CURS #6

Thursday, November 2, 2023 8:27 PM

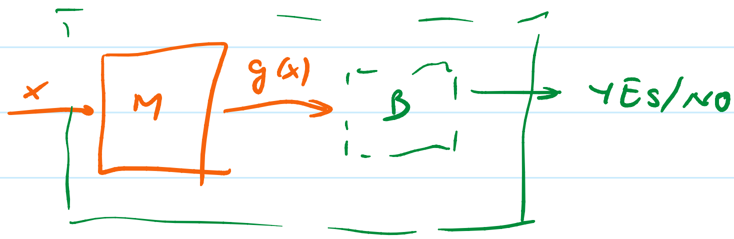
$A \leq_m B$ ddacă există o funcție $g : \Sigma^* \to \Sigma^*$

probleme de decizie.

$\{f : \Sigma^* \to \{0,1\}$ (1/0) $\}$

g calculabilă de o Mașină Turing M care produce întotdeauna un rezultat

$(\forall x) \quad x \in A \Leftrightarrow g(x) \in B$



Obs Dacă B este decisă de o Mașină Turing / alg. și $A \leq_m B$ atunci și problema A este decisă de o Mașină Turing.

A nu este decisă de o M.T și $A \leq_m B$ atunci nici B nu este decisă de o M.T

$W \leq_m K = \{ \langle x, y \rangle \mid M_x(y) \text{ se oprește} \}$

Def O problemă de decizie A este recursivă ddacă

$$1_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$ calculabilă de o Mașină Turing

Exp W, K nu sunt recursive.

Obs $A \leq_m B$ și B recursivă atunci A este recursivă

Prop A, B sunt probleme de decizie

— A recursivă

$x_0 \in B$

— A recursivă

— $B \neq \emptyset, \mathbb{Z}^*$ $\quad x_0 \in B$, $x_1 \notin B$

atunci $A \leq_m B$

Dem. A recursivă $\Rightarrow$ $\exists M$ care decide A

Construiesc g care arată că $A \leq_m B$ ($x \in A \Leftrightarrow g(x) \in B$)

$$g(x) = \begin{cases} x_0 & \text{dacă } A(x) = 1 \\ x_1 & \text{dacă } A(x) = 0 \end{cases}$$

$\boxtimes$

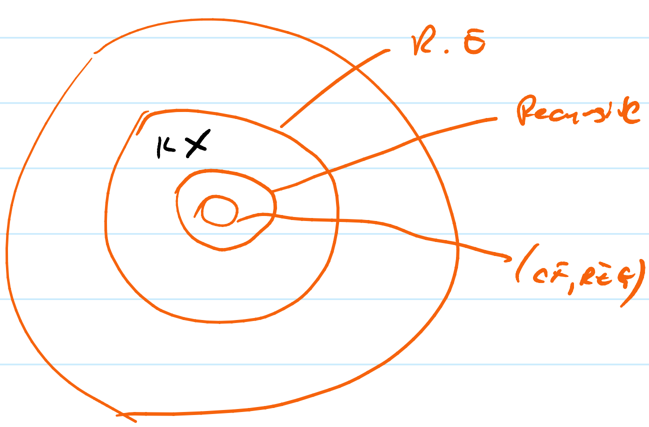
Obs Dacă $M_x(y)$ se oprește pot verifica ușor acest lucru: rulez $U(\langle x, y \rangle)$ până se oprește. Când se oprește returnez 1.

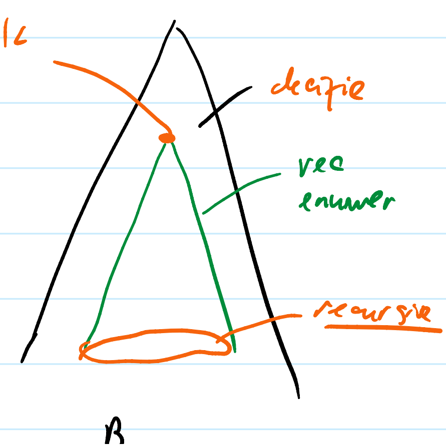
Def O problemă de decizie A se numește recursiv enumerabilă (computably enumerable) semidecidabilă

