Curs 1. Calculabilitate și Complexitate.

Calculabilitate: programe standard
mașini Turing - funcții Turing calc
funcții recursive
- echivalența celor 3 modele.
- Gödelizare (codificare) a PS
- Problema opririi MT este nedecidabilă

Complexitate: - modelele de MT
- clase de complexitate
- relații între aceste clase.
- NP - completitudine.

$x=(x_1,\dots,x_n)$
$y=(y_1,\dots,y_n)$ y_i, x_i secvențe binare.

$\exists\, k\in\mathbb{N}$ a.î. $x_{i_1} x_{i_2}\dots x_{i_k} = y_{i_1} y_{i_2}\dots y_{i_k}$

Demonstrații constructive

Af. $x = 0,x_1x_2x_3\dots x_n\dots$

~~Există un număr $y\in\{0,\dots,9\}$ a.î. y apare de o infinitate de ori în scrierea lui x.~~

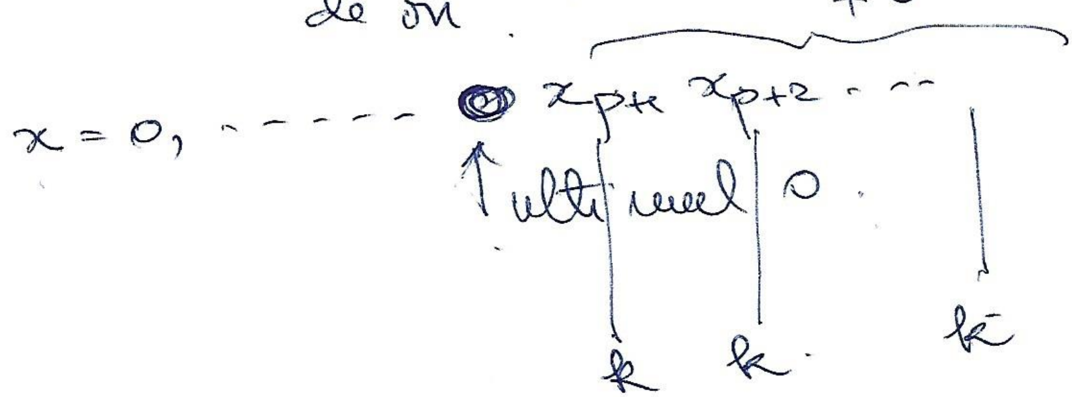
$(p_n)_{n\geq 1}$.

$x_1\neq 0$	$p_1=1$
$x_2\neq 0$	$p_2=1$
$x_3\neq 0$	$p_3=1$.
$x_4=0$	$p_4=0$.
$x_5=0$	$p_5=0$.

Există un k ce apare de o infinitate de ori în șirul p_n.

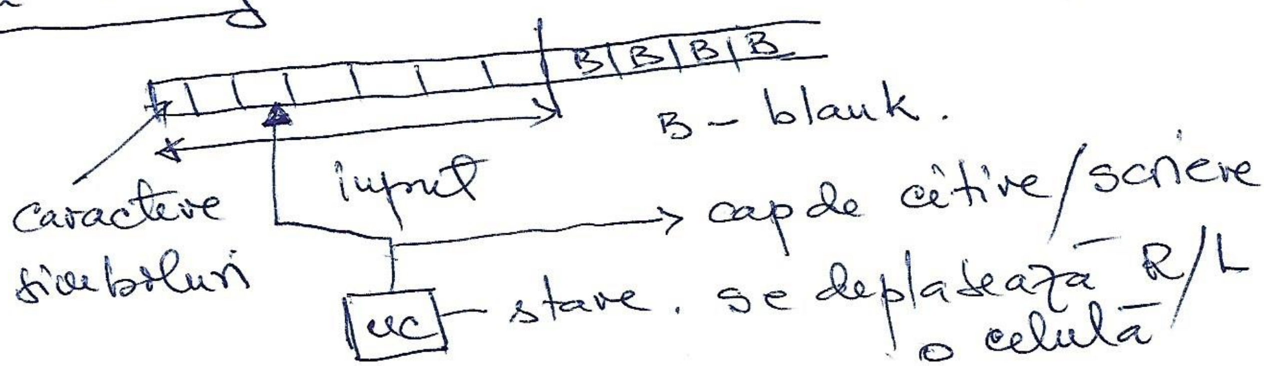
Dem. Caz 1. Scrierea zecimală a lui x are o infinitate de 0. → $k=0$

Caz 2. 0 nu apare de o infinitate de ori.

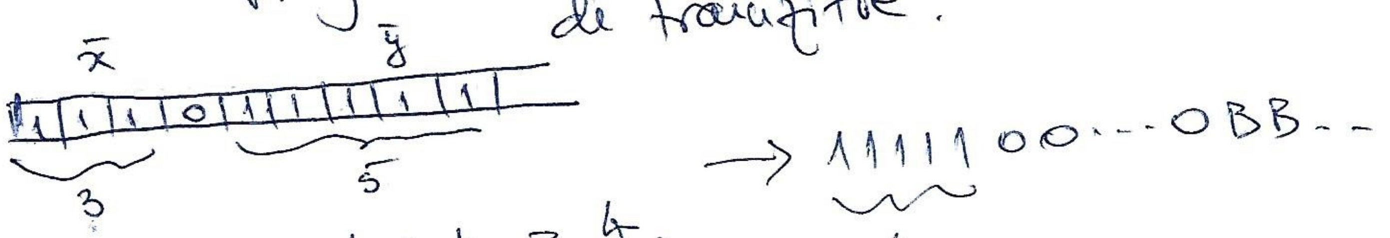


$x = \sqrt{2} - 1$.

Mașina Turing → infinită



Program: funcție de tranziție.



$f(2,4) = 2\log_2 4 = 4$.

