Problema opririi

1. Metoda diagonalizarii
2. Nedecidabilitatea unui limbaj dat
3. Un limbaj care nu este Turing-acceptat

Problema opririi

Definitia 1

Multimea A se n. numarabila $\Leftrightarrow$

A este finita sau

$\exists$ f: N $\rightarrow$ A, bijectiva.

Exemple de multimi numarabile

1. $N' = \{2n \mid n \in N\}$
2. $T = \{(i,j,k) \mid i,j,k \in N\}$

Problema opririi

3. Enumerarea lui Cantor

Multimile N^2 si N au aceeasi cardinalitate (1874)

$$J : N^2 \to N, \quad J(x,y) = \frac{(x+y)(x+y+1)}{2} + x$$

x\y	0	1	2	3	4	5	…
0	0	1	3	6	10	15	…
1	2	4	7	11	16	22	…
2	5	8	12	17	23	30	…
3	9	13	18	24	31	39	…
4	14	19	25	32	40	49	…
5	20	26	33	41	50	60	…
…	…	…	…	…	…	…	…

Problema opririi

functia J este bijectiva;

inversele ei sunt:

$$K, L : N \rightarrow N,$$

$$K(z) = \mu_{x \le z} x \left[\sum_{i=0}^{z} eq(J(x,i), z) = 1 \right], \quad L(z) = \mu_{y \le z} y \left[\sum_{i=0}^{z} eq(J(i,y), z) = 1 \right],$$

$$unde: \quad eq(a,b) = \begin{cases} 1, \, daca \, a = b \\ 0, \, daca \, a \ne b \end{cases} \quad a \div b = \begin{cases} a - b, \, daca \;\; a \ge b \\ 0, \, daca \, a < b. \end{cases}$$

Problema opririi

$J(x,y)$ = numarul lui Cantor asociat perechii (x,y);

Tripletul (J,K,L):

$$J(K(z),L(z))=z; K(J(x,y))=x, L(J(x,y))=y$$

triplet de functii pereche

<u>Contraexemplu</u>

R

Problema opririi

4. Q = {m/n | m,n $\in$ N} este numarabila

Utilizam tehnica diagonalizarii

Construim o matrice:

$$\frac{1}{1}\ \frac{1}{2}\ \frac{1}{3}\ \frac{1}{4}\ \frac{1}{5}\ \ldots\ldots$$
$$\frac{2}{1}\ \frac{2}{2}\ \frac{2}{3}\ \frac{2}{4}\ \frac{2}{5}\ \ldots\ldots$$
$$\frac{3}{1}\ \frac{3}{2}\ \frac{3}{3}\ \frac{3}{4}\ \frac{3}{5}\ \ldots\ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

=> numarul rational m/n: celula aflata la intersectia liniei m cu coloana n.

Problema opririi

$$\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5} \ldots\ldots$$

$$\frac{2}{1}, \frac{2}{2}, \frac{2}{3}, \frac{2}{4}, \frac{2}{5} \ldots\ldots$$

$$\frac{3}{1}, \frac{3}{2}, \frac{3}{3}, \frac{3}{4}, \frac{3}{5} \ldots\ldots$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\frac{1}{1};\frac{1}{2};\frac{2}{1};\frac{3}{1};\frac{2}{2};\frac{1}{3};\frac{1}{4};\frac{2}{3};\frac{3}{2};\frac{4}{1};\frac{5}{1};\frac{4}{2};\frac{3}{3};\frac{2}{4};\frac{1}{5};\ldots\ldots$$

eliminam repetitiile =>

$$\frac{1}{1};\quad \frac{1}{2};\quad \frac{2}{1};\quad \frac{3}{1};\quad \frac{1}{3};\frac{1}{4};\frac{2}{3};\frac{3}{2};\frac{4}{1};\frac{5}{1};\frac{1}{5};\ldots\ldots$$

$$f(1),\ f(2),\ f(3),\ f(4),\ f(5),$$

Problema opririi

=> ∃ Limbaje care nu sunt decidabile si nici macar Turing-acceptate.

? multimea limbajelor este nenumarabila,
multimea MT este numarabila,
o MT poate recunoaste 1! limbaj.

Problema opririi

Propozitia 1

$\exists L \subseteq \Sigma^*: L \neq L(M), \forall M \in MT$

demonstratie

(i) MT = multimea masinilor Turing este numarabila

Observam ca multimea Σ^* este numarabila, $\forall \Sigma$.