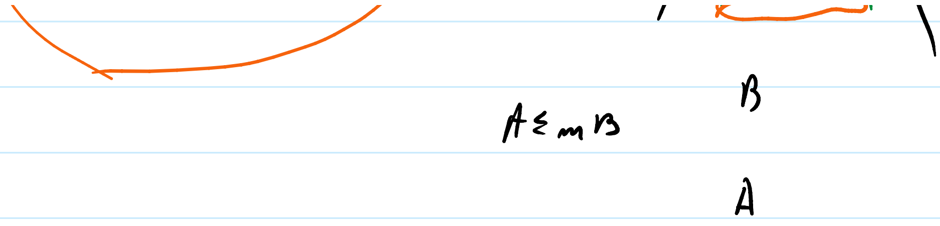
ddacă
$$f_A(x) = \begin{cases} 1 & x \in A \\ \uparrow & x \notin A \end{cases}$$
calculabilă de o Mașină Turing

Obs A recursivă $\Rightarrow$ A recursiv enumerabilă

Obs K este recursiv enumerabilă dar nu este recursivă.







$A \le_m B$

(T) Dacă A este o problemă rec. enumerabilă atunci

$$\boxed{A \le_m K}$$

Dem

A rec. enum $\Rightarrow \exists\, x$ a.î. $A = L(M_x)$

Trebuie să construiesc g a.î.

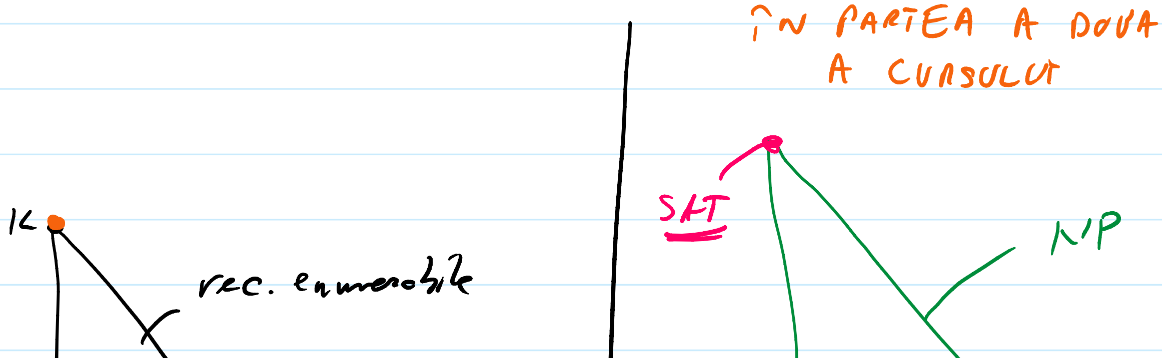
$$\boxed{y \in A \iff g(y) \in K}$$

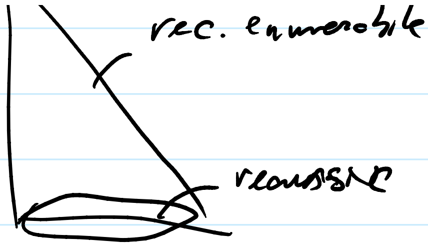
$g(y) = \langle x, y\rangle$ fct calculabilă

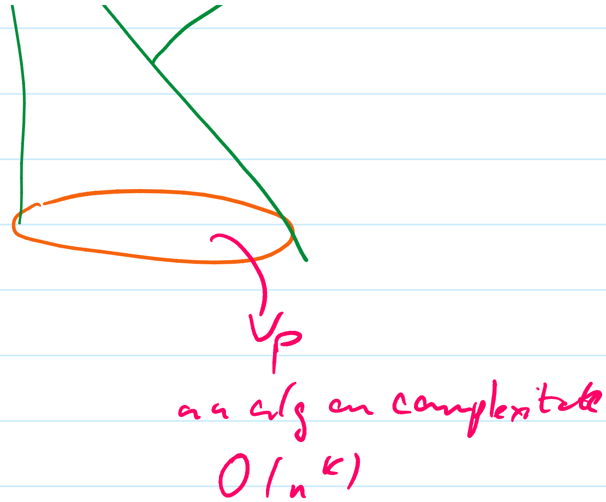
$y \in A \iff M_x(y)$ se oprește $\iff \langle x, y\rangle \in K$ ⊠

Def Problema A se numește completă pt clasa $\mathcal{C}$

- $A \in \mathcal{C}$
- $\forall B \in \mathcal{C}\quad B \le_m A$







Deosebire Știm că $REC \neq R.E.$

Nu știm că $P \neq NP$

(T) A este r.e. și $\bar{A}$ este r.e. $\Longrightarrow$ A recursivă ($\Longleftarrow$)