$f(0,4)$ 1011111

Alan Turing   1350 + 2015 = 22

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

- Q – m. stărilor finită
- V – alfabet intrare finit.
- U – alfabetul benzii $V \subseteq U$ finit.
- $q_0 \in Q$ starea inițială
- $B \in U \setminus V$, blank
- $F \subseteq Q$, $F \neq \emptyset$ stări finale.

$$\delta : (Q \setminus F) \times U \longrightarrow \mathcal{P}_f(Q \times U \times \{R, L\}).$$

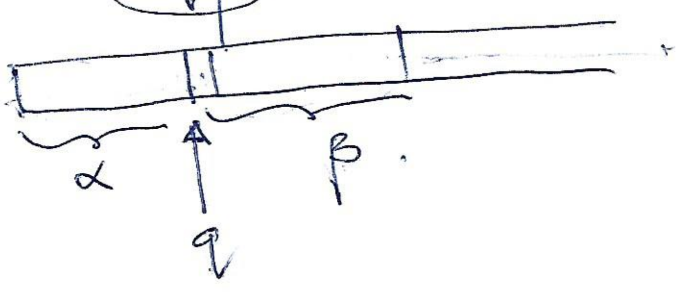
$$(s, b, R) \in \delta(q, a)$$

Mașina este în starea "q" și citește "a", poate trece în starea "s", scrie "b" peste "a" și se deplasează către dreapta.

	a	b	c
s	(.,.) (.,.)		
q		(.,.,.) (.,.,.)	
q_0		$\emptyset$	
f			

Restricții 1) $(s, b, X) \in \delta(q, a)$, $a \neq B$ $\Rightarrow b \neq B$.

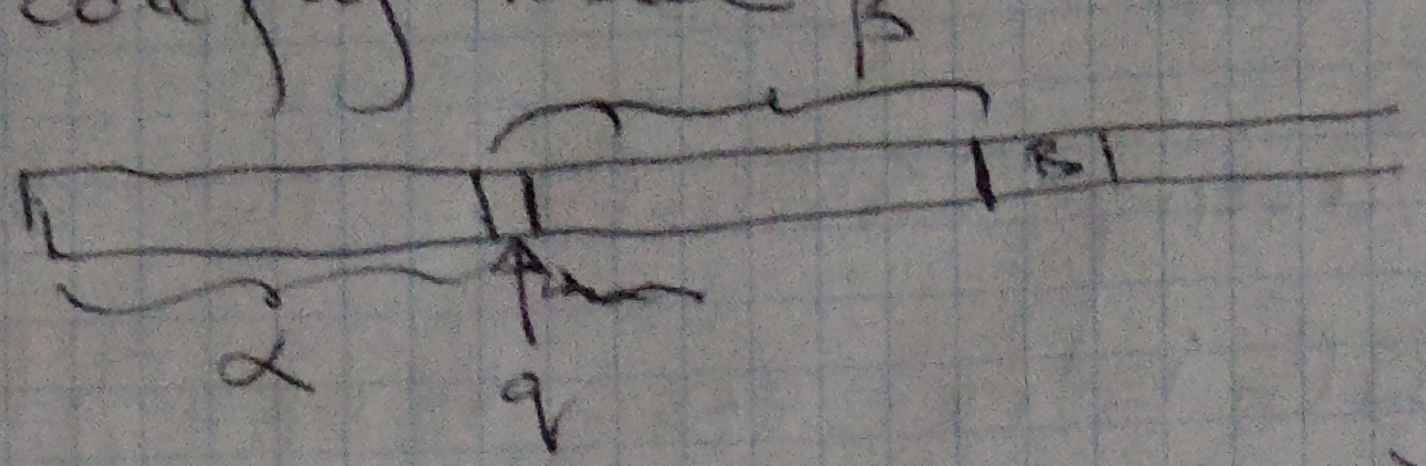
Tranziții   Configurație (ID): $\alpha q \beta$.



Ex. $q\beta$
αq

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

$\alpha q \beta$ – configurație



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

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

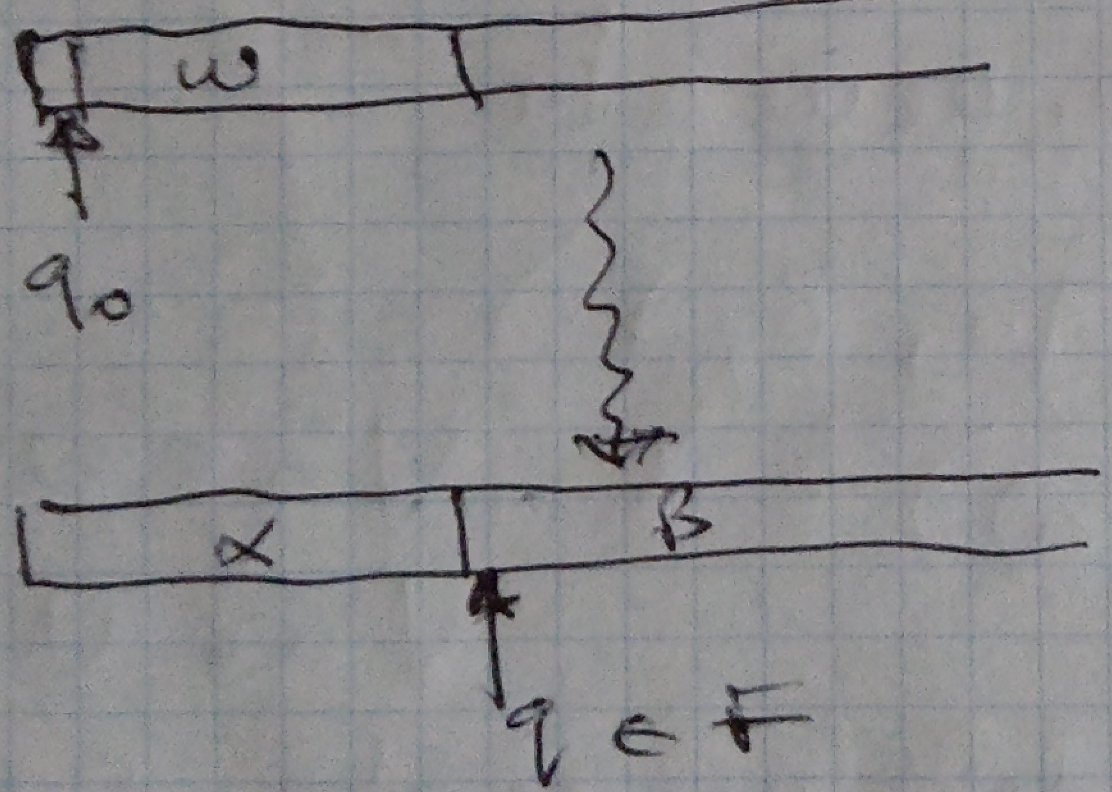
$\alpha a q b \beta \vdash \alpha s a c \beta$, dacă $(s, c, L) \in \delta(q, b)$

