10.10.2019

Examen scris cu 2 sub. - teorie
                        - ex.
- se face media dintre cele 2
- bonus de la seminar (max. 0,5)

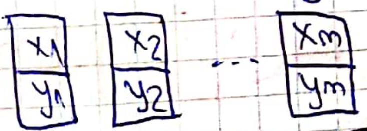
Cuprinsul cursului:

Calculabilitate : programe standard
funcții recursive
mașini Turing
} ≡

nedecidabilitate (problema corespondenței lui Post)
=PCP

Complexitate : clase de complexitate
relații între clasele de complexitate
P = NP ?, (N)P = clasa tuturor problemelor ce se pot rezolva în timp polinomial (nedeterminist)

PCP: Pres. că avem niște cuburi a căror suprafață se suprapune; pe fețele lor sunt string-uri (x-ii și y-ii)



, m dat.

(∃) K>0 a.î.

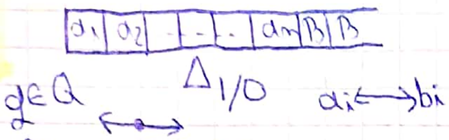
=> x=y.

input: m, $X=(x_1,\dots,x_m)$, $Y=(y_1,\dots,y_m)$

output: există K a.î. $x_{i1}x_{i2}\dots x_{ik} = y_{i1}y_{i2}\dots y_{ik}$?

nu va exista niciodată un algoritm care să rezolve problema.

MAȘINI TURING



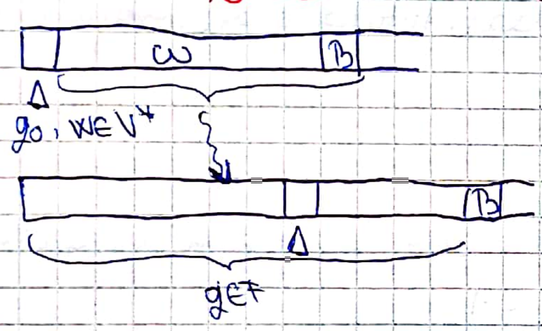
O mașină Turing poate fi folosită cu 2 scopuri:
- acceptat limbaje formale
- calculat funcții $f: \mathbb{N} \xrightarrow{\circ} \mathbb{N}$

fct. nu e complet definită

$\delta \xrightarrow{\circ} \mathbb{N}^m$
$v = (v_1, \dots, v_m) \longleftrightarrow m$

Obs.: $A_1, A_2, \dots$, $f: \mathbb{N} \xrightarrow{\circ} \mathbb{N}$.

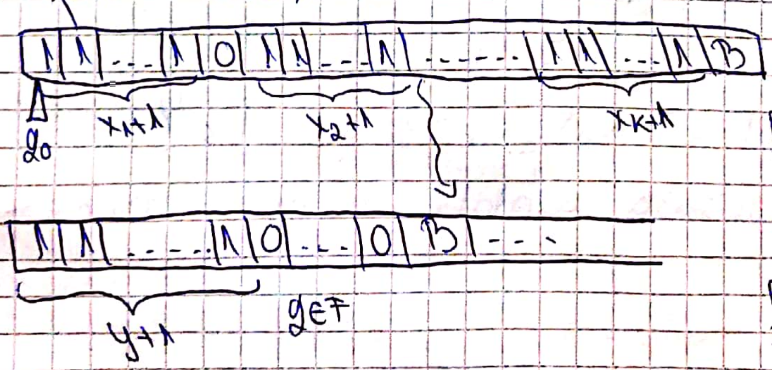
ACCEPTOR DE LIMBAJE



CALCULATOR DE FUNCȚII NATURALE

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

$f(x_1, \dots, x_k)$



$f(x_1, \dots, x_k) = y$

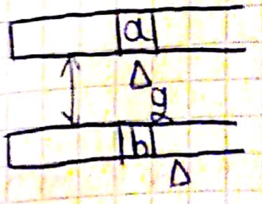
$f(x_1, \dots, x_k) = \uparrow$ mașina nu se oprește

Def.: O mașină Turing este $M = (Q, \overset{\Sigma}{V}, \overset{\Gamma}{U}, \delta, q_0, B, F)$

$V \subset U$; $B \in U \setminus V$; $F \subseteq Q$; $\delta: (Q \setminus F) \times U \longrightarrow \mathcal{P}(Q \times U \times \{R; L\})$ $2^{(\dots)}$

$(p; b; R) \in \delta(q; a)$

stare

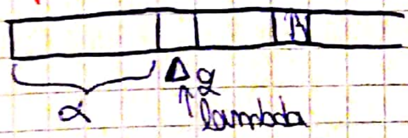


În orice moment dat (∃) un nr. finit de celule care conțin un simbol $\neq B$; toate celelalte celule sunt B.

Obs.: Dacă $(s, b, x) \in \delta(q; a)$, atunci:
- dacă $a \neq B$, atunci $b \neq B$
- dacă $a = B$, atunci $b \in U$

Descriere instantanee/configurație a benzii: $\alpha q \beta$

$\alpha, \beta \in (U \setminus \{B\})^*$, $q \in Q$



(i) $\alpha q a \beta \vdash \alpha b s \beta$, dacă $(s, b, R) \in \delta(q, a)$

(ii) $\alpha q \vdash \alpha b s$, dacă $(s, b, R) \in \delta(q, B)$

(iii) $\alpha c q a \beta \vdash \alpha b s c b \beta$, dacă $(s, b, L) \in \delta(q, a)$

(iv) $\alpha c q \vdash \alpha s c b$, dacă $(s, b, L) \in \delta(q, B)$.

a, b, c sunt definite conform observației

$\vdash^*$ $L(M) = \{ w \in V^* / q_0 w \vdash^* \alpha q \beta,\ q \in F \}$.

17.10.2019

2

$M = (Q, V, U, \delta, q_0, B, F)$

M este deterministă

$card(\delta(q; a)) \leq 1$, $(\forall) q \in Q \setminus F$, $(\forall) a \in U$

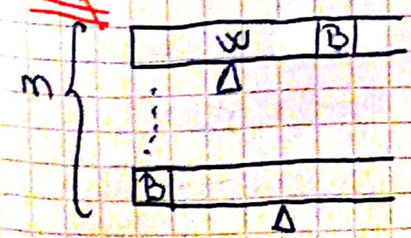
$$\begin{array}{ccc} MTD & \longleftarrow & mMT \\ & \searrow \quad \nearrow & \\ & MTN & \end{array}$$

MTD = mașina turing deterministă

mMT = mașina turing cu m benzi

MTN = mașina turing nedeterministă

Def.: M este o mT cu m benzi dacă $M = (m, Q, V, U, \delta, q_0, B, F)$.



$\delta : (Q \setminus F) \times U^m \to \mathcal{P}(Q \times (U \setminus \{B\})^m \times \{L, R\}^m)$

$(s, b_1, \dots, b_m, x_1, \dots, x_m) \in \delta(q, a_1, \dots, a_m)$.

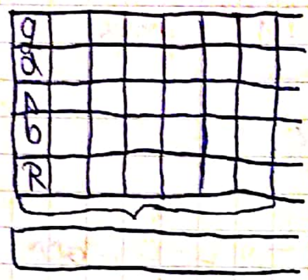
TEOREMĂ:

Pentru orice mașină Turing nedeterministă M (cu o bandă) există o mașină Turing deterministă M' cu 3 benzi a.î. $L(M)=L(M')$.

Dem.:

Fie $M=(Q,V,U,\delta,q_0,B,F)$ o mașină Turing nedeterministă.

M: w ⇝ M': w



q' - starea curentă a lui M'

(i) necorespondență

(ii) corespondență

$E=\{(q,a,s,b,x) / q\in Q\setminus F, s\in Q, a\in U, b\in U\setminus\{B\}, x\in\{R,L\}\}$ } mulțime finită

$(s,b,x)\in\delta(q,a)$

$(E,\leq)$ → se poate defini o rel. de ordine.

Procedura pt. M':

Pas 1) copiază intrarea (w) pe banda 3.

Pas 2) capul de citire/scriere de pe prima bandă revine pe celula inițială

Pas 3) se generează următorul element al conținutului benzii 2 în ordine lexicografică pe E^*.

Pas 4) încercăm să simulăm pe M' „calculul" lui M aflat pe banda 2. $(q,a) \langle {(s,b,R) \atop (s,b,L)}$

a) calculul de pe banda 2 este un calcul invalid.

- restaurăm w pe prima bandă, copiindu-l de pe banda 3.

GOTO Pas 3

b) calculul este valid

- dacă s-a ajuns la o stare finală, atunci M' acceptă și se oprește
- dacă nu s-a ajuns la o stare finală, restaurăm w pe banda 1

GOTO Pas 4.

Dacă M nu se oprește pe intrarea w, nici M' nu se oprește.
Dacă M se oprește întotdeauna pe intrarea w, dar niciodată pe starea finală, M' nu se oprește.

$L(M) = L(M')$, M' = deterministă.

Q.E.D.

