17.11.2017

Calculabilitate și complexitate
- Curs 7 -

$\#(M) \in \{(,),0,1,2,R,L\}^*$ - codificarea lui M (M e mT)

$\#(w)$ - codificarea lui w peste $\{0,1\}$

$\widehat{\#(M)}$ - indexul mașinii Turing
(poz. codificării în șirul codificărilor ordonat lexicografic)

P - problemă de decizie

$P(x_1, ..., x_m)$

$L_P = \{<P(a_1, ..., a_m)> \mid P(a_1, ..., a_m) = true\}$

- REC - familia/clasa mulțimilor recursive
  A este recursivă $\Leftrightarrow$ χ_A este calculabilă cu mT care se oprește
  $\Leftrightarrow$ A ~~este~~ $= L(M)$, M - mT care se oprește

P este decidabilă $\Leftrightarrow L_P \in REC$

- RE - familia mulț. recursive
  $A \in RE$, $A = L(M)$ unde M este mT
  (aici nu se oprește mereu)

- NRE - familia tuturor mulțimilor.

$REC \subseteq RE \subseteq NRE$

$L_u = \{\#(M)\$w \mid w \in L(M)\}$
$\hookrightarrow$ limbajul universal

$L_u \in RE$

$RE \subsetneq NRE$

$L_d = \{w \mid w \notin L(M), \widehat{\#(M)} = \hat{w}\}$

<u>Afirm:</u> $L_d \notin RE$

<u>Dem:</u> Pp absurd $L_d \in RE \Rightarrow$ există M_d cu prop că

$L(M_d) = L_d$

Fie $\widehat{\#(M_d)} = k$	$x \in L_d = L(M_d)$
Fie x a.î. $\hat{x} = k$	$\Updownarrow$
	$x \notin L(M)$ a.î. $\widehat{\#(M)} = \hat{x} = k$

$\Rightarrow$ ~~...~~ $M = M_d$ ↯

$L_u \in RE$

<u>Afirm 2:</u> $L_u \notin REC$

<u>Dem:</u> ~~...~~ Pp absurd $L_u \in REC \Rightarrow$

există M_u — $L_u = L(M_u)$
— se oprește

Construim M_0:

input w

calculează $k = \hat{w}$

calculează $\#(M)$ a.î. $\widehat{\#(M)} = k$

simulează M_u pe intrarea $\#(M)\$w$

M_0 acceptă $\Leftrightarrow$ M_u nu acceptă

$L(M_0) = \{w \mid M_u \text{ nu acceptă } \#(M)\$w\} =$

$= \{w \mid w \notin L(M)\} = L_d$ ↯

$\hat{w} = \widehat{\#(M)}$

Teoremă: $REC \subsetneq RE \subsetneq NRE$

Propriétăți:

1) REC este închisă la $\cup, \cap, C$ complementare

2) $A \in REC$ dacă $A \in RE$ și $CA \in RE$

3) RE este închisă la $\cup, \cap$, dar nu e închisă la C

Dem:

1) ...

2) "numai dacă" $\Leftarrow$: $A \in REC \subseteq RE$

$CA \in REC \subseteq RE$

"dacă" $\Rightarrow$:

3) ..

O proprietate $\mathcal{P}$ pe RE este o submulțime a lui RE

$\mathcal{P} \subseteq RE$

$\mathcal{P}$ este netrivială dacă $\mathcal{P} \neq \emptyset$ și $\mathcal{P} \neq RE$

$L_{\mathcal{P}} = \{ \#(M) \mid L(M) \in \mathcal{P} \}$

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

Teoremă (Rice):

Orice proprietate netrivială pe RE nu este decidabilă.

Dem: Pp absurd că $L_{\mathcal{P}}$ este recursiv $\Leftrightarrow$

Există $M_{\mathcal{P}}$ a.î. $L(M_{\mathcal{P}}) = L_{\mathcal{P}}$

$M_{\mathcal{P}}$ se oprește

Cazul 1: $\emptyset \notin \mathcal{S}$

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

Fie M și w arbitrare, fixate.

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

Input x
Ignoră intrarea x și simulează M pe intrarea w
Dacă M acceptă M' revine la inputul său adică x
Simulează pe M_L pe intrarea x
Acceptă $\Leftrightarrow M_L$ acceptă.

Construim M_0:

Input $\#(M)\$w$
Calculează $\#(M')$, M' e mașina constr. anterior pentru perechea M, w (de pe rândul anterior)
Simulează $M_{\mathcal{S}}$ pe intrarea $\#(M')$
Acceptă $\Leftrightarrow M_{\mathcal{S}}$ acceptă

(i) M_0 se oprește pe fiecare intrare.