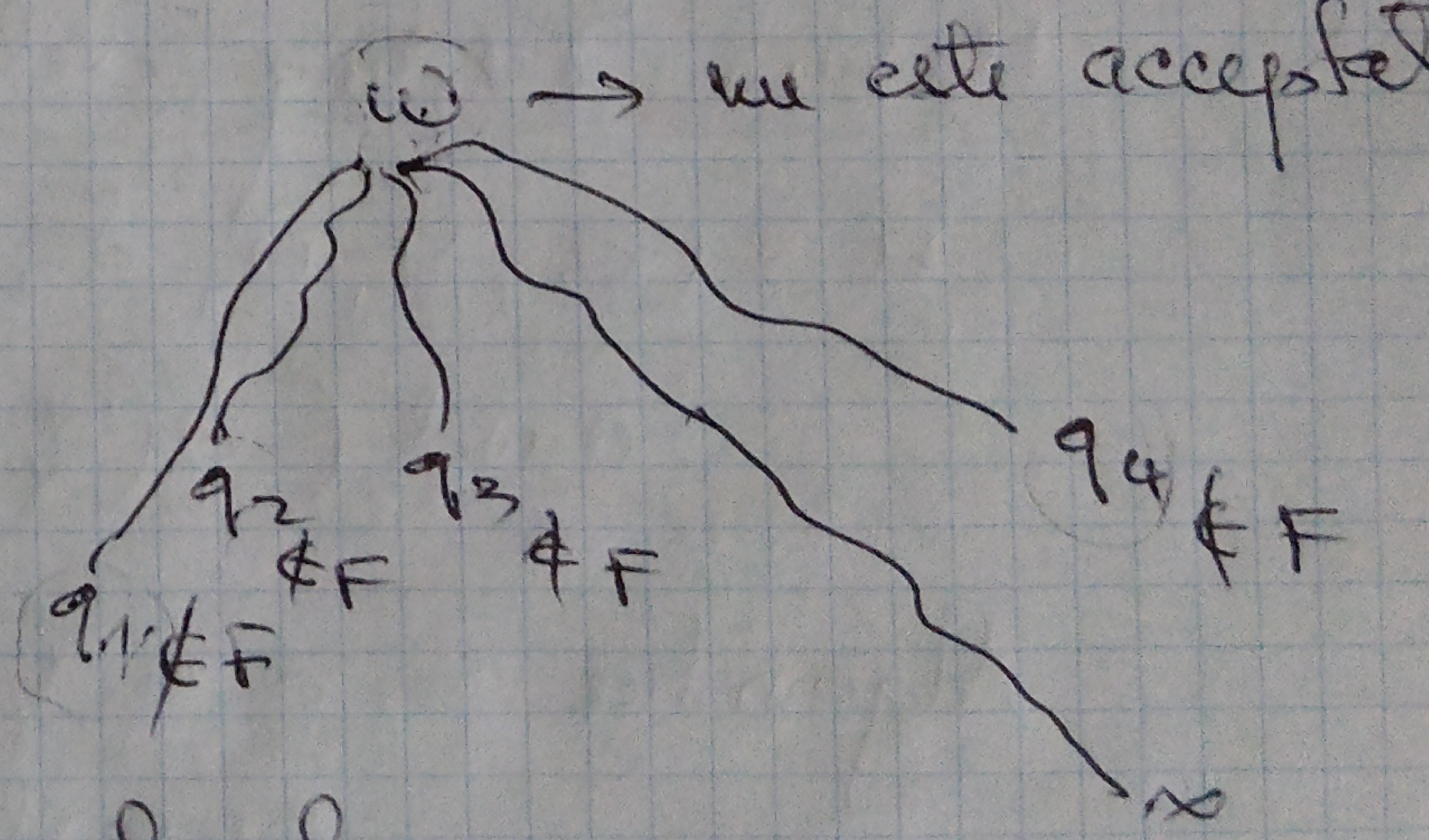
$\alpha a q \vdash \alpha s a b$, dacă $(s, b, L) \in \delta(q, B)$

M – dispozitiv de acceptare
(acceptă limbaje)