$\forall M \in MT$ poate fi codificata peste un alfabet A convenabil ales.

eliminam $\forall w \in A^*: w \neq \langle M \rangle, \forall M \in MT$

$\Downarrow$

obtinem o enumerare a MT.

Problema opririi

(ii) multimea $\mathcal{B}$ a secventelor binare infinite este nenumarabila

Folosim metoda diagonalizarii

ppa ∃ f:N → $\mathcal{B}$, bijectiva a.i. $f(n)=b_n \in \mathcal{B}$ →.

n	f(n)=b_n
1	100…
2	010…
3	110…
4	001…
…	…

n	f(n)=b_n
1	<u>1</u>0000…
2	0<u>1</u>000…
3	11<u>0</u>00…
4	001<u>0</u>0…
5	1010<u>0</u>…
…	…

b=00111….

$b \neq f(n)$, $\forall n \in N$:

a n^a cifra binara din b este:

0, daca a n-a cifra binara din f(n) este 1;

1, daca a n-a cifra binara din f(n) este 0.

=> $\mathcal{B}$ este nenumarabila.

Problema opririi

(iii) multimea $\mathcal{L} = \{L \subseteq \Sigma^* \mid L = \text{limbaj}\}$ este nenumarabila

E suficient sa gasim f: $\mathcal{L} \rightarrow \mathcal{B}$, bijectiva.

Fie $\Sigma^*=\{s_1,s_2,\ldots,s_n,\ldots\}$; $\forall L \rightarrow \lambda_L$ infinita, unica, definita astfel:
cel de-al i-lea bit din λ_L este: 1, daca $s_i \in L$,
0, daca $s_i \notin L$.

Exemplu: $\Sigma=\{0,1\}$, $L=\{w \in \Sigma^* \mid \exists x \in \Sigma^*: w=0x\}$
=> $\Sigma^*=\{\lambda,0,1,00,01,10,11,000,001,010,011,100,\ldots.\}$
L={ 0, 00,01, 000,001,010,011, }
=> $\lambda_L=$ 0 1 0 1 1 0 0 1 1 1 1 0.....
=> f: $\mathcal{L} \rightarrow \mathcal{B}$: $f(L)= \lambda_L$.

Evident: f = bijectiva.

cf. (ii) $\mathcal{B}$ =nenumarabila => $\mathcal{L}$ nenumarabila.

Din (i) si (ii) => exista limbaje care nu sunt Turing-acceptate

Problema opririi

1. Metoda diagonalizarii
2. Nedecidabilitatea unui limbaj dat
3. Un limbaj care nu este Turing-acceptat

Problema opririi

O MT, M, accepta sau nu o anumita secventa de intrare, w?

PROBLEMA ACCEPTABILITATII pt. LIMBAJE RECURSIV ENUMERABILE

PROBLEMA OPRIRII

$ACC_{MT}=\{<M,w> \mid M \in MT, w \in \Sigma^*: M \text{ accepta } w\}$

Teorema 1

Limbajul $ACC_{MT}=\{<M,w> \mid M \in MT, w \in \Sigma^*: M \text{ accepta } w\}$ nu este decidabil.

Problema opririi

Observatia 1

Lb ACC_{MT} este Turing-acceptat dar nu este Turing-decidabil:

U = “Fie secventa de intrare <M,w>, unde $M \in MT$ si $w \in \Sigma^*$:

1. Se simuleaza M pe intrarea w.
2. Daca M accepta w, atunci U accepta <M,w>;
   daca M respinge w, atunci U respinge <M,w>.”

==> U recunoaste limbajul ACC_{MT} dar nu decide asupra lui:
M cicleaza pe w → U cicleaza pe <M,w>.

? M nu se opreste pe w → U respinge <M,w> →
ACC_{MT}= decidabil

Observatia 2

Problema acceptabilitatii pt MT, ACC_{MT}, se n. si PROBLEMA OPRIRII tocmai datorita acestui fapt.

Problema opririi

Observatia 3

U = exemplu de MT universala

O MT oarecare se numeste universala ⇔

ea poate simula orice MT, cu conditia sa primeasca la intrare o descriere corecta a acesteia.

Observatia 4

Teorema 1: MT acceptoare sunt mai puternice decat MT decidente.