(ii) $L(M_0) = \{\#(M)\$w \mid M_{\mathcal{S}} \text{ acceptă } \#(M')\} =$

$= \{\#(M)\$w \mid \#(M') \in L_{\mathcal{S}}\} =$

$= \{\#(M)\$w \mid \underbrace{L(M') \in \mathcal{S}}_{\Downarrow\; L(M') = L \Leftrightarrow w \in L(M)}\} =$

$= \{\#(M)\$w \mid w \in L(M)\} = L_u$ ∞

Cazul 2: $\emptyset \in \mathcal{S} \Rightarrow \emptyset \notin C\mathcal{S}$ și se aplică cazul 1.

17.11.2017

Calculabilitate și complexitate
– Curs 6 –

Codificarea programelor

Variabile : $Y, X_1, Z_1, X_2, Z_2, \ldots$

$\#(Y) = 0$
$\#(Z_3) = 6$ $\Big\rangle$ poziția în șir

Etichete: $E, A_1, A_2, \ldots$

$\#(E) = 1$
$\#(A_3) = 4$ $\Big\rangle$ poz. în șir

Instrucțiuni: $I \longrightarrow \#(I) = \langle \overset{a}{\bullet}, \langle \overset{b}{\bullet}, \overset{c}{\bullet} \rangle\rangle$

c
- $\to 0 : V \leftarrow V$
- $\to 1 : V \leftarrow V + 1$
- $\to 2 : V \leftarrow V \dot{-} 1$
- $\to \#(L)+2 : \text{IF } V \neq 0 \text{ GOTO } L$

a
- $\to 0$: dacă I nu este etichetată
- $\to \#(L)$: dacă I are eticheta L

$b \longrightarrow \#(V)$ (indexul variabilei)

Exemplu:

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

$$A_2 : X_3 \leftarrow X_3 + 1 \leadsto \langle 3, \langle 5, 1 \rangle\rangle =$$

$$= \langle 3, 2^5 \cdot 3 - 1 \rangle = \langle 3, 95 \rangle = 2^3 \cdot 191 - 1 =$$

$$= 8 \cdot 191 - 1 = 1527$$

Viceversa: $2347 \rightsquigarrow ?$

$$\left.\begin{array}{l} 2348 = 2^2 \cdot 587 \\ 587 = 2y+1 \Rightarrow y = 293 \\ 294 = 2^1 \cdot 147 \\ y' = 73 \end{array}\right| \Rightarrow \langle 2, \langle 1, 73 \rangle \rangle$$

$\langle 2, \langle 1, 73 \rangle \rangle \rightsquigarrow A_1 :$ IF $X_1 \neq 0$ GOTO A_{71}

Fie P un program standard cu instrucțiunile $I_1, I_2, \ldots, I_n$

$$\left[\#(I) = 0 \iff I \text{ este } Y \leftarrow Y \right]$$

$$\#(P) = \left[\#(I_1), \#(I_2), \ldots, \#(I_n) \right] - 1 =$$

$$= p_1^{\#(I_1)} \cdot p_2^{\#(I_2)} \cdot \ \ldots \ \cdot p_n^{\#(I_n)} - 1$$

$\cancel{\exists}\ \mathbb{N} \longrightarrow M(P) \longrightarrow$ mult. programelor standard

$\hookrightarrow$ corespondență biunivocă

Ex: 2347

$2348 = 2^2 \cdot 587$

$\hookrightarrow$ al k-lea nr prim $\Rightarrow p_k$

* instr. cu nr 2 *

$$\left. \begin{array}{l} Y \leftarrow Y \\ \ldots \\ Y \leftarrow Y \end{array} \right\} k-2 \text{ ori}$$

* instr cu nr k *

$HALT: \mathbb{N}^2 \to \mathbb{N}$; $HALT(x,y) = \begin{cases} 1, \text{ dacă } P \text{ se oprește pe } x \\ 0, \text{altfel} \end{cases}$

$HALT(x,y)$ = Programul P cu indexul y ($\#(P) = y$) se oprește pe intrarea x

Teoremă:

Funcția HALT **nu** este calculabilă cu programe standard.

Dem:

Pp prin absurd că HALT e calculabil cu PS.

Fie P programul care calculează HALT

Fie A: IF HALT(X,X) GOTO A ⟸ programul P'

P' se termină $\overset{\text{pe intrarea } x}{\Longleftrightarrow}$ HALT (X,X) = 0 ⟺

⟺ programul cu nr x nu se termină pe intrarea x

$\#(P') = y$

~~HALT(y,y) ⟺~~ P' se termină pe intrarea y ⟺

HALT(y,y) ⟺ ! HALT(y,y)

Codificarea mașinilor Turing:

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

Mulțimea universală a stărilor: $q_0, q_1, q_2, \ldots$
$q_0 \to 01$, $q_1 \to 011$, $q_2 \to 0111$

Mulț. universală a simbolurilor: $s_0, s_1, \ldots$
$s_0 \to 02$, $s_1 \to 022$

$(s, b, R) \in \delta(q, a) \longleftrightarrow (q, a, s, b, R)$

Codificare: $(\#(q)\,,\#(a)\,,\#(s)\,,\#(b)\,,R)$

$\#(M) = \#(Q)\#(U)\#(\Sigma)\#(F)\#(\delta)$

mT

$(, , , ,)(, , , ,)\ldots(, , , ,)$

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

$M_0 \longrightarrow \#(M)\#(w)$

M_0 nu va putea să spună dacă M se oprește pe w.