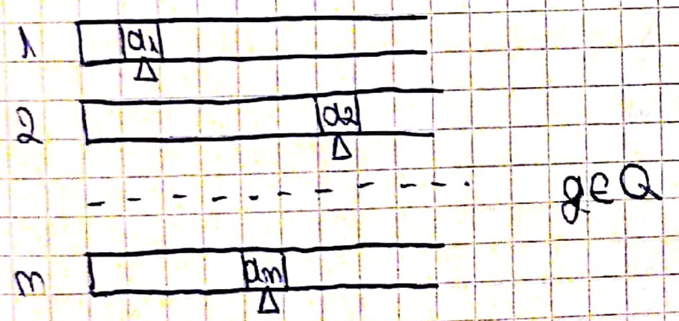
TEOREMĂ:

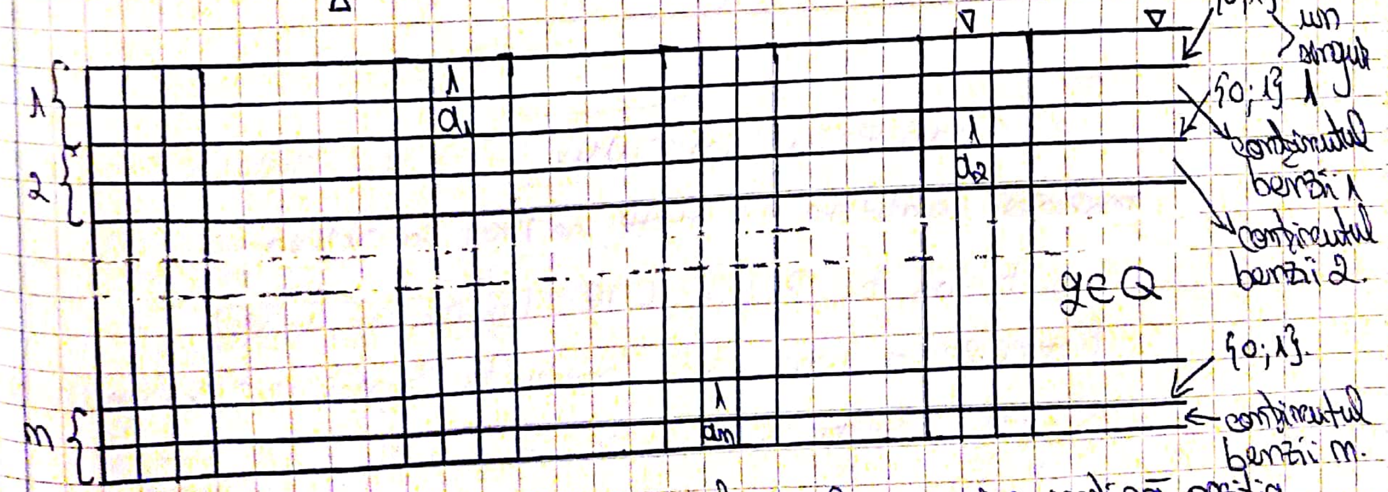
Pentru orice mașină Turing M cu m benzi există o mașină Turing M' cu o bandă a.î. $L(M) = L(M')$. În plus, M' este deterministă dacă M este deterministă.

24.10.2019.

Dem.:

Fie $M = (m, Q, V, U, \delta, q_0, B, F)$ o mașină Turing cu m benzi.





Poziția simbolului λ pe track-urile impare indică poziția capului R/W pe banda asociată.

Fie $\delta(s, b_1, b_2, \dots, b_m, x_1, \dots, x_m) \in \delta(q; a_1; \dots; a_m)$.

Cum funcționează M'? Simulează tranziția.

Pas 1) Se poziționează capul R/W pe prima poziție și memorează starea q.

Pas 2) Scanează întreaga bandă și memorează în stare simbolurile citite de M și pozițiile lor.

$< q, \underbrace{0, \dots, 0}_{m}, \underbrace{0, \dots, 0}_{m} >$

pune a_i unde există λ

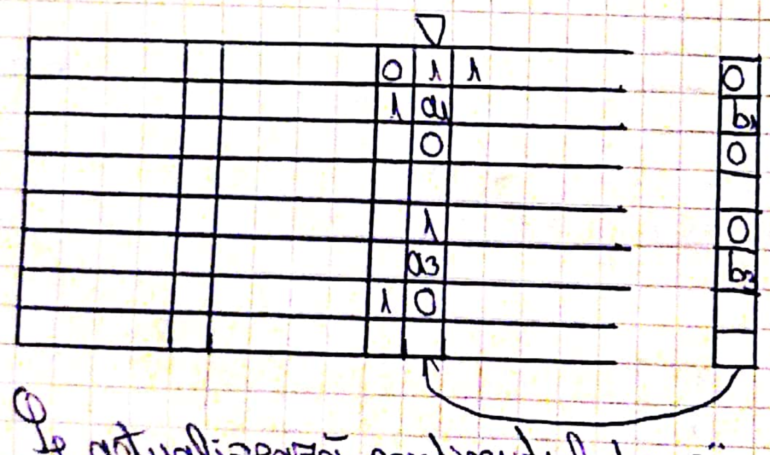
$< q, a_1, \dots, 0, \lambda, \dots, 0 >$

$< q, a_1, \dots, a_m, \lambda, \dots, \lambda >$

Dacă $card(U) = m$, atunci numărul total de posibilități de vectori cu o literă din U sau cu 0 este $(m+1)^m$, care este mare, dar finit.

$(\underbrace{0, \dots, 0}_{m}) \to 0, \ \to \in U$

Pas 3)



Se actualizează conținutul benzii a.î. după actualizare banda respectă condițiile enunțate; se trece în starea s.

$(s, b_1, b_2, b_3, b_4, R, L, L, L) \in \delta(q, a_1, a_2, a_3, a_4)$

Actualizarea unei celule necesită 3 pași: citire/modificare, deplasare dreapta și revenire.

Mulțimea stărilor finale pt. M' este F.

$L(M') = L(M)$; M' det. dacă M det.

Scanned by CamScanner

Funcții recursive $f: \mathbb{N}^k \xrightarrow{\theta} \mathbb{N}$.

Funcții elementare (primitive):

- succesor: $\mathbb{N} \to \mathbb{N}$, $succ(x) = x+1$
- constante: $C_{k_i}: \mathbb{N}^k \to \mathbb{N}$, $C(x_1; x_k) = m$
- proiecții: $\pi_i: \mathbb{N}^k \to \mathbb{N}$, $\pi_i(x_1; x_k) = x_i$

Mașina Turing care calculează funcții elementare

Operații:

1) Compunerea funcțională

$f: \mathbb{N}^k \xrightarrow{\theta} \mathbb{N}$ este definită prin compunerea funcțională din $h: \mathbb{N}^m \xrightarrow{\theta} \mathbb{N}$

$g_i: \mathbb{N}^k \xrightarrow{\theta} \mathbb{N}, i = \overline{1,m}$ dacă

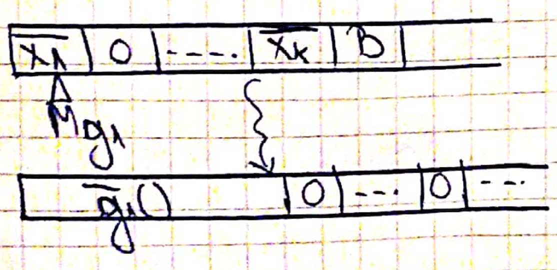
$f(x_1; \dots; x_k) = h(g_1(x_1; \dots; x_k), g_2(x_1; \dots; x_k); \dots; g_m(x_1; \dots; x_k))$

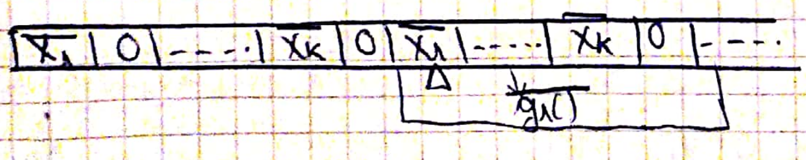
Mașina Turing care calculează funcții compuse

$M_h \to$ calculează h

$M_{g_i} \to$ calculează $g_i, i = \overline{1,m}$

$M_f \xrightarrow{?}$ calculează f.





FUNCȚII RECURSIVE

31.10.2019.

4

- Funcții elementare
- Operații:

1) Compunerea funcțională

L1) Dacă $g_i : \mathbb{N}^m \xrightarrow{\theta} \mathbb{N}, i=\overline{1,m}$ sunt Turing calculabile,
$h : \mathbb{N}^m \xrightarrow{\theta} \mathbb{N}$

atunci f definită prin compunerea funcțională din h și g_i este Turing calculabilă.

2) Recurența primitivă

funcția $f : \mathbb{N}^{k+1} \xrightarrow{\theta} \mathbb{N}$ se obține prin recurență primitivă din $h : \mathbb{N}^{k+2} \xrightarrow{\theta} \mathbb{N}$ și $g : \mathbb{N}^{k} \xrightarrow{\theta} \mathbb{N}$ dacă

(1) $f(x_1, \dots, x_k, 0) = g(x_1, \dots, x_k)$

(2) $f(x_1, \dots, x_k, t+1) = h(x_1, \dots, x_k, t, f(x_1, \dots, x_k, t))$

Ex: fac(n)?

