Logic R: Notes 4.

Troubles with set theory (ZFC) as a "metatheory" for mathematics.

- too many sets $\rightsquigarrow$ byproduct: pathological objects.
- independence of ZFC / undecidability in ZFC $\}$ of fundamental conjectures.

<u>Reaction</u>:

- restrict to objects, whose existence is not problematic:
  - computable objects.

<u>Computability</u>:

Objects: for example <u>natural numbers</u>. represented as:

$$n \longleftrightarrow \underbrace{|||||\ldots|}_{n} \quad \text{(n-many sticks, matches)}$$

or ~~$n = (011001$~~[redacted] $n = (1011001)_2$: binary representation.

Generally:

$\emptyset \neq \Sigma$: a finite set of [redacted] "concrete" objects.. e.g. $\Sigma = \{0, 1\}$.
("alphabet)

$\Sigma^* = \{\underbrace{\text{finite tuples of elements of } \Sigma}\}$ (
words over Σ $\leftarrow$ still concrete objects.
(~~computable~~)

For example

$$\mathbb{N} \approx \Sigma^* \text{ for } \Sigma = \{1\} \quad \text{or} \quad \mathbb{N} \approx \Sigma^* \text{ for } \Sigma = \{0, 1\}.$$

Other ~~computable~~ "concrete" objects:

- subsets of Σ^*, but: not all! (like in ZFC)

LR-N4/2

Intention: identify $\underbrace{\text{concrete}}_{\text{computable}}$ subsets of Σ^*

Similarly: $f: \Sigma^* \longrightarrow \Sigma^*$ we want to focus on "concrete" = computable functions.

$\boxed{\text{Computable} = ?}$

Turing Machine TM : an abstract ~~comp~~ computer.

Alan Turing... died ~1956 ?

[there are many other equivalent formalizations of computability]

Let Σ : a finite alphabet.

Turing machine M over Σ consists of:

(1) working heads (scanners/writers) $G_0, \dots, G_n$ (głowice robocze)

(2) working $\underbrace{\text{tapes}}_{\text{taśmy}}$ $T_1, \dots, T_n$; input tape T_0.

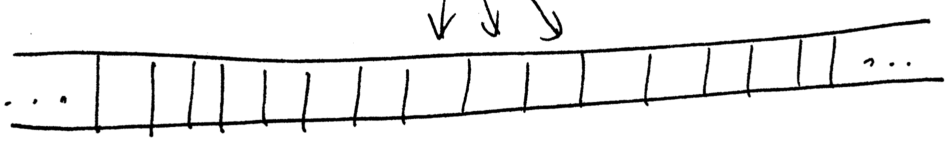
5. Wymagania techniczne i edytorskie

5.1 Zaleca się przygotowywanie prac dyplomowych przy użyciu programu TeX, z uwagi na przystosowanie tego programu do profesjonalnego składu tekstów matematycznych. Dopuszczalne jest przygotowywanie prac dyplomowych przy użyciu programu Microsoft Word lub podobnych edytorów tekstu, pod warunkiem zachowania zasad składu tekstów matematycznych.

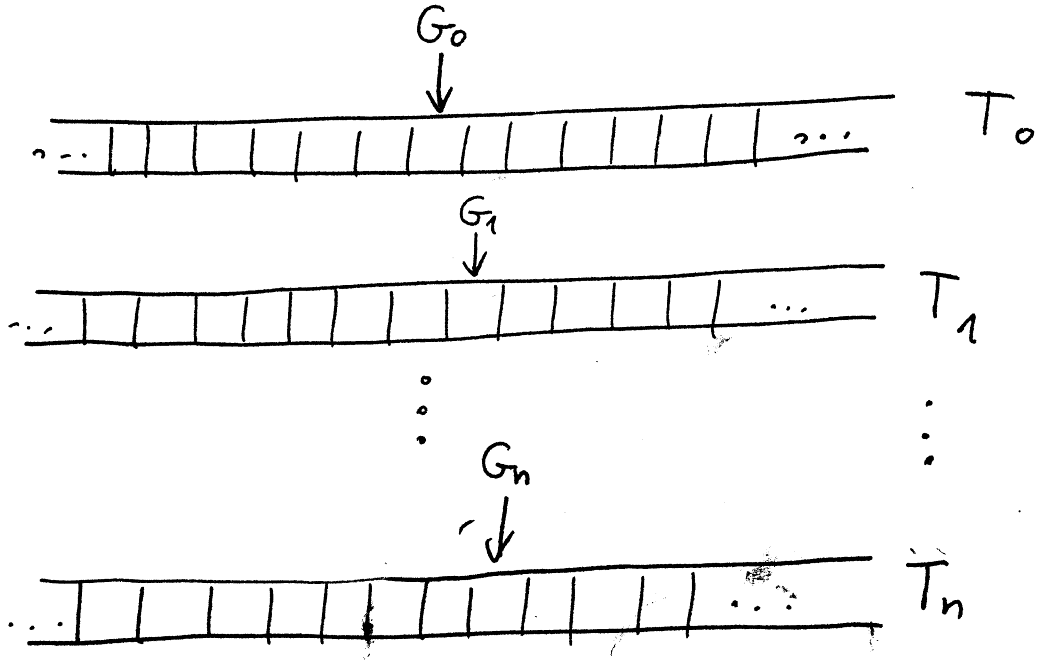
5.2 Strona tytułowa pracy dyplomowej powinna być zgodna ze wzorem umieszczonym na stronie Instytutu Matematycznego (zakładka Praca dyplomowa).