- $\phi^{(n)}(x_1, \ldots, x_n, t) = \varphi_P(x_1, \ldots, x_n)$

  unde $\varphi_P \rightarrow$ funcția calculată de programul P

  $t = \#(P)$

$\phi^{(n)}$ s.n. funcția universală de n variabile

Teoremă: Funcția universală de n variabile este calculabilă cu PS pentru orice n.

Dem: Intrare: $X_1, \ldots, X_n, X_{n+1}$

$\hookrightarrow \#(P),\ P: I_1, \ldots, I_m$

nr. instr. curente

$K \stackrel{\text{not}}{=} \text{Nic} \leftarrow 1$

$S \leftarrow \prod_{i=1}^{n} p_{2i}^{x_i}$

$Z \leftarrow X_{n+1} + 1$

$C: I \leftarrow (Z)_K$

$V \leftarrow l(r(I))$

$TI \leftarrow r(r(I))$

$\cancel{C}$: IF $((K == 0) \| (K > m))$ GOTO F

IF $TI = 0$ GOTO N (de la next → nu se întâmplă nimic)

tipul instr.

IF $TI = 1$ GOTO A

IF $TI = 2$ GOTO D

IF $P_v \nmid S$ GOTO N

~~$K \leftarrow \min_t [\ldots r(r(I)) \ldots]$~~

$K \leftarrow \min_t [TI - 2 = l((Z)_t)]$ // dacă pt niciun t nu e valabil => K = 0

GOTO C

A: $S \leftarrow S * P_V$

GOTO N

D: ~~S ← S~~

IF $P_V \nmid S$ GOTO N

$S \leftarrow S / P_V$

N: $K \leftarrow K+1$

GOTO C

F: $Y \leftarrow (S)_1$

Vineri 17 nov
ora 16:00
~~curs~~ Sala 1

Calculabilitate și complexitate

– Curs 5 –

03.11.2017

Funcții recursive

1. $x!$
2. x^y
3. $p(x) = \begin{cases} x-1, & \text{dacă } x>0 \\ 0, & \text{altfel} \end{cases}$
4. $x \dot{-} y$ (dacă $y>x$, rez. e 0)
5. $|x-y|$
6. $\alpha(X) = \begin{cases} 1, & x=0 \\ 0, & x>0 \end{cases}$
7. $x=y$
8. $x<=y$
9. $y|x$
10. Prime(x)
11. $\lfloor x/y \rfloor = \min\limits_{c} [(c+1)y > x]$
12. $R(x,y) = x \% y$
13. ! p_n – al n-lea număr prim
14. Pereche: $\langle x,y \rangle = 2^x(2y+1) \dot{-} 1$

Obs: $\langle \cdot,\cdot \rangle$ este injectivă în ambele argumente și surjectivă

Dem: $\langle x,y \rangle = z$

$2^x(2y+1) - 1 = z$

$2^x(2y+1) = z+1$

$x = \min\limits_{t} [2^{t+1} \nmid (z+1)]$

$l(z) = x$, $\langle x,y \rangle = z$

$r(z) = y$, $\langle x,y \rangle = z$

15. Gödelizare (a unei secv)

$[a_1, a_2, \ldots, a_m] = p_1^{a_1} \cdot p_2^{a_2} \cdot \ldots \cdot p_m^{a_m}$

$[1,0,3,4] = 2^1 \cdot 3^0 \cdot 5^3 \cdot 7^4$

$z = 34 = 2 \cdot 17$

$\to [1,0,0,0,0,0,1]$

16. $(z)_i$, pt $[1,0,3,4] \Rightarrow (z)_2 = 0$, $(z)_3 = 3$, $(z_4) = 4$

17. $Lt(z)$ = lungimea secvenței a cărei Gödelizare este z

$Lt(34) = 7$

Teoremă:

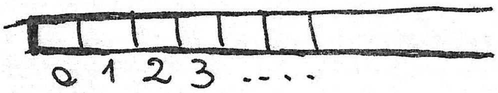
Orice funcție Turing calculabilă este recursivă.

