 0 & \text{if } x \neq x_0 \\ \uparrow & \text{otherwise} \end{cases}$

computable

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

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

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

computable

An inductive reasoning allows to conclude in the general case.

Alternatively:

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

$$\vartheta(x) = \begin{cases} g(x) & \text{if } x \in D \\ \uparrow & \text{otherwise} \end{cases}$$ finite function $\leadsto$ computable

observe

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

computable

decidable (D finite)

computable since it is defined by cases using a decidable predicate and computable function.

<u>Exercise</u>:

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

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