5.3 Strony pracy dyplomowej powinny być numerowane zaczynając od strony tytułowej.

Tape T_i: divided into cells, left- and right-infinite.



M:



Each head G_i sees a single cell of T_i [in any given moment], each cell contains a letter $\in \Sigma$ or is empty blanc B

(3). A finite set S of states of M

- A transition function $f: S \times (\Sigma \cup \{B\})^{n+1} \longrightarrow S \times (\Sigma \cup \{B\})^{n+1} \times \{L, R, 0\}^{n+1}$
- distinguished states $\in S$:
  - s_0 : initial state
  - end : final state
  - yes, no $\in S$.

Operation of M: in <u>time</u>, divided into <u>moments</u>: $0, 1, 2, 3, \ldots$

in steps $t = 1, 2, 3, \ldots$

Step t : ~~the~~ operation of M between moment $t-1$ and moment t.

LR-N 4/4

(1) configuration of M in moment t:

(a) each cell of each T_i contains a letter $\in \Sigma$ or (B) blanc (B)

(b) each head G_i sees a single cell of T_i with content $c_i \in \Sigma \cup \{B\}$.

(c) M is in a state $s \in S$.

(2) step $t+1$ of M (from moment t to moment $t+1$)

(a) calculates $f(s, c_0, \ldots, c_n) = (s', c_0', \ldots, c_n', v_0, \ldots, v_n)$, $v_i \in \{L, R, 0\}$

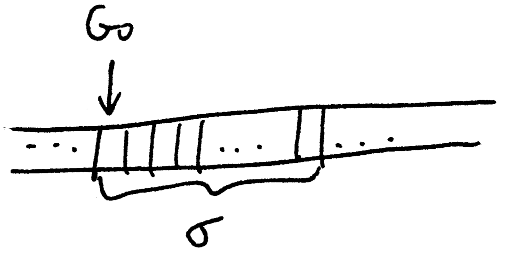
(b) replaces content c_i of the cell of T_i scanned by G_i, by c_i'.

(c) $\begin{cases} \text{if } v_i = L, & \text{moves } G_i \text{ one cell left} \\ \text{if } v_i = R, & \text{moves } G_i \text{ one cell right} \\ \text{if } v_i = 0, & \text{~~does~~ does not move } G_i. \end{cases}$

(d) changes the state of M from s to s'.

(3) configuration of M in moment $t = 0$:

- state $s = s_0$
- on T_0: an initial word $\sigma \in \Sigma^*$
  G_0 sees the cell with the first letter of σ



for $i \geq 0$ G_i sees a cell of T_i, all cells of T_i are empty (blanc).

(4) In moment t:

- if s = end, yes or no, then M ends operation. terminates

(a) if $s = \text{end}$, then the word currently written on T_0 is called the outcome of M on inputed σ.
output — initial word.

(b) If $s = \text{yes}$, we say that M accepts σ

(c) If $s = \text{no}$, we say that M rejects σ.