Dem:

Fie $M = (Q, V, U, \delta, q_0, F, B)$ o mT deterministă care calculează $f: \mathbb{N}^k \to \mathbb{N}$

$Q = \{0, 1, 2, \ldots, m\}$

$U = \{0, 1, 2, 3, \ldots, p\}$



Config $\alpha q \beta$

$\#(\alpha)$ = secv. de numere asociată simbolurilor din α

$\alpha = aab01 \Rightarrow \#(\alpha) = 2,2,3,0,1$

$\#(q)$ = numărul asociat lui q

$\alpha q \beta \longrightarrow \langle \#(q), \langle |\alpha|, [\#(\alpha), \#(\beta)] \rangle \rangle$

Config inițială:

$\langle 0, \langle 0, [\,\underbrace{1,1,1,1,\dots,1}_{x_1}, 0, \underbrace{1,1,1,\dots,1}_{x_2}, 0, \dots, \underbrace{1,\dots,1}_{x_{k+1}}] \rangle\rangle$

Def:

$C_M(x,n)$ – nr atașat configurației mașinii M la pasul n, pe intrarea $x=(x_1,\dots,x_k)$

— este C_M recursivă?

config. inițială

$C_M(x,0) = \langle 0, \langle 0, [\underbrace{1,\dots,1}_{x_1}, 0, \underbrace{1,\dots,1}_{x_2}, 0, \dots, \underbrace{1,\dots,1}_{x_{k+1}}] \rangle\rangle$

$C_M(x,n+1)$ – $C_M(x,n)$, dacă M se oprește la pasul n

— altfel:

- $h_1(z) = \begin{cases} \text{nr. stării în care trece M din config cu nr. atașat } z \text{, dacă } z \text{ e un nr atașat unei config validă} \\ \uparrow \text{ (nedefinit), altfel} \end{cases}$

- $h_2(z) = \begin{cases} \text{poziția pe bandă a capului de citire/scriere în care trece M din config cu nr. atașat } z\text{, dacă } z \text{ e un nr. atașat unei config val.} \\ \uparrow\text{, altfel} \end{cases}$

- $h_3(z) = \begin{cases} \text{nr. atașat conținutului benzii în care trece M din config} \dots\dots \\ \uparrow\text{, altfel} \end{cases}$

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

- $g_1(a,b)$ = nr. stării în care trece M din starea cu nr. a, citind simbolul cu nr b.

- $g_2(a,b) = \begin{cases} 0\text{, dacă M se depl la stânga din starea cu nr a citind simbolul cu nr } b \\ 2\text{, dacă M se depl la dreapta} \dots \end{cases}$

- $g_3(a,b) =$ nr. simbolului scris de M aflându-se în starea cu nr. a și citind simbolul cu nr. b.

$$h_1(z) = g_1\left(l(z), (r(r(z)))_{l(r(z))+1}\right)$$

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

$$h_3(z) = r(r(z)) \Big/ p_{l(r(z))}^{(r(r(z)))_{l(r(z))}} * p_{l(r(z))}^{g_3\left(l(z), (r(r(z)))_{l(r(z))+1}\right)}$$

$$z = [a_1, a_2, \dots, a_i, \dots, a_n] \longrightarrow [a_1, a_2, \dots, b_i, \dots, a_n] \Leftrightarrow$$

$$z \longrightarrow z \big/ p_i^{(z)_i} * p_i^{b_i}$$

h_1, h_2, h_3 rec $\Rightarrow C_M$ rec.

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

$$nr_M(x) = \min_n \left[C_M(x,n) = C_M(x,n+1) \right]$$

$$f(X) = Lt\left(r(r(C_M(x, nr_M(x))))\right) \dot{-} 1$$

$$\begin{tikzcd} MT \arrow[rr] & & FR \arrow[dl] \\ & PS \arrow[ul] & \end{tikzcd}$$

Bibliografie:

1. M. Davis,

27.10.2017

Calculabilitate și complexitate

- Curs 4 -

Programe standard

Funcții recursive

- Funcții elementare:

1) Succesor $succ : \mathbb{N} \to \mathbb{N}$, $succ(x) = x+1$
2) Constante $C_k^{(m)} : \mathbb{N}^m \to \mathbb{N}$, $C_k^{(m)}(x_1, \dots, x_m) = k$
3) Proiecții $\pi_k^{(m)} : \mathbb{N}^m \to \mathbb{N}$, $\pi_k^{(m)}(x_1, \dots, x_m) = x_k$

1) Succesor:

succ: $Y \leftarrow X_1$

$Y \leftarrow Y + 1$

2) $C_k^{(m)}$:

$$\left.\begin{array}{l} Y \leftarrow Y+1 \\ Y \leftarrow Y+1 \\ \vdots \\ Y \leftarrow Y+1 \end{array}\right\} \text{de } k \text{ ori}$$

3) $\pi_k^{(m)} : Y \leftarrow X_k$

- Operații:

1) Compunerea funcțională

$f : \mathbb{N}^k \to \mathbb{N}$ este obținută prin compunere funcțională din $h : \mathbb{N}^m \to \mathbb{N}$

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

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

Întrebare: Dacă g_i și h sunt calculabile cu prog standard este f calculabilă cu PS?

