Funcția Ackerman

$A: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$A(0,n) = n+1$

$A(m+1,0) = A(m,1)$

$A(m+1,n+1) = A(m, A(m+1,n))$

Funcția Collatz

$f: \mathbb{N} \to \mathbb{N}$

while $(x > 1)$
  if x par
    $x = x/2$
  else
    $x = 3x+1$
ret 0

Mulțimea A are același nr de elem cu B $\Leftrightarrow \exists\, f: A \to B$ bijectivă

(T) $\nexists$ f bijectivă $\mathbb{N} \to (0,1)$

MATIASEVICH : Se dă $p(x_1, x_n)$ cu coef întregi. Vreau să decid dacă $\exists (x_1, \ldots, x_m) \in \mathbb{Z}^m$ a.î. $p(x_1 \ldots x_m) = 0$ (Nu are niciun alg.

Problemă de decizie $\to$ Input : $x \in \Sigma^*$, Output : TRUE/FALSE

Funcții de bază $F_0(x) = 0$ (constantă), $F_1(x) = S(x)+1$ (succesor), $F_{i,n}(x_1, \ldots, x_n) = x_i$ (proiecții)

Operații :

① Compunerea : $f: \mathbb{N}^n \to \mathbb{N}$, $g_1, g_2, \ldots, g_n : \mathbb{N}^m \to \mathbb{N}$

$f \circ (g_1, g_2, \ldots, g_n)[x_1, \ldots, x_m] = f(g_1(x_1, \ldots x_m), \ldots g_n(x_1 \ldots x_m))$

② Recursie primitiva

$f(x_1, \ldots, x_n)$

$h(y_1, \ldots y_{n+2})$ $\Rightarrow g(x_1, x_2, \ldots x_n, y+1) = h(x_1, x_2, \ldots x_n, y, g(x_1, \ldots x_n, y))$

$g(x_1, \ldots x_n, 0) = f(x_1, \ldots x_n)$

③ Minimizare $g(x_1, \ldots x_n) = \begin{cases} \text{cel mai mic } y \in \mathbb{N} \text{ ai } f(x_1, \ldots, x_n) = 0 \\ \uparrow \text{ dacă } y \text{ nu există } (\uparrow = \text{nedefinit}) \end{cases}$

Primitiv recursivă : $f: \mathbb{N} \to \mathbb{N}$ dacă pot să o obțin din funcțiile de bază, prin compunere și recursie primitivă.

ACKERMAN : e prim. rec pt că nu crește suf de repede de la un pct.

$f: \mathbb{N} \xrightarrow{\circ} \mathbb{N}$ poate să nu ret. val (fct. parțială)

Parțial recursivă : fct care poate fi obținută din fct. de bază, din operațiile de compunere, recursie primitivă și minimizare

Recursivă : o fct. care e parțial recursivă + totală

Church : fct intuitiv calculabile $\Leftrightarrow$ fct. parțial recursive

(T) O fct $f: \mathbb{N} \xrightarrow{\circ} \mathbb{N}$ e parțial recursivă $\Leftrightarrow$ f e calculabilă de o mașina Turing

(T) $\exists$ o mașină Turing M a.î. $M(x)$ calcl $\langle \alpha, z \rangle$ a.î. $x = \langle \alpha, z \rangle$

$M(x)$ simulează $M_\alpha(z)$. (Mașina Turing Universală)

Dacă $M_\alpha(z)$ acceptă în T pași atunci $M(x)$ acceptă în $CT\log(T)$ pași unde $C > 0$ nu depinde de z.

A o.m recursiv enumerabilă $f(x) = \begin{cases} 1, & x \in A \\ \uparrow, & x \notin A \end{cases}$ poate fi calcl de o MT (parțial rec.)

A s.m recursivă $f(x)=\begin{cases}1, & x\in A\\ 0, & x\notin A\end{cases}$ este calculabilă de o MT care se oprește ~~întotdeauna~~ oricum

Modelul MT e robust la modificări (calcl. aceleași lucruri indif. de model)

$\exists$ MT a.î. $M(y)$, $y=<i,x>$ a.i. $M(y)$ simulează $M_i(x)$

Funcții necalculabile (de mașini Turing)

$U:\mathbb{N}\to\mathbb{N}$, $U(x)=\begin{cases}0, & \text{dacă } M_x(x)=1\\ 1, & \text{altfel}\end{cases}$ (mașina x pe intrarea x) $\longrightarrow$ poate să meargă la infinit