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

$\downarrow$ închiderea reflexivă și tranzitivă a relației $\vdash$

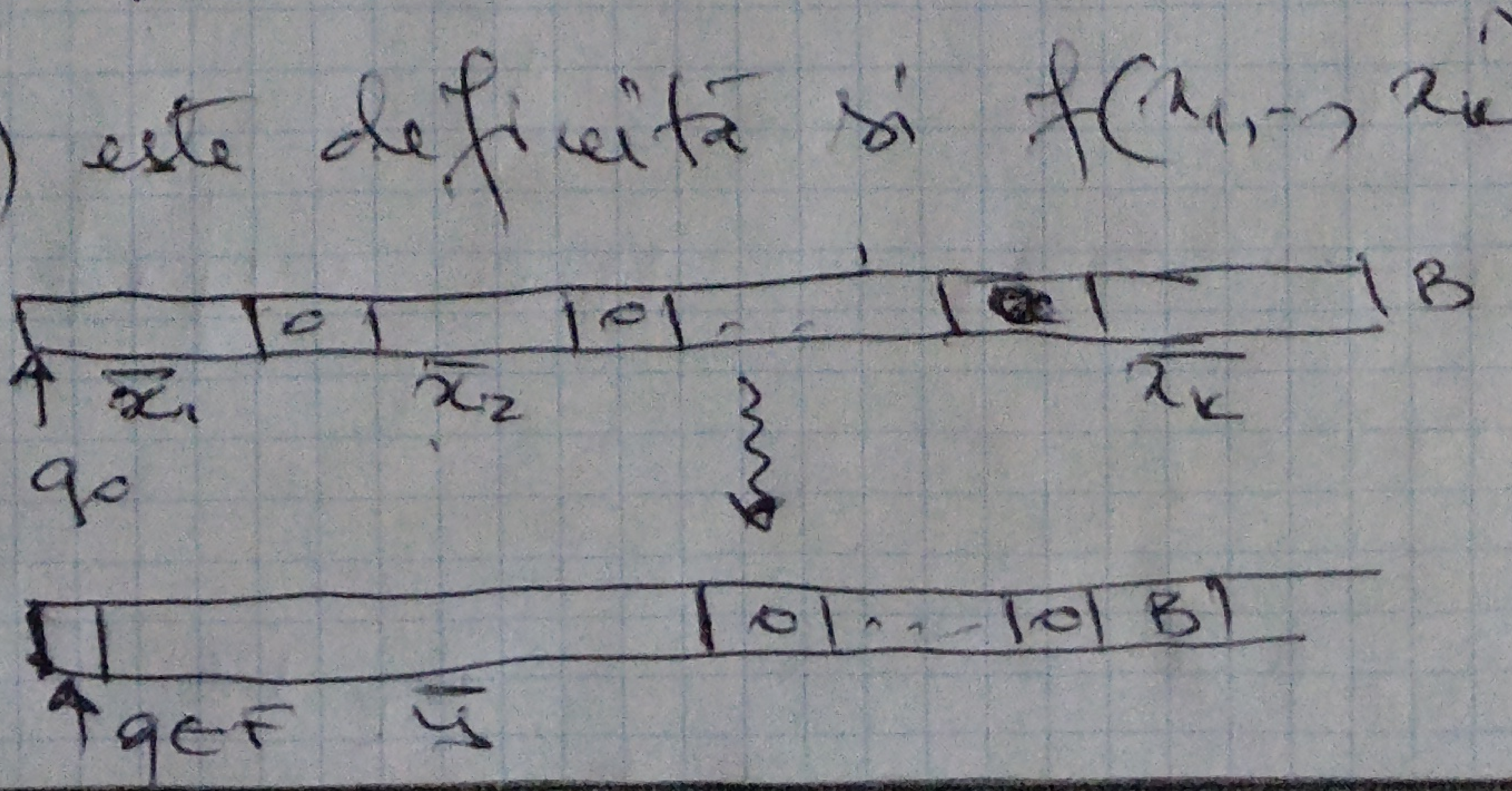




M ca dispozitiv de calcul
funcții naturale

$f : N^k \overset{o}{\longrightarrow} N$

$f(x_1, \dots, x_k)$ este definită și $f(x_1, \dots, x_k) = y$
ddacă



$\overline{x} = \underbrace{1 \dots 1}_{x+1 \text{ ori}}$

$f(x_1, \dots, x_n)$ nu este definită dacă mașina Turing nu se oprește pe intrarea $(x_1, \dots, x_n)$

f s.n. Turing calculabilă dacă există o MT folosită ca dispozitiv de calcul care o calculează.

Construiește o MT:
1) Etapele soluției
2) Fiecare etapă se poate executa pe o MT.
   - Routine cunoscute definite (pseudocod)
   - numărul de stări și simboluri auxiliare folosite este FINIT.