Răspuns:

$Z_1 \leftarrow g_1(x_1, \ldots, x_k)$

$Z_{p+1} \leftarrow g_2(x_1, \ldots, x_k)$

(prima var. de lucru liberă; să nu fi fost fol în g_1)

// toate etichetele sunt diferite între ele

pt a fol. Z_2 tb să aduc toate var în afară ~~loc~~ de Z_1 la 0.

⋮

$Z_m \leftarrow g_m(x_1, \ldots, x_k)$

$Y \leftarrow h(Z_1, \ldots, Z_m)$

② Recurența primitivă

$f: \mathbb{N}^{k+1} \longrightarrow \mathbb{N}$ este obț. prin recurență primitivă din fct $g: \mathbb{N}^k \longrightarrow \mathbb{N}$ și $h: \mathbb{N}^{k+2} \longrightarrow \mathbb{N}$ dacă

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

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

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

exemplu: $fact(t+1) = h(t, fact(t)) =$ ~~produs~~

$= produs(succ(t), fact(t))$

Întrebare: Dacă g și h sunt calc. cu P.S. este f calculabilă cu P.S.?

Răspuns:

$\text{IF } X_{k+1} \neq 0 \text{ GOTO } A_1$

$Y \leftarrow g(x_1, \dots, x_k)$

$\text{GOTO } E$

initializat cu 0

$A_1: Z_1 \leftarrow g(x_1, \dots, x_k)$

$A_2: Z_3 \leftarrow h(x_1, \dots, x_k, Z_2, Z_1)$

$X_{k+1} \leftarrow X_{k+1} - 1$

$Z_2 \leftarrow Z_2 + 1$

$\cancel{\text{IF } X_{k+1} \neq 0}$

$Z_1 \leftarrow Z_3$

$\text{IF } X_{k+1} \neq 0 \text{ GOTO } A_2$

$Y \leftarrow Z_3$

3 Minimizare nemărginită

$f: \mathbb{N}^k \xrightarrow{o} \mathbb{N}$ se obține prin minimizare nemărg. din fct. $g: \mathbb{N}^{k+1} \longrightarrow \mathbb{N}$ dacă

$$f(x_1, \dots, x_k) = \min_t [g(x_1, \dots, x_k, t) = 0]$$

Întrebare: Dacă g e calc. cu P.S. este și f calc cu P.S

$$
R: \left[\begin{array}{l}
A_2: Z_1 \leftarrow g(x_1, \ldots, x_k, Y) \\
\text{IF } Z_1 \neq 0 \text{ GOTO } A_1 \\
\text{GOTO } E \\
A_1: Y \leftarrow Y + 1 \\
\qquad \text{GOTO } A_2
\end{array}\right.
$$

! O funcție este (parțial) <u>recursivă</u> dacă este o funcție (parțial definită) care se obține din funcții elementare, prin aplicarea operațiilor ~~[illegible]~~ (1) - (3)

<u>Teoremă:</u> Orice funcție recursivă este calc. cu programe standard.

$$
\begin{tikzcd}
MT \arrow[rr, dashed, "?"] & & FR \arrow[ll, bend right] \arrow[dl] \\
& PS \arrow[ul] &
\end{tikzcd}
$$

<u>Ex:</u> $f(x_1, x_2) = x_1 + x_2$

$f(x_1, 0) = x_1 = \pi_1^{(1)}(x_1)$

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

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

$\cancel{= succ(x_1, x_2, f(x_1, x_2))}$

$\cancel{= \pi_3^{(3)}(x_1, x_2}$

$= succ(\pi_3^{(3)}(x_1, x_2, f(x_1, x_2))$

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

$f'(x_1, 0) = 0 = C_0^{(1)}(x_1)$

$f'(x_1, x_2+1) = f'(x_1, x_2) + x_2$

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

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

$= f(\pi_2^{(3)}(x_1, x_2, f(x_1, x_2))$

f recurentă primitivă din g și h
g și h sunt Turing Calculabile.

20.10.2017

Calculabilitate și complexitate
– Curs 3 –

M – mT cu n benzi

M' – echivalenta lui M, mT cu o bandă

M': ① Scanează banda pentru a "găsi" și "memora" simbolurile citite de M.

② Se "actualizează" conținutul benzii cf. tranziției efectuate de M.

③ La fiecare început de fază ①/sfârșit de fază ②, starea lui M' este aceeași cu cea a lui M

④ Mulțimea stărilor finale a lui M' este mulț. stărilor finale ale lui M.

$$\langle q, \underbrace{0,0,\dots,0,a,\dots,0,a}_{2m+1} \rangle \to \langle q, a_1, 0, \dots, 0, 0, \dots, 0, 0 \rangle$$

$$\dots \to \langle q, a_1, \dots, a_m, b_1, \dots b_m, s \rangle \to \langle q, 0, \dots, 0, b_1, \dots, b_m, s \rangle$$

Programe standard

Limbajul $\mathcal{S}$:

Variabile: – de intrare: $x_1, x_2, \dots, x_m, \dots$

– de ieșire: y

– de lucru: $z_1, z_2, \dots$

Etichete: $E, A_1, A_2, \dots, A_m, \dots$

Instrucțiuni: $V \leftarrow V$ (variabilă)

$V \leftarrow V + 1$

$V \leftarrow V \dot{-} 1$

(Dacă $V \geq 1$: $V-1$
Altfel nimic)

IF $V \neq 0$ GOTO L

etichetă

↓

Dacă $V=0$, transferul se face la instr. imediat următoare

Dacă $V \neq 0$, transferul se face la prima instr cu eticheta L (prima din program)

Fie P un program standard

- variabilele y și cele de lucru sunt inițializ. cu zero.
- variabilele de intrare au valori numere naturale

(P calculează o $f: \mathbb{N}^l \to \mathbb{N}$)

- instrucțiunile se execută secvențial

Terminarea:
- nu mai sunt instr. de executat
- salt la o etichetă care nu există
- salt la eticheta E

Ex: $f(x) = x$

(*)

$$\left\{\begin{array}{l}
\text{IF } X_1 \neq 0 \text{ GOTO } A_1 \\
Z_1 \leftarrow Z_1 + 1 \\
\text{IF } Z_1 \neq 0 \text{ GOTO } E \\
A_1:\ X_1 \leftarrow X_1 \dot{-} 1 \\
\quad y \leftarrow y + 1 \\
\quad \text{IF } X_1 \neq 0 \text{ GOTO } A_1
\end{array}\right.$$

$\left[\begin{array}{l} Z_1 \leftarrow Z_1 + 1 \\ \text{IF } Z_1 \neq 0 \text{ GOTO } E \end{array}\right] \iff \boxed{\text{GOTO } L}$, L etichetă

Pt ca (*) să fie echivalent cu instrucțiunea $V \leftarrow V'$, trebuie să mai luăm un Z_2 pe care îl incrementăm odată cu y și după "vărsăm" Z_2 în X

$$\left[\begin{array}{l}
A_2:\ Z_2 \leftarrow Z_2 \dot{-} 1 \\
\quad X_1 \leftarrow X_1 + 1 \\
\quad \text{IF } Z_2 \neq 0 \text{ GOTO } A_2
\end{array}\right.$$

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

O funcție $f : \mathbb{N}^k \to \mathbb{N}$ este calculabilă cu prog. standard dacă $\exists\, \mathcal{P}$ a.î. $\forall (x_1, \dots, x_k) \in \mathbb{N}^k$, $\mathcal{P}$ se termină pe intrarea $X_i = x_i$; $i \in \overline{1,k}$ și $y = f(x_1, \dots, x_k) \Leftrightarrow f(x_1, \dots, x_k)$ este definită.

<u>Teoremă</u>: Orice funcție calculabilă cu program standard este Turing calculabilă

<u>Dem:</u>

Fie $\mathcal{P}$ un program standard cu instrucțiunile $I_1, \dots, I_n$, variabilele de intrare $X_1, \dots, X_k$, variabile de lucru $Z_1, \dots, Z_m$

<u>Construim M:</u>

Inițial :

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

$X_i = x_i$, $i = \overline{1,k}$

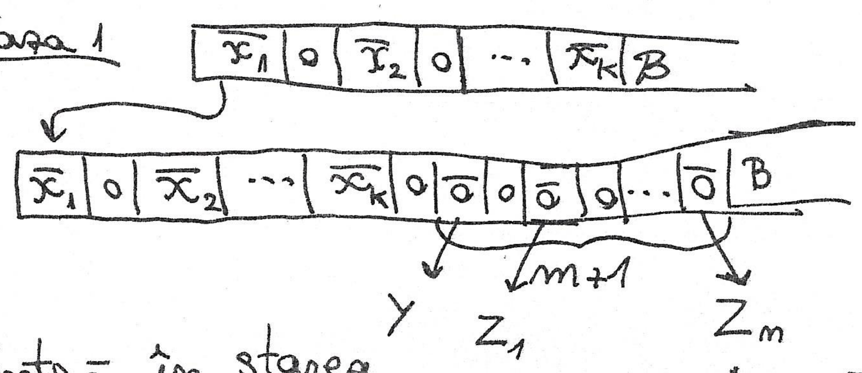
Presupunem că $\mathcal{P}$ se termină pe intrarea $(x_1, \dots, x_n)$

$\overline{y}$	0	0	... 0	B	

$Y = y$

M:

<u>Faza 1</u>



Intră în starea asociată instr $\langle I_1 \rangle$

<u>Faza 2</u> M se găsește în starea $\langle I_j \rangle$, $j = \overline{1,n}$

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

→ Dacă $I_j : V \leftarrow V+1$, M crește secvența de 1 asociată var. V și trece în starea $\langle I_{j+1} \rangle$

→ Dacă $I_j : V \leftarrow V \dot{-} 1$, M verifică dacă secv de 1 asociată lui V este de lungime 1. Dacă <u>da</u>, trece în $\langle I_{j+1} \rangle$. Dacă <u>nu</u>, scade lungimea secvenței cu 1 și trece în $\langle I_{j+1} \rangle$

→ Dacă I_j : IF $V \neq 0$ GOTO L, M verifică dacă secvența de 1 asociată lui V este de lungime 1.