<u>Def</u> Let $L \subseteq \Sigma^*$. M recognizes $L \iff$
"language"

$$(\forall \sigma \in \Sigma^*) \begin{cases} \sigma \in L \Rightarrow M \text{ accepts } \sigma \\ \sigma \notin L \Rightarrow M \text{ rejects } \sigma \end{cases}$$

Def. Let $f : \Sigma^* \xrightarrow{\;\theta\;} \Sigma^*$. M ~~co~~ computes $f \iff \forall \sigma \in \Sigma^*$
partial function (i.e. $\mathrm{Dom}\, f \subseteq \Sigma^*$)

$$\begin{cases} f(\sigma)\downarrow \Rightarrow \text{on input } \sigma\ M \text{ terminates with output } f(\sigma) \\ f(\sigma)\uparrow \Rightarrow \text{on input } \sigma\ M \text{ does not terminate its operation.} \end{cases}$$

where:

$$f(\sigma)\downarrow \;=\; \text{"}\sigma \in \mathrm{Dom}\, f\text{"}, \quad f(\sigma)\uparrow = \text{"}\sigma \notin \mathrm{Dom}\, f\text{"}.$$

<u>Def.</u>

(1) $L \subseteq \Sigma^*$ is TM-computable $\iff \exists M{:}\,\mathrm{TM}$ M recognizes L

(2) $f : \Sigma^* \xrightarrow{\;\theta\;} \Sigma^*$ is TM-computable $\iff \exists M{:}\,\mathrm{TM}$ M computes f.

<u>Example</u> Let $\Sigma = \{1\}$, $L = \{\underbrace{1\ldots1}_{n} : n \text{ even}\}$ ($\approx$ {even numbers} $\subseteq \mathbb{N}$)

L is TM-Computable:

M with 1 tape only: T_0

Go, state $s = s_0$

$$\ldots|B|B|\underbrace{1|1|1|1|\ldots|1}_{n}|B|B|\ldots$$

transition function: (a) $f(s_0, B) = (\text{yes}, B, 0)$

(b) $f(s_0, 1) = (s_1, 1, R)$

(c) $f(s_1, B) = (no, B, 0)$

(d) $f(s_1, 1) = (s_0, 1, R)$

Representation of $\mathbb{N}$:

(a) $\Sigma = \{1\}$, $\mathbb{N} \approx \Sigma^*$ (b) $\Sigma = \{0,1\}$, $\mathbb{N} \approx \Sigma^*$

so natural numbers $\approx$ words over Σ. binary representation.

Def. $L \subseteq \mathbb{N}$ is TM-computable $\iff$ $\exists M$: TM M recognises L

$f: \mathbb{N} \xrightarrow{\ \theta\ } \mathbb{N}$ is TM-computable $\iff$ $\exists M$: TM M computes f

Different approach to COMPUTABILITY.

Recursive functions $f: \mathbb{N}^n \xrightarrow{\ \theta\ } \mathbb{N}$:

(a) basic functions: $S: \mathbb{N} \longrightarrow \mathbb{N}$, $S(x) = x+1$ successor function

$O: \mathbb{N}^n \longrightarrow \mathbb{N}$, $O(\bar{x}) = 0$

$I: \mathbb{N} \longrightarrow \mathbb{N}$, $I(x) = x$, $I^n_j: \mathbb{N}^n \longrightarrow \mathbb{N}$

$I^n_j(x_1, \dots, x_n) = x_j$

(b) defining schemes:

(a) composition: Given $f(x_1, \dots, x_n)$, $g_1(\bar{y}_1), \dots, g_n(\bar{y}_n)$

obtain $h(\bar{y}_1, \dots, \bar{y}_n) = f(g_1(\bar{y}_1), \dots, g_n(\bar{y}_n))$

(b) simple recursion:

Given $f(\bar{x})$, $g(\bar{x}, y, z)$ obtain $h(\bar{x}, y)$ such that

$$\begin{cases} h(\bar{x}, 0) = f(\bar{x}) \\ h(\bar{x}, n+1) = g(\bar{x}, n, h(n, \bar{x})) \end{cases}$$

(3) operation minimum:

given $f(\bar{x}, y)$ obtain $h(\bar{x})$ such that

$$h(\bar{x}) = \min\{y : f(\bar{x}, y) = 0\}.$$

Warning to defining schemes (1)–(3):
functions f, g may be partial, then h also may be partial:

Ad (1): $h(\bar{y}_1, \dots, \bar{y}_n)\downarrow \Leftrightarrow g(\bar{y}_1)\downarrow, \dots, g(\bar{y}_n)\downarrow$ and

$$f(g(\bar{y}_1), \dots, g(\bar{y}_n))\downarrow$$

Ad (2): $h(\bar{x}, 0)\downarrow \Leftrightarrow f(\bar{x})\downarrow$

~~h(n+1,~~ $h(\bar{x}, n+1)\downarrow \Leftrightarrow h(\bar{x}, n)\downarrow$ and $g(\bar{x}, n, h(\bar{x}, n))\downarrow$

Ad (3):

$h(\bar{x})\downarrow \Leftrightarrow$ there is y s.t. $f(\bar{x}, y) = 0$ and

$$\forall y' < y\ (f(\bar{x}, y')\downarrow \text{ and } f(\bar{x}, y') \neq 0)$$

Def. Rec = the smallest family of functions $f : \mathbb{N}^n \to \mathbb{N}, n \geq 0$, containing base functions and closed under defining schemes.