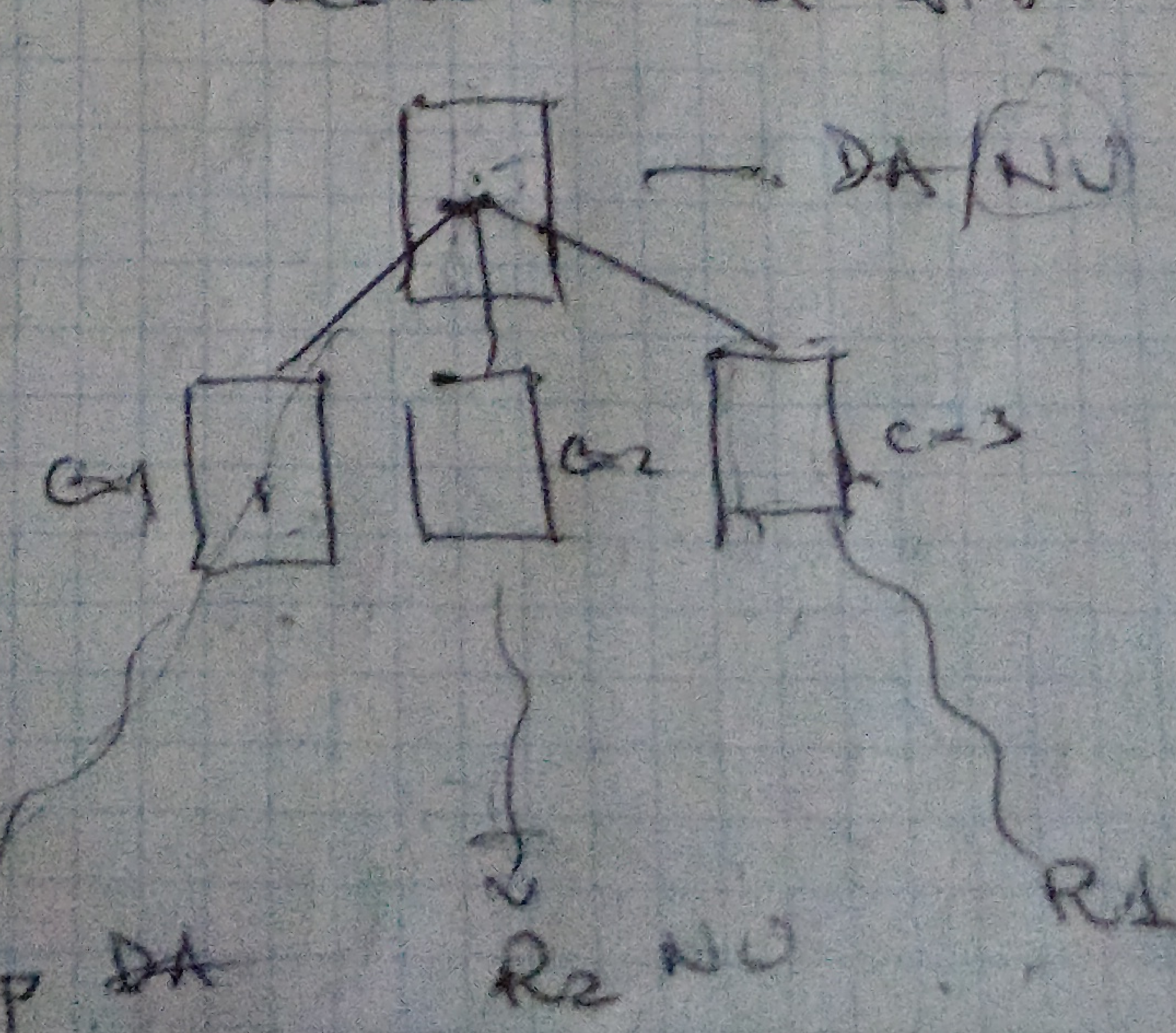
Mașină Turing deterministă

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

$$\mathrm{card}(\delta(q, a)) \le 1 \quad \forall q \in Q, \ \forall a \in U$$

Algoritm = o MT deterministă care se oprește pe fiecare intrare.

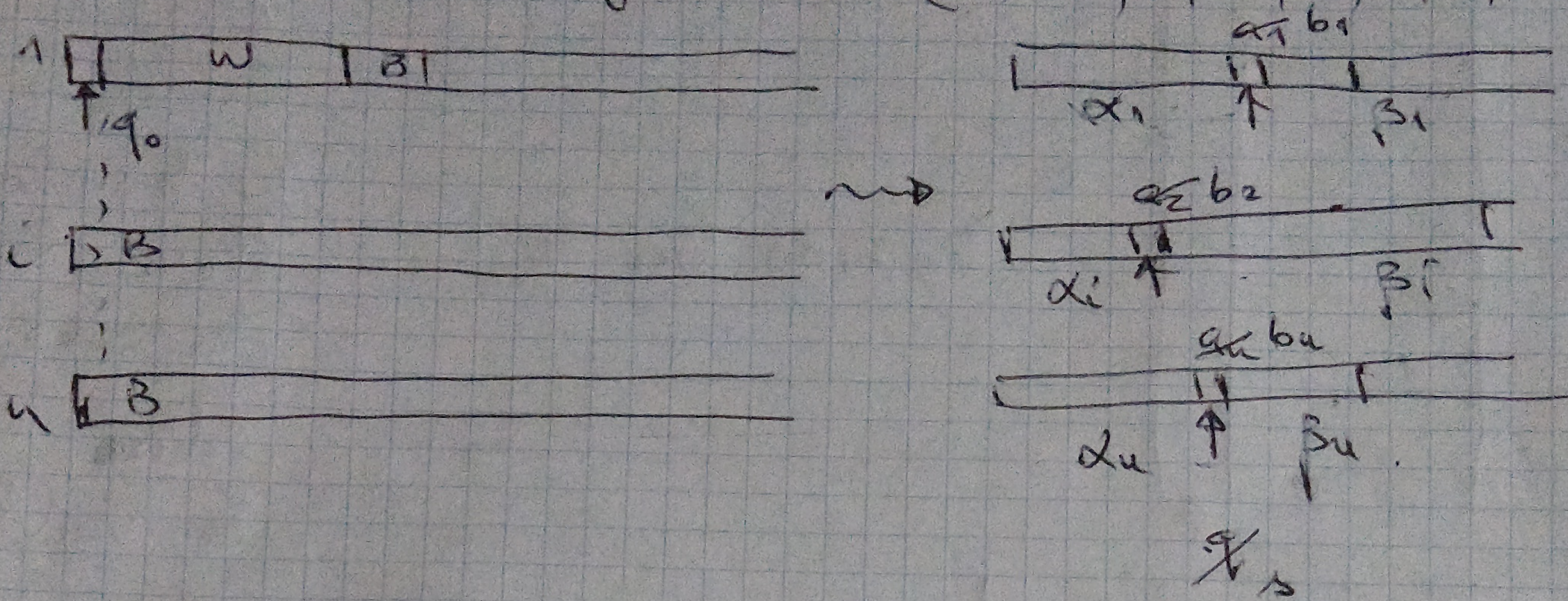
MT deterministă $\stackrel{?}{=}$ MT nedeterministă



Scop: $MTD \equiv MTN$

$nMTD$

O MT cu n benzi: $M = (n, Q, V, U, \delta, q_0, B, F)$



$$\delta : (Q \times U^n) \setminus F \longrightarrow \mathcal{P}_f(Q \times U^n \times \{L, R\}^n)$$