(T) Funcția U nu e parțial recursivă (nu poate fi calcl de o MT)

Problema Opririi

$Halt(<\alpha,x>)=\begin{cases}1, & \text{dacă } M_\alpha(x)\downarrow \text{ (se oprește)}\\ 0, & \text{altfel}\end{cases}$

(Un program α pe un input x se termină sau rulează la ∞)
e necalculabilă

Nu avem algoritm ~~pt~~ pt problema Domino Wang pt că dacă ar exista, am avea și pt HALT

S(n) - nr max de pași efectuați de o MT cu n stări până se oprește

(T) Fct. S(n) nu poate fi calcl. de o MT.

Mulțime recursivă : $A\subset\Sigma^*$, $f_A(x)=\begin{cases}1, & x\in A\\ 0, & x\notin A\end{cases}$ poate fi calculată

Pb opririi: $K=\{<\alpha,x> \mid M_\alpha(x)\downarrow\}$ > $g_K(<\alpha,x>)=\begin{cases}1, & M_\alpha(x)\downarrow\\ \uparrow\end{cases}$ poate fi calcl. de MT

(T) K e rec. enum., dar nu e rec.

Mulțime recursiv enumerabilă : $A\subset\Sigma^*$ s.m recursiv enumerabilă $\iff g_A(x)=\begin{cases}1, & x\in A\\ \uparrow, & x\notin A\end{cases}$ calcl. de MT

A recursivă : $\exists$ MT pt A a.î. $x\in A? \to M(x)=1$; $x\in\bar{A}? \to M(x)=0$

A recursiv enumerabilă : $\exists$ MT pt A a.î. $x\in A? \to M(x)=1$; $x\in\bar{A}? \to M(x)=\uparrow$

Dacă A ~~se~~ reduce la B $\Rightarrow$ B e mai grea în complexitate. ($A\le B$)

Dacă $\left.\begin{array}{l} A\subseteq\Sigma^*, A\neq\emptyset, \Sigma^* \\ B \text{ e rec. enumerabilă} \\ A \text{ e rec.}\end{array}\right\}\Rightarrow A\le_m B$

(T) K are prop că $\forall$ A rec. enumerabilă $\Rightarrow A\le_m K$ (este completă pt. rec. enum)

(T) A rec. enum $\iff \exists$ $P(\cdot,\cdot)$ predicat recursiv a.i. $A(x)=\exists y\, P(x,y)$

A rec. enumerabilă $\not\Rightarrow \bar{A}$ rec. enumerabilă

A recursivă $\Rightarrow \bar{A}$ recursivă

(T) A recursivă $\iff$ A și $\bar{A}$ recursiv enumerabile

Complexitate polinomială P : ușor de găsit sol

Complexitate exponențială NP: e ușor de verificat sol, dar e greu de găsit

NP P $P = \bigcup_{c \geq 1} DTIME(n^c)$

$DTIME(f) = \{A \mid A$ poate fi rez de o MT a.î. $M(x)$ face $O(f(|x|))$ pași

NP: $L \subseteq \{0,1\}^*$ e în clasa NP $\Leftrightarrow \exists$ polinom p $p: \mathbb{N} \to \mathbb{N}$ și o

MT a.î. (1) $M(y)$ rulează în $O((y)^k)$ pași
(2) $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ a.î. $M(<x, u>) = 1$

Un limbaj $A \in NP \Leftrightarrow \exists f(\cdot,\cdot)$ calculabil în timp polinomial

a.î. $x \in A \Leftrightarrow \exists y \in \{0,1\}^{p(|x|)}$ a.î. $f(x,y) = y$

Mașina Turing nedeterministă: dintr-o stare pot merge în 2 stări

O problemă A se reduce la B dacă avem o fct f care duce un x din A în B și B acceptă f(x). ($A \leq_m^P B$)

NP-HARD : Pb A s.m NP-HARD, dacă $\forall B \in NP$ $B \leq_m^P A$

NP-COMPLETĂ: $A \in NP$ și $\forall B \in NP$ $B \leq_m^P A$

(T) 3-SAT e NP-COMPLETĂ (K-SAT, $k \geq 3$ NP-COMPLETĂ)

$$\left.\begin{array}{l}\text{Dacă } A \text{ e NP-COMPLETĂ} \\ B \text{ e în NP} \\ A \leq_m^P B\end{array}\right\} \Rightarrow B \text{ e NP-completă}$$