Dacă <u>da</u>, M trece în $\langle I_{j+1} \rangle$

Dacă <u>nu</u>, M trece în starea $\langle I_r \rangle$, unde

$$r = \begin{cases} n+1, & \text{dacă nu există instr cu eticheta L (sau dacă L este E)} \\ t, & t = \min\{ s \mid I_s \text{ are eticheta } L \} \end{cases}$$

<u>Faza 3:</u> M se găsește în starea $\langle I_{n+1} \rangle$

Conținutul benzii se reduce doar la secv de 1 asociată lui y (shiftare la stânga, după scriem 0 la dreapta lui y). După, M trece în stare finală.

13.10.2017

Calculabilitate și complexitate
— Curs 2 —

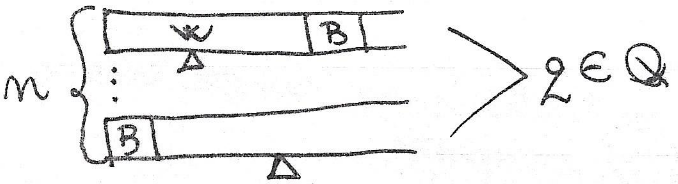
Mașina Turing deterministă

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

M este deterministă dacă $card(\delta(q,a))\leq 1, \forall q\in Q$
$\forall a\in U$

Alg.: O mașină Turing deterministă care se oprește pe fiecare intrare

Mașina Turing cu n benzi



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

$\delta:(Q\times U^n)\longrightarrow 2^{(Q\times(U\setminus B)^n\times\{R,L\}^n)}$

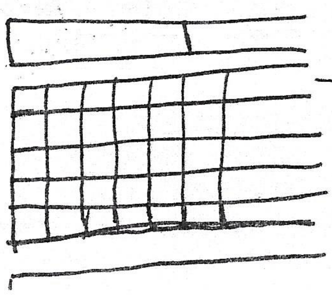
$(s,b_1,\dots,b_m,x_1,\dots,x_m)\in\delta(q,a_1,\dots,a_m)$

Teoremă: Pentru orice mașină Turing nedeterministă M, există o mT deterministă M' cu 3 benzi echivalentă cu M.

Dem:

M: 

$M=(Q,V,U,\delta,q_0,F,B)$ dată

M': 

$W=Q\times U\times Q\times(U\setminus B)\times\{R,L\}, \leq$ (relație de ordine)

$W=\{(w_1,w_2,w_3,w_4,w_5)|\dots\}$

W^* $\underbrace{(\ X\ )\cdots(\ )}_{\text{secv. de vectori}} \leq \underbrace{\qquad}_{\text{secv. de vectori}}$

ord lexicografică

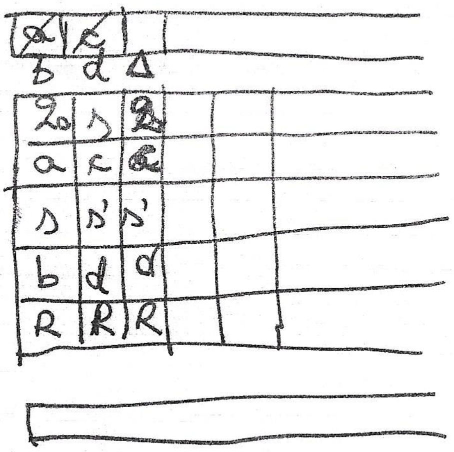
aa ≥ b (pt că 'b' e mai scurt decât 'aa')

aa ≤ bb

M': ① copiază w de pe prima bandă pe banda 3

② generează succesorul conținutului de pe banda 2 pe banda 2

③ Simulează calculul lui M dat de secvența de pe banda 2 pe intrarea de pe banda 1.



Pe banda 2 generez toate calculele posibile și imposibile și le verific (în ordine lexicografică)

- Dacă simularea se încheie cu succes într-o stare finală, atunci M' acceptă și se oprește.
- Altfel, se copiază banda 3 pe banda 1 și se reia pasul ②.