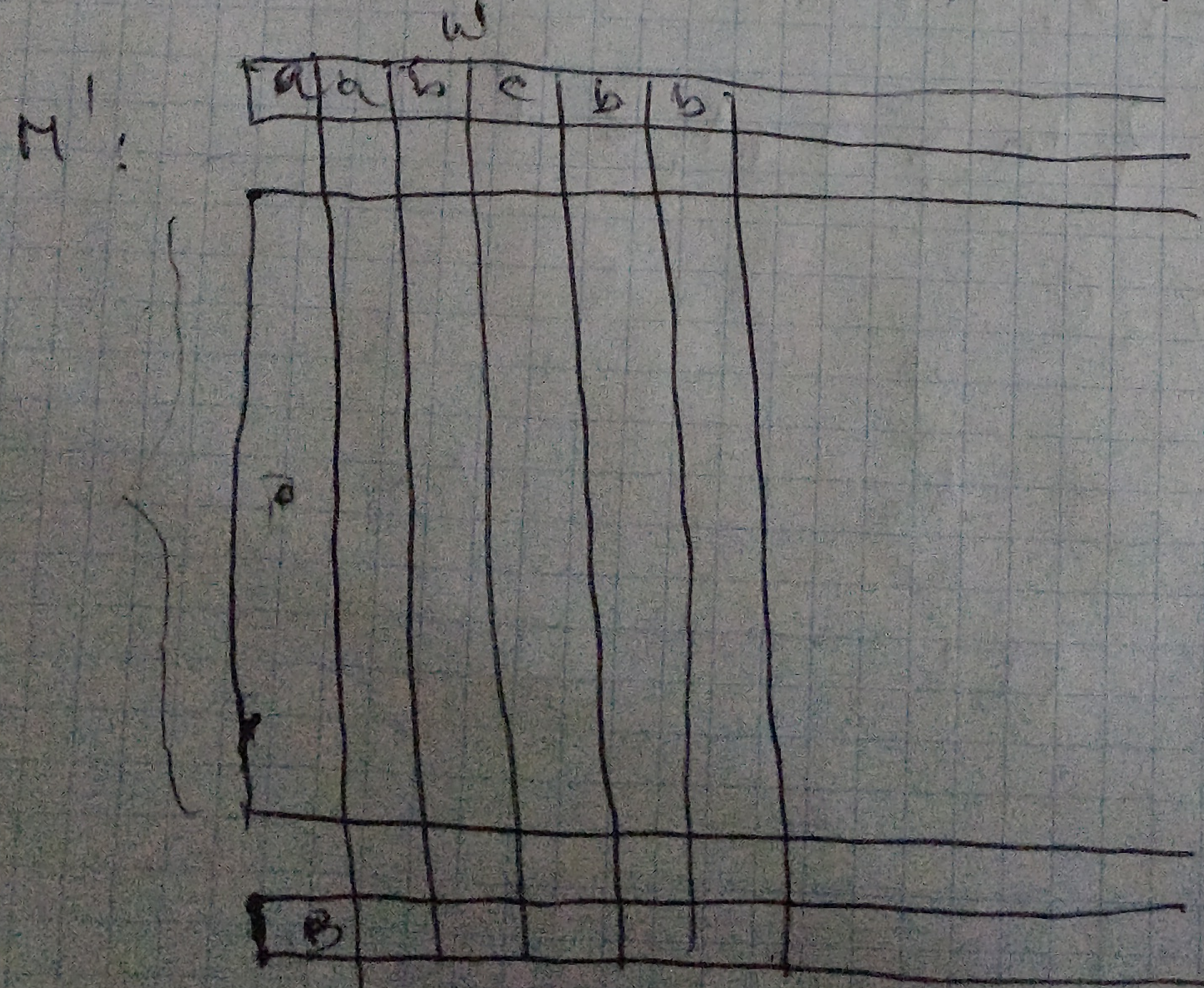
$$(s, b_1, \dots, b_n, \underbrace{X_1, \dots, X_n}_{\in \{L, R\}}) \in \delta(q, a_1, \dots, a_n)$$

<u>Teoremă</u> Pentru orice MT nedeterministă M, <u>există</u> o MT deterministă, cu 3 benzi, M' a.î. $L(M) = L(M')$.