Dem

$$M_1 \longrightarrow M_1(x) = \begin{cases} 1 & x \in A \\ \uparrow & x \notin A \end{cases}$$

$$M_2 \longrightarrow M_2(x) = \begin{cases} 1 & x \notin A \\ \uparrow & x \in A \end{cases}$$

$M(x)$ rulează în paralel $M_1(x), M_2(x)$

$$M(x) = \begin{cases} 1 & \text{dacă } M_1(x) = 1 \\ 0 & \text{dacă } M_2(x) = 1 \end{cases}$$

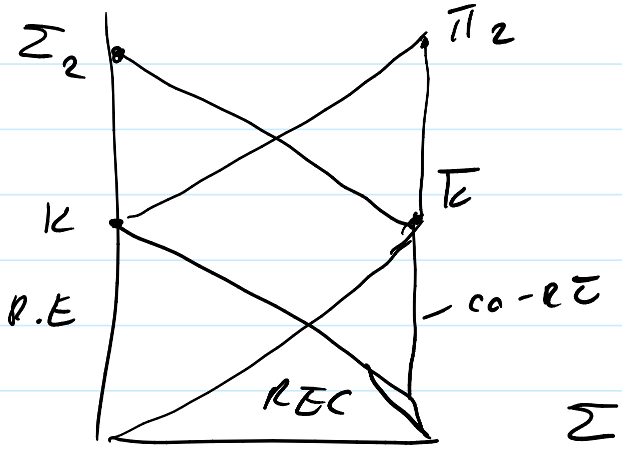
$\square$

Obs $\bar{K} = \{ \langle x, y \rangle \mid M_x(y) \text{ nu se oprește} \}$ nu este r.e.

Dem Pp $\overline{K}$ r.e, K r.e $\Big\} \Rightarrow$ K ar fi recursivă

Def A se numește co-R.E $\Leftrightarrow$ $\overline{A}$ r.e.

Obs $\overline{K}$ este completă pt co-R.E.



Σ_n, Π_n — ierarhia analitică

$$\Sigma_n = \{ A \mid \underbrace{\exists_{x_1} \forall_{x_2} \exists_{x_3} \forall}_{n \text{ cuantificatori}} \; Q(x_1 \dots x_n) \}$$

recursiv

$$\Pi_n = \{ A \mid \underbrace{\forall \exists \cdots \exists}_{n}$$

Take home O problemă de logica predicatelor cu $n \geq 2$ cuantificatori are șansa să fie mai grea decât pb. opririi.

COMPLEXITATE KOLMOGOROV

01010101 nu este aleator

010010001… nu este aleator

$\Big\} \to$ au programe simple

$\underbrace{011010010}_{n}$ $\qquad$ $\underbrace{\text{print('011010010')}}_{n+O(1)}$

$$K(y) = \min\{\,|x| : M_x(\lambda) = y\,\}$$

(T) $\exists\, c > 0$ a.î. $\forall\, y$ $K(y) \le |y| + c$

Def Un cuvânt y se numește aleator.

$$\boxed{K(y) \ge |y| + O(1)}$$

(T) – Există cuv. aleatoare oricât de lungi (majoritatea cuvintelor)

– Fcția $K(y)$ nu poate fi calculată de o M.T.

Kolmogorov Complexity
Algorithmic Information Theory.

Funcția "Busy Beaver"

RADÓ (1962) $\Sigma(n)$ o mașină Turing cu n stări care pe intrarea vidă $\{1\}$ scrie cât mai mulți de 1

$S(n)$ Mașină Turing cu n stări și alfabet $\{1\}$ care se oprește în cel mai mare nr de pași

$BB(n)$

Cum rezolvăm pb opririi dacă știm $BB(n)$

$M_x(y) \longrightarrow \underset{n \text{ stări}}{M_{f(x,y)}}(\lambda)$ o rulez $BB(n)$ pași

dacă se oprește $M_x(y) = 1$

altfel $M_x(y) = 0$

(T) $f(n) = BB(n)$ nu este calculabilă de o M.T.

$BB(1) = 1$

$BB(2) = 6$

$BB(3) = 21$

$BB(4) = 107$

$BB(5) > 47.176.870$

$BB(6) > 8 \cdot 10^{16}$

"Humanity might never know BB(7)"