(T) 2-SAT e P

(T) Variabilele SAT sunt fie în P, fie în NPC

(T) ILP (integer linear programming) e NPC

(T) IS independent SAT e NPC ($3-SAT \leq_m^P IS$)

ϕ - m clauze, m variabile ; G graf

ϕ satisfiabilă $\Leftrightarrow \exists$ m vf independente în G

NP-COMPLETE : A s.m NPC $\Leftrightarrow$ a) $A \in NP$
b) $\forall B \in NP$ $B \leq_m^P A$

Lemă: A, B sunt NPC $\Rightarrow A \leq_m^P B$ și $B \leq_m^P A$

(T) Fie A o pb NPC. Atunci $P = NP \Leftrightarrow$ A are alg polinomial

(T) Pb CLIQUE e NPC

(T) Vertex cover e NPC

(T) $P \neq NP \Rightarrow \exists A \in NP$ a.î. A nu e NPC și $A \notin P$

DPLL - (Davis - Putnam) - alg pt SAT (backtracking)

x literar pur dacă x apare doar pozitiv sau doar negativ în ϕ

Reg 1. x lit pur => îl alegi ca variabilă și nu faci backtracking

Reg 2. Dacă vreo var apare într-o clauză unitară (de lungime 1)

=> var bine definită

[DPLL] = backtracking + clauze pure + clauze unitare

CLIQUE ≠ MAX-CLIQUE

$A \leq_T^P B$ (A se reduce Turing la B) dacă ∃ o mașină Turing care rulează în timp polinomial și folosește o subrutină pt B care decide pb A.

① $\overline{SAT} \leq_T^P SAT$

② MAX-CLIQUE $\leq_T^P$ SAT CLIQUE $\leq_m^P$ SAT

QBF (quantified boolean formula) - toate variabilele sunt cuantificate

O mult A rec. enum <=> ∃ predicat P recursiv a.î. $\forall x \in \Sigma^*$

$x \in A \iff \exists t$ a.î. $P(x,t) = TRUE$

O mulțime A e NP <=> ∃ predicat P calculabil în timp polinomial

$P(x,y) \to (|x|+|y|)^{d'}$ și un pol $g(\cdot)$ $\forall x \in \Sigma^*$

$x \in A \iff \exists\, y \in \{0,1\}^{g(|x|)}$ a.î. $P(x,y) = TRUE$

NP ~ rec. enum

P ~ rec.

$A \in NP \not\Rightarrow \overline{A} \in NP$ (clasa NP nu e închisă la complement)

PSPACE = {A | ∃ o MT determinista care decide $x \in A$ folosind spațiu de lucru $\leq g(|x|)$ }
↑polinom

Ⓣ QBF e completă pt clasa PSPACE

$P \subseteq PSPACE$

$P \subseteq \underset{CO-NP}{\overset{NP}{}} \quad PSPACE$

Ⓣ Savitch: Fie A o pb de decizie care poate fi rez. de o MT nedet care fol pe intr. x spațiu $S(x)$. Atunci ∃ o MT deterministă M' care decide pb A și M'(x) folosește cel mult spațiul $\leq S^2(|x|)$

Fct. $PARITY_n(x_1, \dots, x_n) = \begin{cases} 1, & x_1 \oplus x_2 \oplus \dots \oplus x_n = 1 \\ 0, & x_1 \oplus \dots \oplus x_n = 0 \end{cases}$

(T) Nu putem calcula $PARITY_n$ cu circuite booleene a.î. adâncimea unui circuit $\leq k$ (k ct.), nr porți $\leq poly(n)$, chiar dacă dau voie $\wedge, \vee$ de orice paritate

Pb care NU pot fi rez alg

- Matiyasevich (nu e rec) e recursiv enumerabilă
  pb nu are alg => nu e rec
- pb opririi
- pavaje Wang

O mașină Turing cu 3 benzi nu calc mai mult decât una cu 2 benzi

Probleme NP:

- 0/1 knapsack
- TSP — NPC
- colorări în graf — NPC
- ciclu hamiltonian
- SAT — NPC
- SUDOKU

Problemele NPC se reduc unele la altele

O mulțime se num. p selectivă dacă avem o fct calculabilă în timp polinomial a.î.

$f(x, y) =$ fie x, fie y

Dacă x sau y aparțin A atunci $f(x, y)$ aparține A

Scanned with CamScanner