<u>Dem</u> $M = (Q, V, U, \delta, q_0, B, F)$

$\Downarrow ?$

$M' = (3, Q', V, U', \delta', q_0, B, F')$

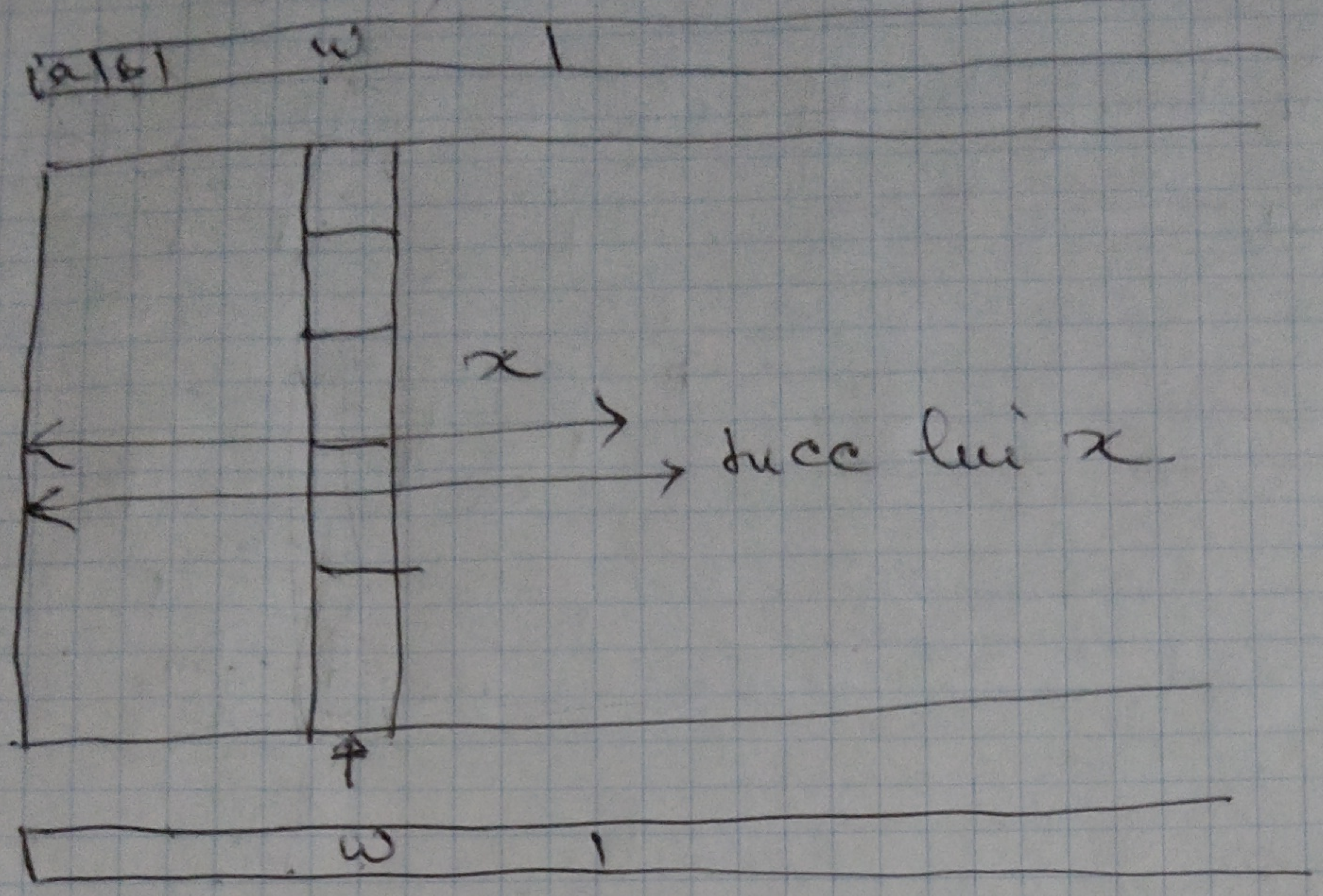


<u>Etapa 1</u>

Copiază w de pe prima bandă pe banda 3

$\delta(q_0, (a, , B)) = (q_1, a, -, a, R, R)$

$\delta(q_0, a, B) = (q_0, a, a, R, R)$
$\delta(q_0, b, B) = (q_0, b, b, R, R)$
$\delta(q_0, c, B) = (q_0, c, c, R, R)$



Etapa 2

Ignoră banda 1.

Generez pe banda 2 elementul succesor al celui curent în mulțimea E.

$$E = \{(q, a, s, b, x) \mid q, s \in Q, a \in U, b \in U \setminus \{B\}, x \in \{L, R\}, (s, b, x) \in \delta(q, a)\}$$

$$\subseteq Q \times U \times Q \times U \setminus \{B\} \times \{L, R\} \cup \{(B, \dots, B)\}$$

Fiecare element al lui E va fi considerat un simbol pe banda 2

$E = \{s_1, s_2, \dots, s_t\}$ ca urmare a definiției $(E, \leq)$

Ex $S = \{1, 2, 5, 10, 4\}$

$(S, \leq) = \{10, 5, 4, 1, 2\}$

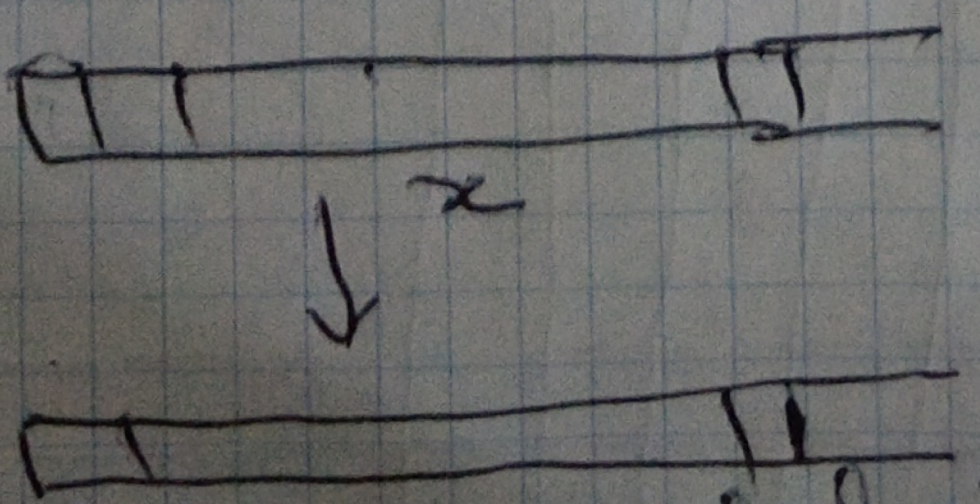
$E = \{(\quad), (\quad), (\quad), \dots, (\quad)\}$