$k=0 \quad fac(0)=1=C_1$

$fac(n+1) = (n+1) \cdot fac(n) = h(n, fac(n)) = h(succ(n), fac(n))$

$h(a,b) = a*b$

Pres. că g și h sunt Turing calculabile. Este f Turing calculabilă?

$\overline{x_1}$	0	$\overline{x_2}$	0	...	$\overline{x_k}$	0	$\overline{x_{k+1}}$	...

P1) Dacă $x_{k+1}=0$, atunci transformă ultimul 1 în 0. Aplică M_g la banda obținută.

P2) Dacă $x_{k+1} \neq 0$

$\overline{x_1}$	0	...	$\overline{x_k}$	0	$\overline{x_{k+1}}$	0	$\overline{x_m}$	0	...	$\overline{x_k}$	0	$\overline{x_{k+1}-1}$	0	$\overline{x_1}$	0	..	$\overline{x_{k+1}-2}$	0	...		

M_h

...	$\overline{x_1}$	0	...	$\overline{0}$	0	$\overline{x_1}$	...	$\overline{x_k}$

M_h $\quad$ M_g

$f(x_1, x_k, 0)$

$f(x_1, \dots, x_k, 1)$

L2) Dacă g și h sunt Turing calculabile, atunci f este Turing calculabilă

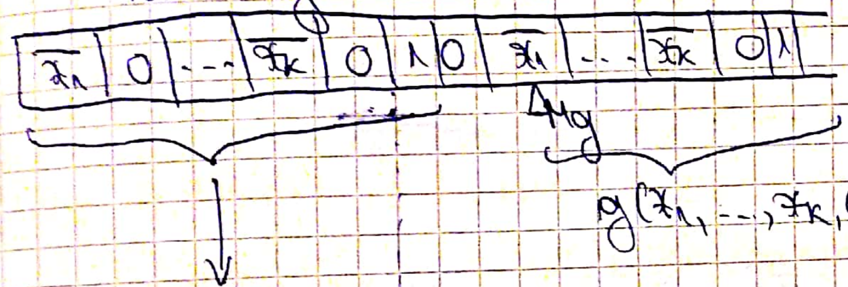
3) Minimizare mărginită

$g: \mathbb{N}^{k+1} \longrightarrow \mathbb{N}$

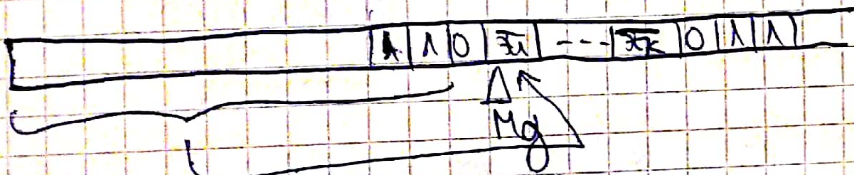
$f: \mathbb{N}^{k} \overset{o}{\longrightarrow} \mathbb{N}$ se obține prin minimizare nemărginită din g dacă

$f(x_1, \dots, x_k) = \min_y \{ g(x_1, \dots, x_k, y) = 0 \}$

Pres. că g este T calculabilă. Este f Turing calculabilă?



$g(x_1, \dots, x_k, 0) \neq 0$



L3) Dacă g este Turing calculabilă, atunci f este Turing calculabilă.

Def.: O funcție este (parțial) recursivă dacă se obține din funcții elementare prin aplicarea de un număr finit de ori a operațiilor ①–③

TEOREMĂ:

Orice funcție (parțial) recursivă este Turing calculabilă.

$f(x_1, x_2) = x_1 + x_2$

$f(x_1, 0) = x_1 = \pi_1(x_1)$

$f(x_1, x_2+1) = f(x_1, x_2) + 1 = succ(f(x_1, x_2)).$

$succ(\pi_3(x_1, x_2, f(x_1, x_2)))$

PROGRAME STANDARD

Limbajul:
- variabile:
  - intrare: $x_1, x_2, \ldots$ $\in \mathbb{N}$
  - ieșire: y 0
  - lucru: $z_1, z_2, \ldots$ 0
- etichete: $E, A_1, A_2, \ldots$
- instrucțiuni
  - $V \leftarrow V$, V este variabilă
  - $V \leftarrow V+1$, V este variabilă
  - $V \leftarrow V \dot{-} 1$
  - IF $V \neq 0$ GOTO L — V - variabilă, L - etichetă

Dacă val. lui V este 0, se trece la instrucțiunea următoare

Dacă val. lui V nu este 0 se face salt la prima instrucțiunea etichetată cu L.

Obs.: cu PS calculăm funcții

$f: \mathbb{N}^k \longrightarrow \mathbb{N}$ $\quad f(x_1, \ldots, x_k)$

Dacă $f(x_1, \ldots, x_k) = \underset{\text{e definit}}{\downarrow}$, atunci programul se termină și

$y = f(x_1, \ldots, x_m)$. $\quad x_1 = x_1$

$A_2: x_{s+} \leftarrow x_{s+1}$ $\quad x_2 = x_2$ ...

$x_k = x_k$

Program standard

Secvență finită de instrucțiuni

Terminare

1) Nu mai sunt instrucțiuni de executat
2) Salt la eticheta E
3) Salt la o etichetă ce nu există

altfel programul nu se oprește.

$f(x_1) = x_1$

$y \leftarrow x_1$

$y \leftarrow x_1$
IF $x_1 \neq 0$ GOTO A_1
$z_1 \leftarrow z_1 + 1$
IF $z_1 \neq 0$ GOTO E
A_1: $x_1 \leftarrow x_1 - 1$ $\quad z_2 \leftarrow z_2 + 1$
$y \leftarrow y + 1$
IF $x_1 \neq 0$ GOTO A_1 $\quad A_2$
A_2: $z_2 \leftarrow z_2 - 1$
$x_1 \leftarrow x_1 + 1$
IF $z_2 \neq 0$ GOTO A_2
GOTO L:
$z_j \leftarrow z_j + 1$
IF $z_j \neq 0$ GOTO L

07.11.2019

5

$V \leftarrow V'$

Pres. că $f: \mathbb{N}^k \to \mathbb{N}$ este calculabilă cu programe standard

$V \leftarrow f(x_1, \ldots, x_k)$.

P [.... care calculează fct. f.
....
....
....
....
....

$A_i: V \leftarrow Y$

Obs.: 1) variabila V nu apare în instrucțiunile lui P.
2) variabilele și etichetele din programul $V \leftarrow Y$ sunt diferite de variabilele și etichetele din programul P.
3) programul P se poate presupune (fără a pierde din generalitate) că se termină doar în una din următoarele situații:
- salt la eticheta E
- terminarea instrucțiunilor.

(3) Programul P (înăuntru) se modifică saltul la eticheta E cu salt la o nouă etichetă care va fi eticheta instrucțiunii $V \leftarrow Y$.

A1: Orice funcție elementară este calculabilă cu programe standard

Dem.: succ: $Y \leftarrow X_1$
$Y \leftarrow Y+1$

$C_m^k(x_1, \ldots, x_k) = m$

m ori $\begin{cases} Y \leftarrow Y+1 \\ \ldots \\ Y \leftarrow Y+1 \end{cases}$

$z \leftarrow z+1$

$\pi_i(x_1, \ldots, x_k) = x_i$

$Y \leftarrow X_i$

Scanned by CamScanner

Operația de compunere funcțională:

$$f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k))$$

A2: Dacă o funcție este obținută prin compunere funcțională din funcții calculabile cu programe standard, atunci este calculabilă cu programe standard.

$Z_{i_1} \leftarrow g_1(x_1, \ldots, x_k)$

$Z_{i_2} \leftarrow g_2(x_1, \ldots, x_k)$

$\ldots$

$Z_{i_m} \leftarrow g_m(x_1, \ldots, x_k)$

$Y \leftarrow h(Z_{i_1}, \ldots, Z_{i_m})$

cu observațiile 1), 2), 3)

Obs.: 4) La fiecare program pt. macroinstrucțiuni se face o copie a variabilelor de intrare $x_1, \ldots, x_k$

Recurența primitivă:

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

$$f(x_1, \ldots, x_k, t+1) = h(x_1, \ldots, x_k, t, f(x_1, \ldots, x_k, t))$$

A3.: O funcție obținută prin recurența primitivă din funcții calculabile cu programe standard este calculabilă cu programe standard

$Z_{i_1} \leftarrow g(x_1, \ldots, x_k)$

A_{i_2}: IF $x_{k+1} = 0$ GOTO A_{i_1}

$Z_{i_3} \leftarrow h(x_1, \ldots, x_k, Z_{i_2}, Z_{i_1})$

$x_{k+1} \leftarrow x_{k+1} \dot{-} 1$

$Z_{i_2} \leftarrow Z_{i_2} + 1$

$Z_{i_3} \leftarrow Z_{i_3}$

GOTO A_{i_2}

A_{i_1}: $Y \leftarrow Z_{i_1}$

Minimizare nemărginită

$$f(x_1, \ldots, x_k) = \min_z [g(x_1, \ldots, x_k, z) = 0]$$

A4.: O funcție obținută prin minimizare nemărginită dintr-o funcție calculabilă cu PS este calculabilă cu PS.