- f is recursive $\Leftrightarrow f \in$ Rec

Def. $A \subseteq \mathbb{N}^n$ is recursive $\Leftrightarrow \chi_A \in$ Rec.

Examples:

$$+ : \begin{cases} x + 0 = 0 = O(x) \\ x + (n+1) = (x+n) + 1 = S(x+n) \end{cases}$$

$$\cdot : \begin{cases} x \cdot 0 = 0 \\ x \cdot (n+1) = x \cdot n + x. \end{cases}$$

- The set $\mathbb{P}$ of prime numbers is recursive.
- the function $(n \longmapsto p_n = n\text{-th prime number})$ is recursive

Proof

(1) $P(x)$: $\begin{cases} P(0)=0 \\ P(n+1)=n \end{cases}$ predecessor function $P \in Rec.$

(2) $x \dot{-} y = \begin{cases} x-y, \text{ when } x \geq y \\ 0, \text{ when } x<y \end{cases}$ $\quad \begin{cases} x \dot{-} 0 = x \\ x \dot{-} (n+1) = P(x \dot{-} n) \end{cases}$

natural subtraction

(3) Let $H(x,y) = (x \dot{-} y) + (y \dot{-} x)$: $x=y \Leftrightarrow H(x,y)=0$

(4) $f(y) \in Rec \Rightarrow f'(x) = \prod_{y<x} f(y)$ recursive.

$$\begin{cases} \prod_{y<0} f(y) = 1 = S(0) \\ \prod_{y<n+1} f(y) = f(n) \cdot \prod_{y<n} f(y) \end{cases}$$

(5) $x \in \mathbb{P} \Leftrightarrow \forall y<x \, \forall z<x \;\; y \cdot z \neq x$

$\Leftrightarrow \forall y<x \, \forall z<x \;\; H(x, y\cdot z) \neq 0$

$\Leftrightarrow g(x) = \prod_{y<x} \prod_{z<x} H(x, y \cdot z) \neq 0$

(6) Let $h(x) = \min(g(x), 1) = 1 \dot{-} (1 \dot{-} g(x))$, $h: \mathbb{N} \to \{0,1\}$

[similarly $F: \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ $\quad F(x,y) = \begin{cases} 1, & x=y \\ 0, & x \neq y \end{cases}$] $\quad x \in \mathbb{P} \Leftrightarrow h(x)=1$

$Rec \ni h = \chi_{\mathbb{P}}$ so $\mathbb{P} \in Rec.$

(7) $p_0 = 2$

$p_{n+1} = \min\{x : x > p_n \text{ and } x \in \mathbb{P}\}$

$= \min\{x : (p_n+1) \dot{-} x = 0 \text{ and } 1 \dot{-} h(x) = 0\}$

$= \min\{x : ((p_n+1) \dot{-} x) + (1 \dot{-} h(x)) = 0\}.$

Thm (1) $A \subseteq \mathbb{N}^n$ is recursive $\Leftrightarrow$ A is TM-computable

(2) $f : \mathbb{N}^n \to \mathbb{N}$ is recursive $\Leftrightarrow$ f is TM-computable.

Proof. $\Rightarrow$ obvious.

[· basic functions are TM-computable,
· TM-computable functions are closed under defining schemes]

$\Leftarrow$ (2), $n=1$. sketch.

Assume M: TM computing $f : \mathbb{N} \to \mathbb{N}$
(under some representation of natural numbers as words).

Assume M has k tapes, the set of states S, transition function F.

• $\mathrm{conf}(M)$ = [content of tapes, state(M), positions of working heads of M]
configuration

$\approx$ coded as a natural number n.

for example $n = p_0^{\varepsilon_0} p_1^{\varepsilon_1} \dots p_k^{\varepsilon_k}$ codes $\langle \varepsilon_0, \varepsilon_1, \dots, \varepsilon_k \rangle$.

We define $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

. $g(t, n)$ = configuration of M on input n, in moment t.