q_0, q $w = \{(q_0,a,q_0,a,R),(q_0,a,q_0,a,L),(q_0,a,q_0,b,R),\ldots\}$

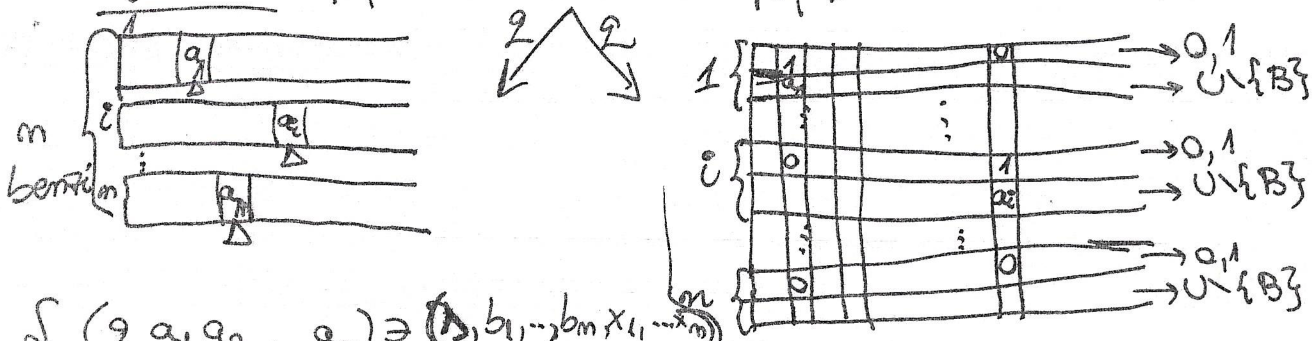
a, b

R, L

$$\begin{pmatrix} q_0 \\ a \\ q_0 \\ b \\ R \end{pmatrix}\begin{pmatrix} q_1 \\ b \\ q_0 \\ b \\ R \end{pmatrix}\begin{pmatrix} q_0 \\ a \\ q_0 \\ b \\ R \end{pmatrix}$$

<u>Teoremă</u>: Pentru orice mașină Turing M cu n benzi, există o mt M' cu o bandă echivalentă cu M. În plus, M' este deterministă dacă M este deterministă.

Dem: M   M':



$$\delta(q, a_1, a_2, \dots, a_m) \ni (p, b_1, \dots, b_m, x_1, \dots, x_m)$$

06.10.2017

Calculabilitate și complexitate
– Curs 1 –

Examen: 2 subiecte → teorie + problemă

Nota: 25% seminar
25% exercițiu examen
50% teorie examen

Pt promovare: Nota finală 5.

- Mașina Turing
- Programe standard
- Funcții recursive

(1) Mașina Turing

a_1	a_2	...	...	a_n	B	

Δ

q ↳ cap de citire/scriere
→ stare $\in Q$

Definiție: O mașină Turing este o structură

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

Q: mult. finită de stări
V: alfabetul de intrare
U: alfabetul de lucru (al benzii), $V \subseteq U$
$q_0 \in Q$: stare init.
$F \subseteq Q$: mult. stărilor finale
$B \in (U \setminus V)$: simbolul blank
$\delta : Q \times U \longrightarrow 2^{Q \times (U \setminus \{B\}) \times \{L, R\}}$

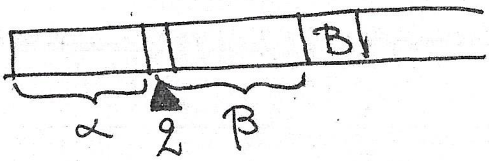
$\delta(q,a) \ni (s,b,R)$

mT $\rightarrow$ dispozitiv de acceptare (A)
$\quad\ \hookrightarrow$ dispozitiv calcul funcții (C)

<u>Configurație instantanee</u> (iD): $\hookrightarrow$ description

$\alpha q \beta$ unde

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



$\alpha q x \beta \vdash \alpha y s \beta$, dacă $\delta(q,x) \ni (s,y,R)$

$\alpha q \vdash \alpha y s$, dacă $\delta(q,B) \ni (s,y,R)$

$\alpha x q y \beta \vdash \alpha s x z \beta$, dacă $\delta(q,y) \ni (s,z,L)$

$\alpha x q \vdash \alpha s x z$; $\delta(q,B) \ni (s,z,R)$

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

$\boxed{f : \mathbb{N}^k \longrightarrow \mathbb{N}}$ $\qquad (a_1, \dots, a_n) \rightsquigarrow \underbrace{2^{a_1} \cdot 3^{a_2} \cdot \dots \cdot p_n^{a_n}}_{x}$; $p_n \rightarrow$ al n-lea număr prim

Dacă $x = 2^5 \cdot 7^2 \Rightarrow (5,0,0,2)$

$\boxed{\text{Ex:}}$ $f(x_1, \dots, x_k)$; $\overline{x} = \underbrace{11\cdots1}_{x+1 \text{ ori}}$

$\underbrace{1|1|\cdots|1}_{\overline{x_1}}|0|\underbrace{1|\cdots|1}_{\overline{x_2}}|0|\underbrace{1|\cdots|1}_{\overline{x_k}}|B$

($f(x_1 \dots x_k)$ nu e def $\Leftrightarrow$ mT nu se oprește)

$\underbrace{1|\cdots|1}_{\overline{y} = f(x_1, \dots, x_m)}|0|0|\cdots|0|B$