A_{i2}: $Z_{i1} \leftarrow g(X_1, \dots, X_k, Z_{i2})$

IF $Z_{i1} = 0$ GOTO A_{i1}

$Z_{i2} \leftarrow Z_{i2} + 1$

GOTO A_{i2}.

A_{i1}: $Y \leftarrow Z_{i2}$.

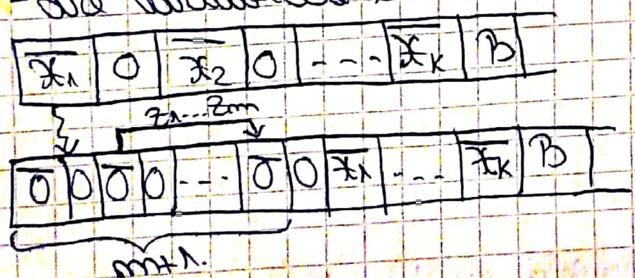
TEOREMĂ:
Orice funcție (parțial) recursivă este (parțial) calculabilă cu PS.

Dem.: $A_1 + A_2 + A_3 + A_4$.

$$\begin{tikzcd}
FR \arrow[rr] \arrow[dr] & & FTC \arrow[ll, bend right, dashed] \\
& FPS \arrow[ur] &
\end{tikzcd}$$

TEOREMĂ:
Orice funcție calculabilă cu PS este Turing calculabilă.

Dem.: Fie $f: \mathbb{N}^k \to \mathbb{N}$ calculabilă cu PS.
Există P care calculează f:
- are instrucțiunile $I_1, \dots, I_n$
- are variabilele locale $Z_1, \dots, Z_m$.



$\langle I_1 \rangle$ starea curentă

Inductiv, P urmează să execute instrucțiunea I_j. Mașina va fi în starea $\langle I_j \rangle$ și banda va conține valorile tuturor variabilelor din P.

P execută I_j.

Ce face M:

I_j este $V \leftarrow V$: M schimbă starea în $\langle I_{j+1} \rangle$

I_j este $V \leftarrow V+1$: M adaugă un 1 la segmentul de bandă corespunzător lui V și trece în $\langle I_{j+1} \rangle$.

I_j este $V \leftarrow V \dot{-} 1$: găsește segmentul corespunzător lui V. Dacă acest segment are doar un 1, trece în $\langle I_{j+1} \rangle$. Altfel, reduce nr. de 1, completează banda, trece în $\langle I_{j+1} \rangle$

I_j este IF $V \neq 0$ GOTO L: găsește segmentul corespunzător lui V. Dacă acesta are doar un 1, trece în $\langle I_{j+1} \rangle$.

Dacă are mai mulți de 1, trece în $\langle I_e \rangle$, unde

$$e = \begin{cases} \min\{r \mid I_r \text{ are eticheta } L\}. \\ m+1, \text{ dacă } L = E \text{ sau eticheta } L \text{ nu există.} \end{cases}$$

În starea $\langle I_{m+1} \rangle$ se păstrează primul segment de 1 (segment asociat lui Y) și restul se face 0.

21.11.2019.

6

$$\begin{tikzcd} FR \arrow[rr, shift left] \arrow[dr] & & MT \arrow[ll, shift left, "\#"] \\ & PS \arrow[ur] & \end{tikzcd}$$

TEOREMĂ:

Orice funcție calculabilă cu MT este funcție recursivă.

1) $f(x,y) = x \cdot y$

2) $f(x) = x!$

3) $p(x) = \begin{cases} x-1, & \text{f. } x \neq 0 \\ 0, & \text{altfel} \end{cases}$

4) $x \dot{-} y = \begin{cases} x-y, & \text{dacă } x \geq y \\ 0, & \text{altfel} \end{cases}$

5) $|x-y|$

6) $\alpha(x)=\begin{cases}1, & x=0\\ 0, & x\neq 0\end{cases}$

7) $x=y$

8) $x<y$

9) x/y

10) $Prime(x)=\begin{cases}1, & x \text{ este prim}\\ 0, & \text{altfel}\end{cases}$

11) $[x/y]$

12) $R(x,y)=x \bmod y$.

13) $p_m(p(m))$ = al m-lea nr. prim

14) $\langle x,y\rangle=2^x(2y+1)-1$

15) $l(z)=x$ a.î. $(\exists)\, y$ $\langle x,y\rangle=z$.

16) $[a_1,\dots,a_m]=p_1^{a_1}\dots p_m^{a_m}$

17) $(x)_i=b$ a.î. $x=[a_1,\dots,a_{i-1},b,a_{i+1},\dots,a_m]$.

18) $L_t(x)=m$ a.î. $x=[a_1,\dots,a_m]$.

$L_t(84)=4$.

13) $2^x(2y+1)=14 \begin{cases} x=1\\ y=3\end{cases}$

$r(z)=y$ a.î. $(\exists)\, x$ cu $\langle x,y\rangle=z$.

17) $x=84$

$(x)_3$

$84=2^2\cdot 3\cdot 7$

$84=[2;1;0;1]$

$(84)_3=0$

Dem. (TEOREMA).

Fie M o MT deterministă, $M=(Q,V,U,\delta,q_0,B,F)$.

Presupunem pt. toate MT că toate stările lor sunt în urm. fel.: $q_0,q_1,\dots$ (enumerare a tuturor stărilor posibile); $s_0,s_1,\dots$ (enumerare a tuturor simbolurilor posibile).

Configurația unei MT: $\langle a, \langle b, c\rangle\rangle$

- a = poziția stării curente în enumerarea stărilor
- b = poziția capului R/W pe bandă
- c = Gödelizarea conținutului benzii în care fiecare simbol este reprezentat prin poziția sa în enumerarea simbolului

Ex.:

1	0	a	b	b	1	B	...

$s_1\ s_0\ s_2 \qquad s_3$

$c = [1, 0, 2, 0, 0, 1, 3] = 2 \cdot 5^2 \cdot 13 \cdot 17^3$.

Presp. că M calculează $f: \mathbb{N}^k \to \mathbb{N}$, $f(x_1, \dots, x_m)$.

$C_M(\bar{x}, m)$ = nr. atașat configurației mașinii M pe intrarea $x_1, \dots, x_k$ la pasul m.

$\bar{x} = (x_1, \dots, x_m)$.

$C_M(\bar{x}, 0) = \langle 0, \langle 0, p_1 \cdot p_2 \cdot \ldots \cdot p_{x_1+1} \cdot p_{x_1+3} \cdot \ldots \cdot p_{x_1+x_2+3}\rangle\rangle$

$\bar{x}_1$	0	$\bar{x}_2$	0	...	$\bar{x}_m$	B

1111 0 111 0 1 0

$x_1 = 3$
$x_2 = 2$
$x_3 = 0$

$p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_6 \cdot p_7 \cdot p_8 \cdot p_{10}$.

1) C_M este recursivă

Dacă $C_M(\bar{x}, m) = C_M(\bar{x}, m+1)$, atunci $C_M(\bar{x}, m+p) = C_M(\bar{x})$, $(\forall)\ p \geq 0$.

Definim:

$h_1(z) = \begin{cases} \text{nr. stării în care trece mașina } M \text{ din configurația codificată cu } z, \text{ dacă } z \text{ este o codificare VALIDĂ} \\ \uparrow \text{ (nedefinit), altfel} \end{cases}$

$h_2(z) = \begin{cases} \text{poziția în care ajunge capul R/W al lui } M \text{ din configurația codificată cu } z, \text{ dacă } z \text{ este o codificare validă} \\ \uparrow \text{ (nedefinită), altfel} \end{cases}$

$h_3(z) = \begin{cases} \text{nr. atașat conținutului benzii în care ajunge mașina } M \text{ din config. codificată cu } z, \text{ dacă } z \text{ este o codificare validă} \\ \uparrow \text{ (nedefinită), altfel.} \end{cases}$

$C_M(x, n+1) = \langle h_1(C_M(x,n)), \langle h_2(C_M(x,n)), h_3(C_M(x,n))\rangle\rangle$

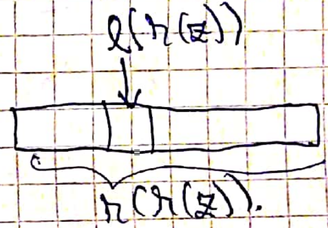
2) h_1, h_2, h_3 sunt recursive.