Problema opririi

demonstratie (T1)

Ppa ca limbajul ACC_{MT} este decidabil $\Rightarrow$

H = “Fie secvenţa de intrare <M,w>, unde M $\in$MT şi w$\in\Sigma$*:

1. Se simuleaza M pe intrarea w.
2. Daca M accepta w, atunci H se opreste si accepta <M,w>;

   daca M nu accepta w, atunci H se opreste si respinge <M,w>.”

Problema opririi

D apelează H pentru a determina ce acţiune execută M atunci când primeşte, ca secvenţă de intrare w , propria sa descriere <M>;

după ce D a obţinut această informaţie ea execută exact acţiunea contrară, i.e.:

D = “Fie secvenţa de intrare <M> , unde $M \in MT$:

1. Se rulează H pe secvenţa de intrare <M,<M>>.
2. Se returnează rezultatul opus rezultatului returnat de H, adică,

   dacă H acceptă intrarea <M,<M>> (ceea ce se întâmplă atunci când M acceptă <M>) atunci D respinge;

   dacă H respinge intrarea <M,<M>> (ceea ce se întâmplă atunci când M nu acceptă <M>) atunci D acceptă <M>.”

Problema opririi

Dar MT M din secventa de intrare este oarecare: ce se intampla cand D primeste la intrare propria sa descriere <D>?

$$\Rightarrow \; D(\langle D\rangle) = \begin{cases} \text{respinge, dacă } D \text{ acceptă } \langle D\rangle. \\ \\ \text{acceptă, dacă } D \text{ nu acceptă } \langle D\rangle, \end{cases}$$

$\Rightarrow$ contradicţie

$\Rightarrow$ nici D nici H nu pot exista în realitate

$\Rightarrow$ limbajul ACC_{MT} nu este decidabil.

Problema opririi

Fie multimea tuturor masinilor Turing, MT:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	…
M_1	acc	␣	acc	␣	…
M_2	acc	acc	acc	acc	…
M_3	␣	␣	␣	␣	…
M_4	acc	acc	␣	␣	…
…	…	…	…	…	…

i.e.: celula (i,j)= $\begin{cases} \text{acc, daca } M_i \text{ accepta intrarea } \langle M_j \rangle, \\ \text{␣, dacă } M_i \text{ nu accepta intrarea } \langle M_j \rangle \text{ (respinge sau cicleaza)} \end{cases}$

Problema opririi

H	<M_1>	<M_2>	<M_3>	<M_4>	…
M_1	acc	resp	acc	resp	…
M_2	acc	acc	acc	acc	…
M_3	resp	resp	resp	resp	…
M_4	acc	acc	resp	resp	…
…	…	…	…	…	…

i.e.: celula (i,j)= rezultatul returnat de H pe intrarea <M_i,<Mj>>.

Problema opririi

H	$<M_1>$	$<M_2>$	$<M_3>$	$<M_4>$	…	$<D>$	…
M_1	acc	resp	acc	resp	…	acc	…
M_2	acc	acc	acc	acc	…	acc	…
M_3	resp	resp	resp	resp	…	resp	…
M_4	acc	acc	resp	resp	…	acc	…
…	…	…	…	…	…		…
D	resp	resp	acc	acc	…	???	…
…	…	…	…	…	…		…

Problema opririi

1. Metoda diagonalizarii
2. Nedecidabilitatea unui limbaj dat
3. Un limbaj care nu este Turing-acceptat

Problema opririi

Definiţia 2

Limbajul L $\subseteq$ **Σ*** se numeşte co-Turing-acceptat $\Leftrightarrow$

$\exists$ L' $\subseteq \Sigma^*$, L' = Turing-acceptat astfel incat L' = $\Sigma^* \setminus$ L.

Teorema 2

$\forall$ L $\subseteq \Sigma^*$: L este decidabil $\Leftrightarrow$

L este Turing-acceptat şi co-Turing-acceptat.

demonstratie ”$\Rightarrow$”

ip: L este decidabil $\Rightarrow$ L este Turing-acceptat.

dar: L este decidabil $\Rightarrow$ $\neg$**L** este decidabil

$\Rightarrow$ $\neg$**L** este Turing-acceptat

$\Rightarrow$ L este co-Turing-acceptat (cf. def.).

Problema opririi

"$\Leftarrow$"

ip: L este Turing-acceptat şi co-Turing-acceptat

$\Rightarrow$ L şi $\neg$**L** sunt Turing-acceptate (cf. def.)

$\Rightarrow \exists\ M_1, M_2 \in MT$: $L = L(M_1)$, $\neg$**L** $= L(M_2)$.