• $g \in \mathrm{Rec}$

Let $h(n) = \min\{t : M \text{ stops in moment } t, \text{ on input } n\}$

$\Updownarrow$

[state s recovered from $g(t,n)$ is end]

~~Therefore~~

Let $f(n)$ = content of the input tape in ~~moment~~ moment $h(n)$

(may be recursively recovered from $g(n, h(n))$).

Therefore $f \in Rek$.

<u>Def</u> $A \subseteq \mathbb{N}$ is recursively enumerable $\Leftrightarrow$

$A = \emptyset$ or $\exists f: \mathbb{N} \xrightarrow[\text{recursive}]{} \mathbb{N}$ $A = Rng(f)$

<u>Church thesis</u>. Assume $A \subseteq \mathbb{N}$.

Then A is recursive $\Leftrightarrow$ A is computable

(i.e. there is an algorithm determining, for $n \in \mathbb{N}$, if $n \in A$)

<u>Fact</u> Assume $A \subseteq \mathbb{N}$. If both

A and $\mathbb{N} \setminus A$ are recursively enumerable.

<u>Proof</u> Wlog $A \neq \emptyset \neq \mathbb{N} \setminus A$. Choose (total) recursive f, g with

$A = Rng\, f$, ~~B~~ $\mathbb{N} \setminus A = Rng\, g$.

An algorithm determining for $n \in \mathbb{N}$: if $n \in A$

1. ~~Calcul~~ Compute $f(0), g(0), f(1), g(1), \ldots$
2. When in sequence $f(0), g(0), f(1), g(1), \ldots$, <u>n appears</u>

   then <u>answer</u> if $n = f(i)$, then $n \in A$

   if $n = g(i)$, then $n \notin A$

5. Wymagania techniczne i edytorskie

5.1 Zaleca się przygotowywanie prac dyplomowych przy użyciu programu TeX, z uwagi na przystosowanie tego programu do profesjonalnego składu tekstów matematycznych. Dopuszczalne jest przygotowywanie prac dyplomowych przy użyciu programu Microsoft Word lub podobnych edytorów tekstu, pod warunkiem zachowania zasad składu tekstów matematycznych.

5.2 Strona tytułowa pracy dyplomowej powinna być zgodna ze wzorem umieszczonym na stronie Instytutu Matematycznego (zakładka Praca dyplomowa).

5.3 Strony pracy dyplomowej powinny być numerowane zaczynając od strony tytułowej.

Formally:

Let $r(n) = \min\{i : \cancel{f(i,n)}\ f(i) = n \text{ or } g(i) = n\} =$

$= \min\{i : H(f(i),n) \cdot H(g(i),n) = 0\}$

$r \in Rec$

$n \in A \iff f(r(n)) = n \iff H(f(r(n)),n) = 0$

$\iff \underbrace{1 \dot{-} H(f(r(n)),n)}_{\chi_A \in Rec.} = 1$

Thm. · Rec is countable.

· There are countably many recursively enumerable sets.

Thm. There is a recursively enumerable, non-recursive set $A \subseteq \mathbb{N}$.

Proof. 1. $\exists \underset{\text{recursive}}{f : \mathbb{N} \times \mathbb{N} \multimap \mathbb{N}}\ \forall \underset{\text{recursive}}{g : \mathbb{N} \multimap \mathbb{N}}\ \exists n\ f(n,\cdot) = g(\cdot)$

[f is called a universal recursive function]

proof of 1:

- we enumerate effectively "recipes" for recursive functions. $\alpha_0(\cdot), \alpha_1(\cdot), \alpha_2(\cdot), \ldots$
- $f(n,m) = $ (recipe α_n applied to m)

  $f \in Rec.$

2. Let $A = \{x : f(x,x) = 0\}$: recursively enumerable.

proof: an algorithm generating A:

- i-th step: perform i-many steps of computation of $f(x,x)$ for all $x \leq i$.

~~the~~ We list those $x \leq i$ such that in this stage $f(x,x)$ is computed and equals 0.

In this way we create an (infinite) recursive list ~~{~~ of natural numbers, enumerating A,
[in each stage we add finitely many members to the list]

(3) A is not recursive.

<u>proof</u> (a.a.) Suppose $\chi_A \in \mathrm{Rec}$. Then $\chi_A(\cdot) = f(n,\cdot)$ for some $n \in \mathbb{N}$

and $f(n,\cdot)$ is <u>total</u>

Then $f(n,n) = 0 \iff n \in A \iff \chi_A(n) \neq 0 \iff f(n,n) \neq 0$ $\lightning$.