$g_1(a,b) = \begin{cases} \text{nr. stării în care trece M citind simbolul de pe poz. b în} \\ \text{enumerarea simbolurilor aflată în starea numerotată a.} \\ \uparrow, \text{ altfel.} \end{cases}$

nr. unei stări; indicele simbolului

$g_2(a,b) = \begin{cases} 0, \text{ dacă M din starea cu nr. a și citind simb. cu nr. b se} \\ \text{deplasează la stânga.} \\ 2, \text{ dacă se deplasează la dreapta.} \end{cases}$

$g_3(a,b) = \begin{cases} \text{nr. simbolului scris de M peste simbolul cu nr. b. în} \\ \text{starea cu nr. a} \\ \uparrow, \text{ altfel.} \end{cases}$



g_1, g_2, g_3 sunt recursive :

$h_1(z) = g_1(l(z), (r(r(z)))_{l(r(z))})$

$t \geq 0.$

$h_2(z) = g_2(l(z), r(r(z))_{l(r(z))} + l(r(z)) \dot{-} 1)$

$h_3(z) = r(r(z)) \Big/ p_{l(r(z))}^{(r(r(z)))_{l(z)}} * p_{l(r(z))}^{g_3(l(z), (r(r(z)))_{l(z)})}.$

h_1, h_2, h_3 recursive $\Rightarrow C_M$ recursivă.

$nr_M(x)$ = nr. de pași după care M se oprește pe intrarea x.

$nr_M(x) = \min_n \{C_M(x,n) = C_M(x, n+1)\} \Rightarrow$ RECURSIVĂ

$f(x_1, \ldots, x_k) = L_1(\underbrace{r(r(C_M(x, nr_M(x))))}_{\text{conținutul benzii când se oprește.}}) - 1.$

28. 11. 2019.

CODIFICAREA PROGRAMELOR STANDARD.

Enumerăm: - etichete: $E, A_1, A_2, \dots$

- variabile: $y, x_1, x_2, z_1, z_2, \dots$

Fie I o instrucțiune.

$\#(I)$ = nr. atașat instrucțiunii I

$= \langle a, \langle b, c\rangle\rangle$.

$$a = \begin{cases} 0, & \text{dacă } I \text{ nu este etichetată} \\ \#(L), & \text{dacă } I \text{ este etichetată cu eticheta } L. \end{cases}$$

$b = \#(V)^{-1}$, V este variabila din I.

$$c = \begin{cases} 0, & \text{dacă } I: V \leftarrow V \\ 1, & \text{dacă } I: V \leftarrow V+1 \\ 2, & \text{dacă } I: V \leftarrow V+1. \\ \#(L)+2, & \text{dacă } I: \text{IF } V \neq 0 \text{ GOTO } L \end{cases}$$

$$\langle a, b\rangle = 2^a(2b+1)-1.$$

Ex: 1) $A_2: Z_1 \leftarrow Z_1+1$

$\langle 3, \langle 2, 1\rangle\rangle = \langle 3, 2^2(2+1)-1\rangle = \langle 3, 11\rangle = 183.$

2) $\#(I) = 264.$

$\langle a, \langle b, c\rangle\rangle = 264$

$2^a(2\langle b, c\rangle + 1) - 1 = 264$

$2^a(2\langle b, c\rangle + 1) = 265 \Rightarrow a = 0.$

$2\langle b, c\rangle = 264 \Rightarrow \langle b, c\rangle = 132 \Rightarrow 2^b(2c+1) = 133.$

$b = 0 \Rightarrow 2c = 132 \Rightarrow c = 66.$

I: IF $y \neq 0$ GOTO A_{63}.

3) $\#(I) = 0$ $\quad y \leftarrow y$

$\#(P)$ P are instrucțiunile $I_1, I_2, \dots, I_m$

$$\#(P) = [\#(I_1), \dots, \#(I_m)] - 1.$$

Ex: $\#(P) = 66$ $66+1 = 67$.

$67 = 2^0 \cdot 3^0 \cdot 5^0 \cdot 7^0 \cdot 11^0 \cdot 13^0 \cdot 17^0 \cdot 19^0 \cdot 23^0 \cdot 29^0 \cdot 31^0 \cdot 37^0 \cdot 41^0 \cdot 43^0 \cdot 47^0 \cdot 53^0 \cdot$
$\cdot 61^0 \cdot 67^1 \Rightarrow 19$ instrucțiuni

$\varphi_P^{(m)}$ = funcția de m variabile calculată de programul P.

$\phi^{(m)}(x_1, \dots, x_m, z)$ = funcția universală de m variabile
$= \varphi_P^{(m)}(x_1, \dots, x_m)$ cu z = nr. atașat programului P $\#(P)$.

TEOREMĂ: (Universalitate)

Funcția universală de m variabile este calculabilă cu programe standard.

Dem.:

$k \leftarrow 1$

$L \leftarrow L_t(x_{m+1}+1)$ $\quad M = \prod_{i=1}^{m} p_{2i}^{x_i}$

$\downarrow$ codul programului

[C]: IF $(k=0) \vee (k > L)$ GOTO F

$I \leftarrow (x_{m+1}+1)_k$

$V \leftarrow l(r(I)) + 1$ (poz. variabilei)

$T \leftarrow r(r(I))$

IF $T=0$ GOTO N

IF $T=1$ GOTO A

IF $T=2$ GOTO S

IF $p_V \nmid M$ GOTO N

$k \leftarrow \begin{cases} \min_R \{ l((x_{m+1}+1)_R) = T-2 \}, R \text{ există și } l((x_{m+1}+1)_R)+1 \\ 0, \text{ dacă nu există } R \text{ sau } l((x_{m+1}+1)_R) = 1 \end{cases}$

GOTO C.

{A}: $M \leftarrow M * p_v$
GOTO N

{S}: IF $p_v \nmid M$
GOTO N
$M \leftarrow M / R$

{N}: $k \leftarrow k+1$
GOTO C

{F}: $y \leftarrow (M)_1$

$$HALT(x,y) = \begin{cases} 1, & \text{programul cu nr. } y \text{ se termină pe intrarea } x \\ 0, & \text{altfel} \end{cases}$$

$= \varphi_P^{(1)}(x,y) = \downarrow$ cu $\#(P) = y$.

TEOREMĂ: (Problema opririi)
Predicatul HALT nu este calculabil cu PS.

Dem.:
Pres. red. abs. că $(\exists)$ un program P care calculează HALT.
A: IF HALT(x,y) GOTO A $\longrightarrow P'$
P' calculează $\neg HALT(x,y)$, $(\forall) x, y$.
Fie $z = \#(P')$ $HALT(x,z) \equiv \neg HALT(x,y)$, $(\forall) x, y$.
$y = z$ ↯

5.12.2019.

EXISTENȚA PROGRAMULUI UNIVERSAL
PROBLEMA OPRIRII UNUI PS.

Problema de decizie:
$P: D \longrightarrow \{0, 1\}$
$L_P = \{ \underbrace{\langle a_1, \ldots, a_m \rangle}_{\text{codificare a instanțelor } a_1, \ldots, a_m} / P(a_1, \ldots, a_m) \}$

P este decidabilă dacă există o mT M a.î.:
- se oprește pe fiecare intrare } algoritm
- este deterministă
- $L(M) = L_P$

A - mulț. recursivă χ_A este decidabilă

A - recursiv enumerabilă dacă χ_A este calculabilă cu mT.
se oprește dacă face parte din A — semialgoritm

A - nerecursiv enumerabilă

PROPRIETĂȚI:

1) A este recursivă $\Longrightarrow$ A este recursiv enumerabilă
2) A este recursivă $\Longleftrightarrow$ A și C_A sunt recursiv enumerabile
3) A și B recursive $\Longrightarrow$ $A \cup B$, $A \cap B$ și C_A sunt recursive
4) A și B sunt recursiv enumerabile $\Longrightarrow$ $A \cup B$ și $A \cap B$ sunt recursiv enumerabile

Dem:

1) trivial

2) "dacă" cele 2 mT se rulează alternativ
"numai dacă" A este recursivă $\Longrightarrow$ A este rec-enum. (1)
A este recursivă $\Longrightarrow$ C_A este recursivă

3) trivial

4) cele 2 mT se rulează alternativ:
- dacă cel puțin una se oprește, intrarea $\in A \cup B$
- dacă amândouă se oprese, intrarea $\in A \cap B$.

CODIFICAREA MAȘINII TURING

Codificare preliminară:

- fie $q_0, q_1, \dots$ o enumerare a tuturor stărilor posibile
- fie $s_0, s_1, \dots$ o enumerare a tuturor simbolurilor posibile.

(q, a, q', b, x)

$x \to L, R.$

$$\underbrace{\#(q_0)\$\#(q_1)\$\dots\$\#(q_m)}_{\#(Q)}\$\underbrace{\#(s_0)\$\dots\$\#(s_p)}_{\cup \ \setminus\{B\}}$$

$(\#(q)\#(a)\#(q')\#(b)\#(x))$.

$\#M \in \{(,),0,1,2,R,L\}^*$

$\#(M) \in \{0,1\}^+$

R,	L,	q_0,	s_0,	q,	s_1
10	100	10^3	10^4	10^5	

Enumerare a codificărilor tuturor mT cu intrări 0, 1.

$\hat{M}$ poziția lui $\#(M)$ în enumerare

Fie $w_1, w_2, \dots, w_n, \dots$ enumerarea tuturor mT binare.