Definim MT, M, astfel :

M = "Fie cuvântul de intrare $w \in \Sigma^*$:

1. Se rulează M_1 şi M_2 în paralel pe intrarea w.
2. Dacă M_1 acceptă w atunci M acceptă w iar dacă M_2 acceptă w atunci M respinge w."

Rularea în paralel a celor doua **MT** M_1 şi M_2 : înzestrarea lui M cu 2 benzi de lucru.

Simularea se face alternativ: M simulează câte un pas al fiecărei maşini până când una dintre ele se opreşte.

M decide asupra limbajului L:

fie un cuvânt oarecare $w \in \Sigma^* \Rightarrow w \in L$ sau $w \in \neg$**L**

$\Rightarrow$ fie M_1 fie M_2 acceptă intrarea w.

Dar, conform definiţiei sale, M se opreşte oridecâteori M_1 sau M_2 acceptă cuvântul de intrare primit.

În plus, M acceptă toate secvenţele din L şi respinge toate secvenţele din $\neg$**L**

$\Rightarrow$ M decide asupra limbajului L $\Rightarrow$ L este decidabil.

Problema opririi

Corolarul 1

Limbajul $\overline{ACC}_{MT}$ NU este Turing-acceptat.

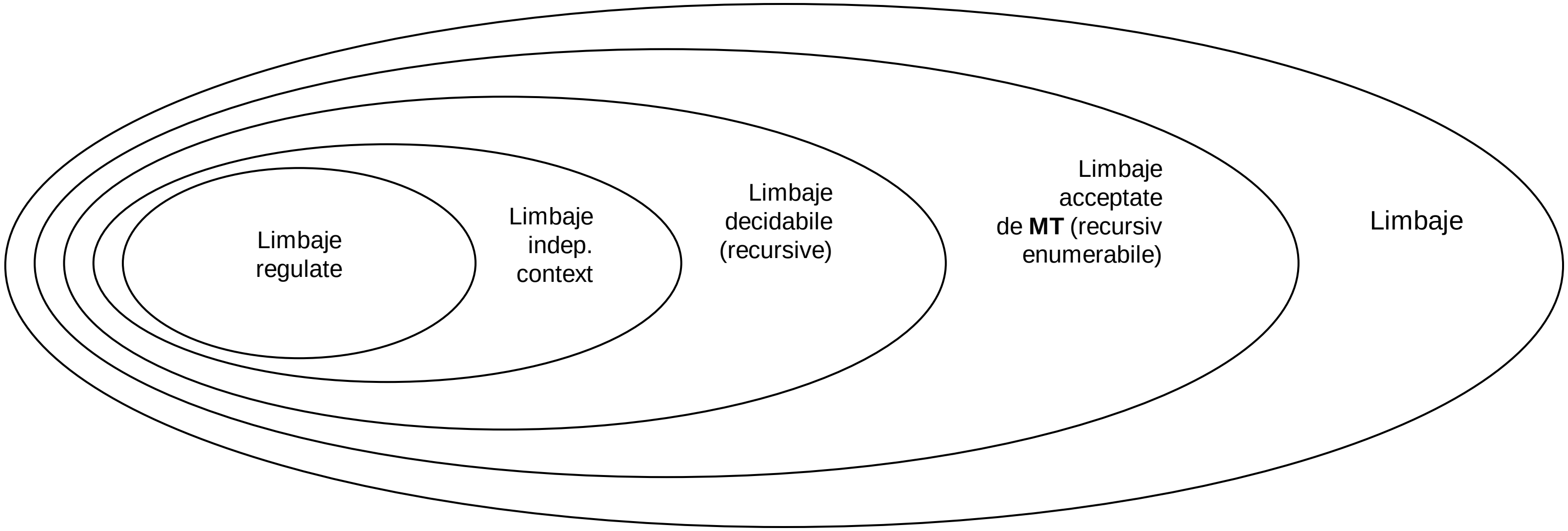
demonstraţie

Ppa: limbajul $\overline{ACC}_{MT}$ este Turing-acceptat

Am dem ca limbajul ACC_{MT} este Turing-acceptat

$\Downarrow$

ACC_{MT} este decidabil (cf. Teorema 2) $\Rightarrow$ contradicţie cu Teorema 1.



Problema opririi

1. Metoda diagonalizarii
2. Nedecidabilitatea unui limbaj dat
3. Un limbaj care nu este Turing-acceptat