$L_u = \{\langle M, w\rangle / w \in L(M)\} \to$ funcția universală

$L_u = L(M_u)$

M_u este recursiv enumerabilă

este recursivă?

$L_d \to$ limbaj diagonal.

AFIRMAȚIE:

$L_d = \{w_j \notin L(M), \hat{M} = j\}$ nerecursiv enumerabile

Dem.: Pres. A că există M' a.î. $L(M') = L_d$

Fie $\hat{M'} = k$

$w_k \in L_d \iff w_k \notin L(M') = L_d$.

AFIRMAȚIE:

$L_m \in$ recursiv enumerabilă \ recursiv

Dem.:

Pres. A că $L_m \in$ recursiv $\Rightarrow M_m$ se oprește

Construim M': intrare w

- găsește j a.î. $w = w_j$
- găsește M_1 a.î. $\hat{M_1} = j$
- simulează M_m pe intrarea $\langle \#(M_1), w_j \rangle$
- acceptă $w \Leftrightarrow M_m$ respinge $\langle \#(M_1), w_j \rangle$

$w = w_j \in L(M') \Leftrightarrow \langle \#(M_1), w_j \rangle \notin L_m$
$\Leftrightarrow w_j \in L(M_1) \Leftrightarrow L_d = L(M')$.

TEOREMA LUI RICE:

Orice proprietate netrivială pe familia recursiv enumerabilă este nedecidabilă.

14 10.2019.

I. Rice

Henry Gordon Rice.

① proprietate pe RE este o submulțime $\mathcal{P} \subseteq RE$. O proprietate $\mathcal{P}$ e trivială dacă $\mathcal{P} = \emptyset$ sau $\mathcal{P} = RE$.

$L_{\mathcal{P}} = \{ \#(M) / L(M) \in \mathcal{P} \}$ (codul lui M)

Proprietatea $\mathcal{P}$ este decidabilă dacă $L_{\mathcal{P}}$ este recursiv.

TEOREMA:

Orice proprietate netrivială pe RE este nedecidabilă.

Dem.: Pp. că $\mathcal{P}$ netrivială este decidabilă, $L_{\mathcal{P}} = M_{\mathcal{P}}$ — se oprește întotdeauna

Cazul I: $\emptyset \notin \mathcal{P}$.

Fie $L \in \mathcal{P}$ a.î. $L = L(M_L)$.

Fie $< \#(M), w >$ arbitrară, fixată.

Construiesc o mașină M' a.î. $(M') = \begin{cases} L, & \text{dacă } w \in L(M) \\ \emptyset, & \text{altfel.} \end{cases}$

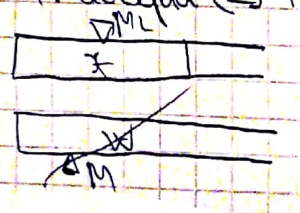
Descriere M': x

P_1: ignoră pe moment x.

P_2: verifică dacă $w \in L(M)$ prin simularea lui M pe intrarea w.

P_3: dacă M acceptă, atunci simulează M_L pe intrarea x.

P_4: acceptă $\Leftrightarrow$ M_L acceptă



Construim M_u

Descrierea M_u: input $<\#(M), w>$.

P_1: "Găsesc" M'.

P_2: Simulează $M_{\mathcal{P}}$ pe intrarea $\#(M)$.

P_3: M_u acceptă $<\#(M), w> \Leftrightarrow M_{\mathcal{P}}$ acceptă intrarea $\#(M)$.

(a) M_u se oprește întotdeauna.

(iii) $L(M_m) = ?$

$\langle \#(M), w \rangle \in L(M_m) \iff \#(M') \in L_{\mathcal{P}}$

$\iff L(M') \in \mathcal{P}$

$\iff L(M') = L$

$\iff w \in L(M)$

Caz II: $\emptyset \in \mathcal{P}$.

Fie $\overline{\mathcal{P}}$ complementul lui $\mathcal{P}$.

$\emptyset \notin \overline{\mathcal{P}}$

Întreaga dem. de dinainte funcționează și aici.

$\Rightarrow \overline{\mathcal{P}}$ nu e decidabilă

$\Rightarrow \mathcal{P}$ nu e decidabilă.

COMPLEXITATE

Complexitate
- abstractă = nu contează timp și spațiu
- descriptivă (nu o vom face) = contează să aibă cât mai puține instrucțiuni
- calculului < timp, spațiu)

Măsura timp:

Def.: model:
- se oprește pe fiecare intrare
- are un nr. de benzi infinite la ambele capete.
- capetele de citire/scriere pot staționa.

string

$Time(M, x) = \min\{t \ / \ M$ se oprește pe intrarea x în t pași$\}$

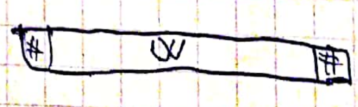
$Time(M, n) = \max\{Time(M, x) \ / \ |x| = n\}$.

$(N)(D)\ TIME_k(f(n)) = \{L \ / \ (\exists)\ M$ a.î. $L(M) = L$,
M are k benzi,
M este nedeterministă / det.
$Time(M, n) \leq f(n), (\forall)\ n \geq n_0\}$

Obs.: $f(n) \geq n+1$

Măsura spațiu:

Def.: model: MT off-line

(i) o bandă de intrare finită 

capul este de citire

(ii) un număr de benzi infinite la dr. și mărginite la stg.

capetele sunt R/W și pot staționa.

$Space(M, x, c)$ = maximul de celule folosite pe o bandă auxiliară în calculul c.

$Space(M, x) = \min\{Space(M, x, c) \mid c$ este un calcul pe intrarea $x\}$.

$Space(M,n) = \max\{Space(M,x) \mid |x| = n\}$.

$(N)(D)SPACE_k(f(n)) = \{L \mid \exists M$ a.î. $L = L(M)$,
M are k benzi,
M este det. / nedet.
$Space(M,n) \le f(n), \forall n \ge n_0\}$

Obs.: $f(n)$.

Ex.: $L = \{wcw \mid w \in \{a,b\}^*\}$.

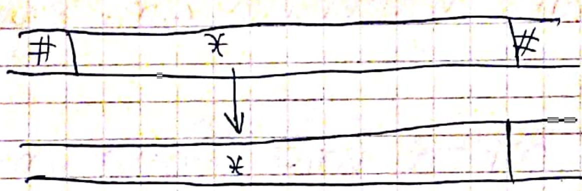
Sol. 1:

$Time(M_1, n) \le n + \frac{n^2}{2}$, $L \in NTIME_1\left(n + \frac{n^2}{2}\right)$.

$L \in DTIME_1\left(\frac{n(n+1)}{2} + \frac{n^2}{2}\right)$.

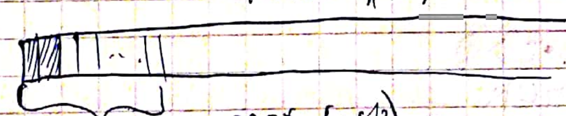
$L \in NTIME_2(n)$.

Var. 1:



$L \in (N)(D)SPACE(n+1)$.

$L \in NSPACE_1(n)$

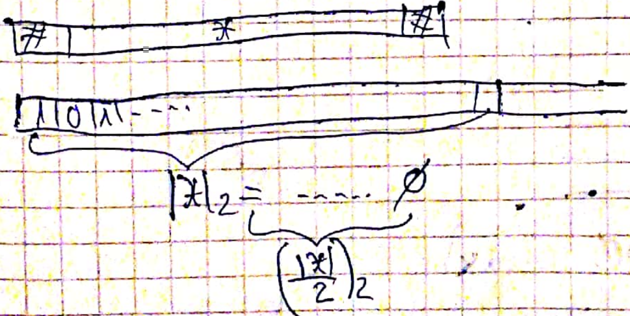


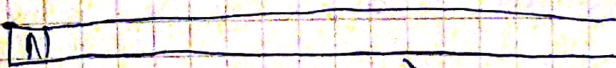
$L \in DSPACE_1(x/2)$.

$L' \in NTIME_1\left(n + \frac{n^2}{2}\right)$

$L \in DTIME_1\left(\frac{n(n+1)}{2} + \frac{n^2}{2}\right)$.

$L \in NTIME_2(n)$.





$L \in DSPACE_2(\log n)$.

12.12.2019

10

- Eliminarea constantelor
- Comprimarea benzilor

TEOREMĂ:

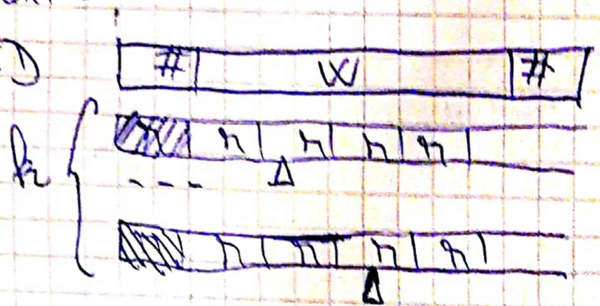
$$(N)(D)SPACE_k(f(n)) = (N)(D)SPACE_k(c\,f(n))\ (\forall)c>0.$$

Dem.:

Este suficient $(N)(D)SPACE_k(f(n)) \subseteq (N)(D)SPACE_k(c\,f(n))$, $0<c<1$

Alegem: DSPACE

Fie M o mTD

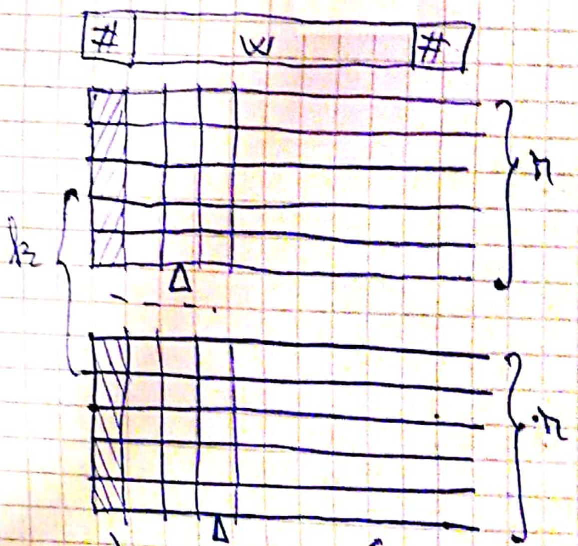


Obs.: Orice intrare de dimensiune $< n_0$ se poate decide în spațiul 1.

Fie $0<c<1$, arbitrar, fixat. Construim M':
- deterministă
- k benzi
- $L(M') = L(M)$
- $space(M', n) \le c \cdot f(n)$, $(\forall) n \ge n_0$.

Construcția M':

Fie $r \in \mathbb{N}$ a.î. $r \cdot c > 2$.



$q \in Q$

Starea curentă $(q, i_1, \dots, i_k)$

În M avem tranziția $(p, b_1, \dots, b_k, x_1, \dots, x_k) = \delta(q, a_1, \dots, a_k)$

$x_i \in \{-1; 0; 1\}$ stg, stă pe loc, dreapta.

În M' avem tranziția corespunzătoare $\delta'((q, i_1, \dots, i_k), v_1, \dots, v_k) =$
$= ((p, t_1, \dots, t_k), v_1', \dots, v_k', y_1, \dots, y_k)$

$$\begin{pmatrix} v_1[i_1] = a_1 \\ v_j[i_j] = a_j \end{pmatrix}$$

$$v_j'[p] = \begin{cases} v[p], & p \neq i_j \\ b_j, & p = i_j \end{cases}$$

$$t_j = \begin{cases} i_j + x_j, & \text{dacă } 1 \le i_j + x_j \le r \text{ și } y_j = 0 \\ 1, & \text{dacă } i_j + x_j = r+1 \text{ și } y_j = 1 \\ r, & \text{dacă } i_j + x_j = 0 \text{ și } y_j = -1 \end{cases}$$

$F' = F \times \{1, r\}^k$

M':
- deterministă
- k benzi
- $L(M') = L(M)$

$$Space(M', n) \le \left\lceil \frac{Space(M,n)}{r} \right\rceil \le \frac{Space(M,n)}{r} + 1 \le \frac{f(n)}{r} + 1 \le c \cdot f(n) \quad r$$

$f(n) + r \le r \cdot c \cdot f(n)$

$f(n) + r < 2 \cdot f(n) \le r \cdot c \cdot f(n)$

TEOREMĂ:

$\lim_{n \to \infty} \frac{f(n)}{n} = \infty$, $k \geq 2$ | $(N)(D)TIME_k(f(n)) = (N)(D)TIME_k(c \cdot f(n)), (\forall) c > 0$

când se pot eliminat const.?

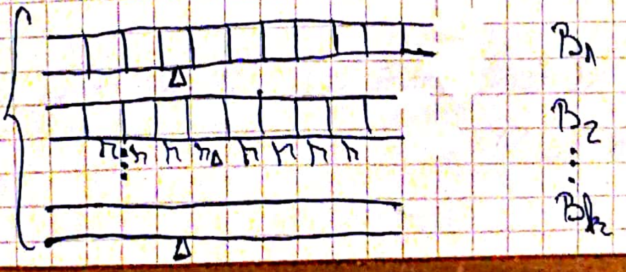
Dem.: Fie $k \geq 2$ $\quad (N)(D)TIME_k(f(n)) \subseteq (N)(D)TIME_k(c \cdot f(n))$.

$0 < c \le 1$

Alegem NTIME

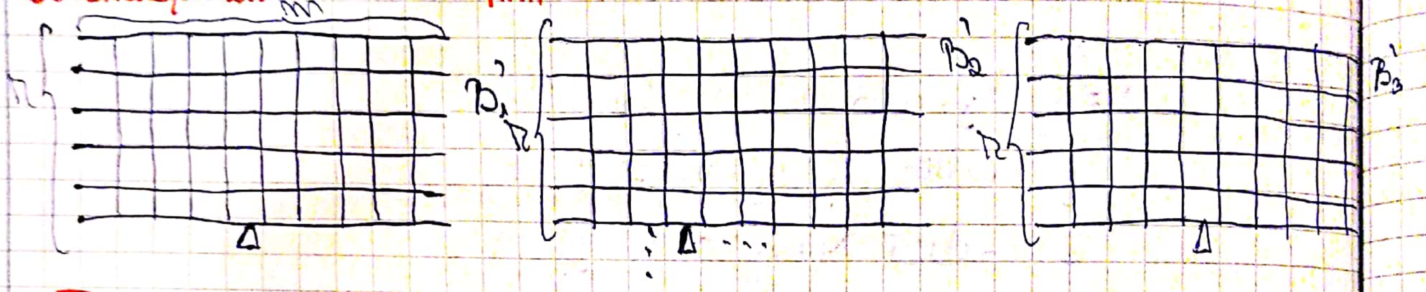
Fie M o mTN, $TIME(M, n) \le f(n)$, $(\forall) n \geq n_0$

M:



Construim M': → model.
↳ k_2 benzi
↳ $L(M) = L(M')$
↳ $Time(M', n) \leq c \cdot f(n)$.

Construcția lui M': $|m| = m$



Funcționarea lui M':

P1): Citește r simboluri consecutive de pe banda 1 și le scrie ca un simbol pe banda 2. (trebuie ≥ 2 benzi) timp: $n+1$.

P2): Poziționează capul R/W de pe B_2' la începutul benzii timp: $\lceil \frac{n}{r} \rceil$.

Facem corespondența: $B_1 \longrightarrow B_2'$
$B_2 \longrightarrow B_1'$
$B_j \longrightarrow B_j'$

M se află în starea q și ef tranziția $(s, b_1, \dots, b_{k_2}, X_1, \dots, X_k) = \delta(q, a_1, \dots, a_k)$

P3): Simulează tranziția curentă a lui M

Starea curentă $M'(q, i_1, \dots, i_k)$

- citește de pe fiecare bandă simbolul curent, cel din stg. și cel din dr. timp: 4.
- memorează în stare simbolurile citite

q - stare

s_1^1, s_2^1, s_3^1 — B_1'

s_1^2, s_2^2, s_3^2

$s_1^{k}, s_2^{k}, s_3^{k}$ ...

CAZ 1: Încearcă să depășească cele 3 segmente pe una dintre benzi.

Fie s starea lui M

t_1^1, t_2^1, t_3^1 - conținutul celor 3 segmente de pe B_1.

t_1^2, t_2^2, t_3^2 - conținutul celor 3 segmente de pe B_2.

t_1^k, t_2^k, t_3^k - conținutul celor 3 segmente de pe B_k.

Poziția capului de citire/scriere pe fiecare bandă luat ca:

- pe ce segment se află
- poz. în interiorul segmentului

Actualizează configurația sa:

$$\left.\begin{array}{l}\text{stare} \ldots \\ \text{pozițiile } i_1, \ldots, i_k \\ \text{simbolurile curente și vecini}\end{array}\right\} \text{Timp: } 4$$

$M' \to$ model:
- are k benzi
- $L(M) = L(M')$ {dacă M se oprește, și M' se oprește}

$$\text{Time}(M', n) \le n + 1 + \left\lceil \frac{n}{r} \right\rceil + 8\left\lceil \frac{f(n)}{r} \right\rceil$$

$$\le n + 1 + \frac{n+1}{r} + \frac{8 f(n)}{r} + 8$$

$$\le \frac{f(n)}{d} + 10 + \frac{f(n)}{rd} + \frac{8 f(n)}{r}$$

Fie $f(n) \ge n+1$, $n \ge 9 \Rightarrow 10 \le f(n)$

$$\le f(n) \left[\frac{1}{d} + 1 + \frac{1}{rd} + \frac{8}{r}\right]$$

$$\le f(n) \underbrace{\left[\frac{1}{d} + 1 + \frac{c}{16d} + \frac{8c}{16}\right]}_{\le c}$$

$16 + 16d + c + 8cd \le 16dc \Rightarrow 8d(2-c) \le -16 - c \Rightarrow *$

$* \ 8d(c-2) \ge 16 + c \Rightarrow d \ge \frac{16+c}{8(c-2)}$

$\text{Time}(M', n) \le c\, f(n)$, $(\forall)\, n \ge \max(n_d, 9)$

cel puțin r pași ca să iasă în afara segmentului

$\lim \frac{f(n)}{n} = \infty$

$(\forall)\, d > 0\ (\exists)\, n_d$ a.î. $(\forall)\, n \ge n_d$, $f(n) \ge nd \Rightarrow n \le \frac{f(n)}{d}$

$rc \ge 16 \Rightarrow \frac{1}{r} \le \frac{c}{16}$

19.12.2019

TEOREMĂ:

$$\left.\begin{array}{l}\lim\limits_{n\to\infty} \frac{f(n)}{n} = \infty \\ k\geq 2\end{array}\right| \Rightarrow (N)(D)TIME_k(f(n)) = (N)(D)TIME_k(c\cdot f(n)), (\forall) c>0$$

TEOREMĂ:

$$(N)(D)TIME_k(cn) = (N)(D)TIME_k((1+\varepsilon)n), (\forall) k\geq 2, \varepsilon>0$$

Dem.:

Aceeași dem.

La evaluarea timpului mașinii M':

$$TIME(M', n) \leq \cancel{n} + \frac{n}{r} + \frac{8cn}{r} + 10 \leq \cancel{n} + \varepsilon n$$

Aleg r a.î. $10 \leq \frac{n}{r} \Longrightarrow \boxed{n \geq 10r}$

$$\frac{\cancel{n}}{r} + \frac{8c\cancel{n}}{r} + \frac{\cancel{n}}{r} \leq \varepsilon \cancel{n}$$

$r \geq \frac{2+8c}{\varepsilon} \longrightarrow$ Alegerea lui r din dem.

Dacă $n<10r$, atunci decizia se ia în $n+1$ pași

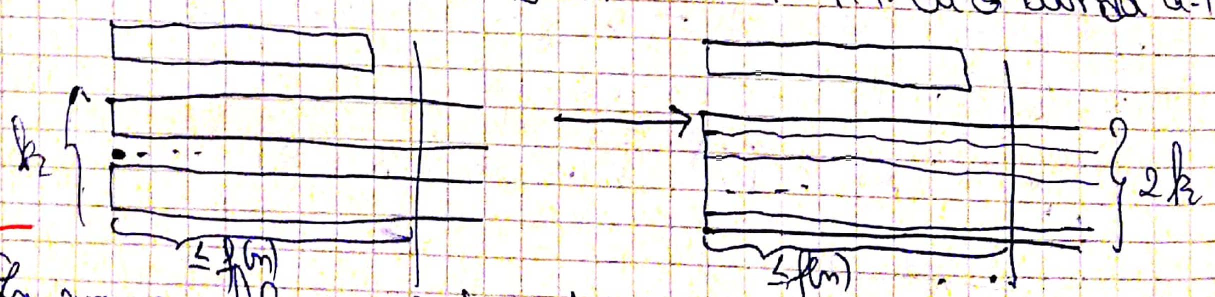
COMPRIMAREA BENZILOR

TEOREMĂ:

$$(N)(D)SPACE_k(f(n)) = (N)(D)SPACE_1(f(n)), (\forall) k\geq 1, f(n)\geq 1$$

Dem.:

M este o M.T. cu k benzi $\longrightarrow$ M' M.T. cu o bandă a.î. $L(M)=L(M')$



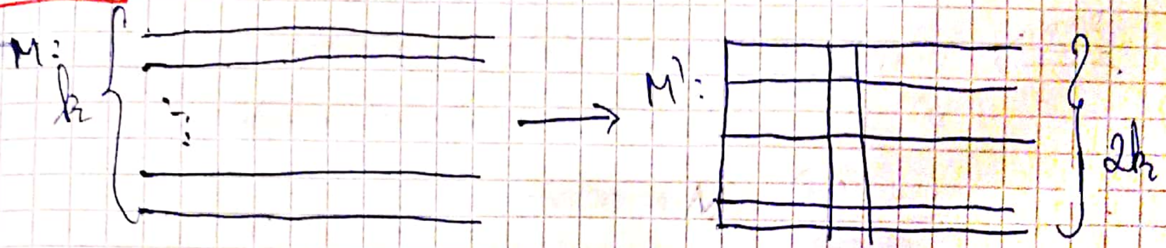
La examen: Alegem să dem. teorema în primul caz (echivalența mașinilor det. și nedet.).

→ + dem. de la echivalența maș. Turing det. și nedet. (cea cu 3 benzi).

TEOREMA:

$(N)(D)TIME_k(f(n)) \subseteq (N)(D)TIME_1(f^2(n))$, $(\forall) k \geq 1$, $f(n) \geq n$

Dem.:



Timp M': $\leq f(n) + 3f(n) = 4f(n)$ (pt. o mișcare a lui M)

Timp $(M', n) \leq 4f(n) \cdot f(n) = 4f^2(n)$

M are $k \geq 2$ benzi

$\downarrow$

M_1 are $k \geq 2$ benzi

Timp $(M_1, n) = \frac{1}{2} f(n)$

M se înlocuiește cu M_1: $4\left(\frac{1}{2} f(n)\right)^2 = f^2(n)$.

(în limbaj: accelerez mașina M cu un factor $\frac{1}{2} f(n)$)

TEOREMĂ:

Pt. orice fct. recursivă $f(n)$ există un limbaj recursiv L a.î. $L \notin DTIME(f(n))$.

Dem.:

1) Exemplu pt. L recursiv.

2) Exemplul $L \notin DTIME(f(n))$.

dacă nu are indice ⇒ nu contează câte benzi are

(1) $L = \{x_j / x_j \in \{0,1\}^+, x_j \notin L(M), \hat{M} = j$ în timp $f(n)\}$

poziția

L este recursiv.

Fie M maș. Turing

P1): Calculează pe o bandă $f(n)$

P2): Pe intrarea x găsește j a.î. $x = x_j$

P3): Găsește M' a.î. $\hat{M}' = j$
P4): Simulează M' pe intrarea w.

exp.: M' acceptă în timp $f(n)$: M respinge
M' nu decide în timp $f(n)$: M acceptă
$L = L(M)$, M se oprește.

ii) Pres. A că $L = L(M_1)$ a.î. $Time(M_1, n) \le f(n)$
Fie $\hat{M}_1 = p$, $w_p \in L = L(M_1) \Longleftrightarrow w_p \notin L(M_1)$ în timp $f(n)$ ↯

IERARHII DE CLASE DE COMPLEXITATE

Def.: O funcție $f(n)$ este spațiu/timp construibilă dacă există o MT a.î. $(\forall) n \geq 1$ $(\exists)$ w_n cu $|w_n| = n$ și $Space(M, w_n)$ / $Time(M, w_n) = f(n)$. În general $Space(M, n)$ / $Time(M, n) \le f(n)$, $(\forall) n \geq 1$.

Def.: O funcție $f(n)$ este spațiu/timp construibilă complet dacă există o MT a.î. $(\forall) n \geq 1$ și $(\forall) w$, $|w| = n$, $Space(M, w)$ / $Time(M, w) = f(n)$.

TEOREMĂ:

$S_2(n)$ este spațiu construibilă complet
$S_2(n) \geq \log n$
$\lim_{n \to \infty} \frac{S_1(n)}{S_2(n)} = 0$

$\Rightarrow DSPACE(S_2(n)) \setminus DSPACE(S_1(n)) \neq \emptyset$

Dem.: Fie M. Pe input-ul w:
P1) Marchează $S_2(n)$ celule pe o bandă (S_2 este spațiu construibilă complet).
M va avea spațiul mărginit de $S_2(n)$.
$L(M) \in DSPACE(S_2(n))$.

B) input: $m \in \{0,1\}^*$ ⟶ codificarea unei mT

$\downarrow$

$\#(M')$ ⟶ codul unei mașini Turing M' oarecare.

M' ⟶ deterministă

⟶ $Space(M', m) \leq S_1(m)$.

⟶ are k_1 benzi și t simboluri de lucru

presp. $k_1 = 1$.

P3) Simulează M' pe intrarea $\#(M')$.

__Obs.:__ 1) Orice simbol al lui M' va fi codificat în binar.

2) Pt. simulare, M are nevoie de spațiu mărginit de $S_1(m) \cdot \log t$.

3) Dacă $S_1(m) \cdot \log t \leq S_2(m)$, atunci M acceptă $\Longleftrightarrow$ $\Longleftrightarrow$ M' respinge.

Există (codificare) $\#(M') = m$ a.î. $|m| = p$ a.î. $S_1(p) \cdot \log t \leq S_2(p)$

$$\lim \frac{S_1(m)}{S_2(m)} = 0 \Rightarrow (\exists)\ m_0 \text{ a.î. } (\forall)\ m \geq m_0,\ S_1(m) \cdot c \leq S_m(m).$$

$\#(M') = 0000\ldots0\ \#(M')$.

$\downarrow$

codificarea rămâne aceeași

intrarea lui M este for $|\#(M')| \geq m_0$

altfel este $\underbrace{00\ldots0}_{m_0}\#(M')$