$E = \{s_1, s_2, s_3, \dots, s_t\}$

n

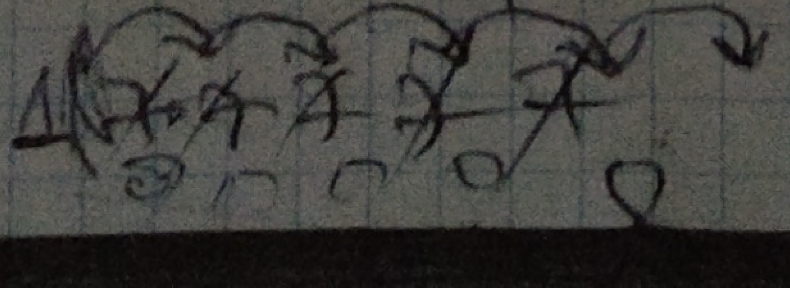
B

Subproblemă



$21704B$

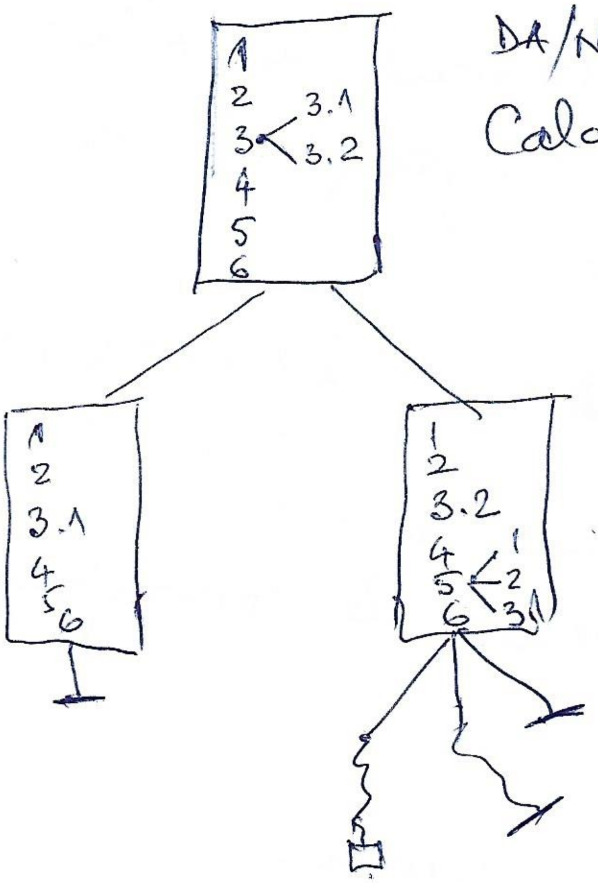
4 0

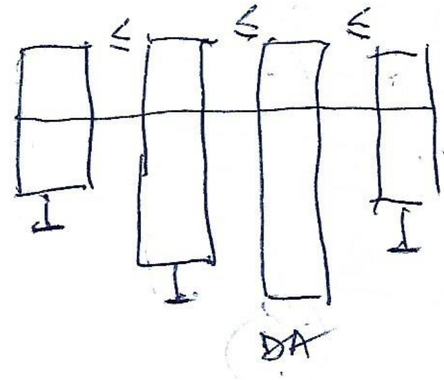


= CURS 3 =

DA/NU

Calcul (Output)





Bactracking

Masina Turing

Secventa

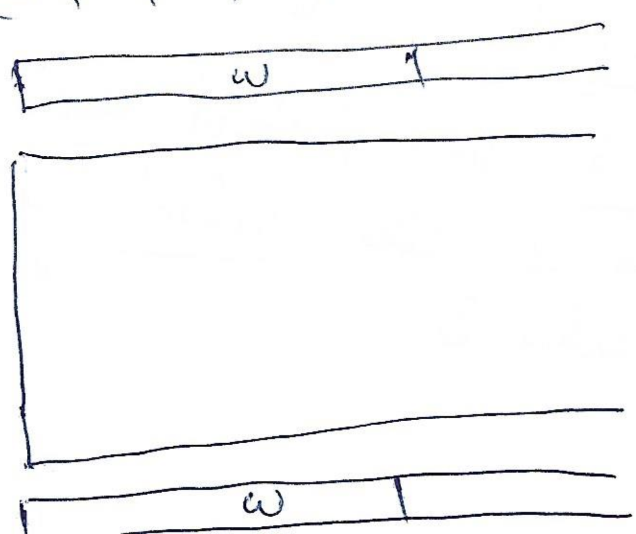
$(q_1, a_1, s_1, b_1, x_1)(q_2, a_2, s_2, b_2, x_2)$

$\ldots (q_n, a_n, s_n, b_n, x_n)$

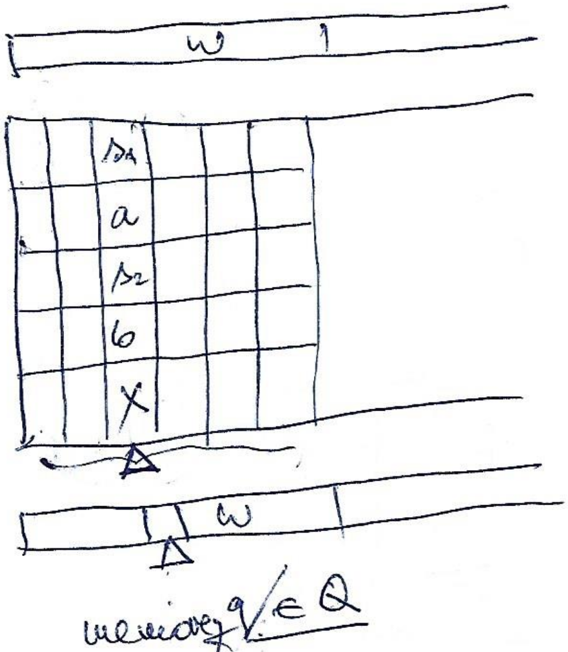
Teorema Pentru orice MT M nedeterminista exista o MT M' determinista cu 3 benzi a.i. $L(M) = L(M')$.

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

Construim M'



Etapa 1:
Copiem w de pe banda 1 pe banda 3



Etape 2.

Pe banda 2 se generează succesorul în ordine lexicografică în E^*

$$E = \underbrace{Q \times U \times Q \times U \times \{L, R\}}_{(q, a, s, b, X)}$$

E^* = mulțimea secvențelor de cvintupluri din E

$(E, \leq)$

$\Sigma = \{a, b\}$ $a \leq b$.   abb $<$ aaaaa

aabba $\geq$ aaaba

Pe ce poziție se situează b?

~~a, aa, aaa,~~

a, b, aa, ab, ba, bb, aaa, ...

aababb, $n = 173$.

Etapa 3 Verificăm dacă calculul de pe banda 2 este valid pe intrarea aflată pe banda 3.

E3.1. Calculul se blochează

3.2. Calculul se termină dar starea nu este finală (ultima memorată)

3.3. Calculul se termină și starea este finală. (Acceptă w).

$$\Updownarrow$$

M acceptă w. $\quad L(M) = L(M')$

E3.1 și E3.2: GOTO Etapa 1. QED.

Mașina M' este deterministă.

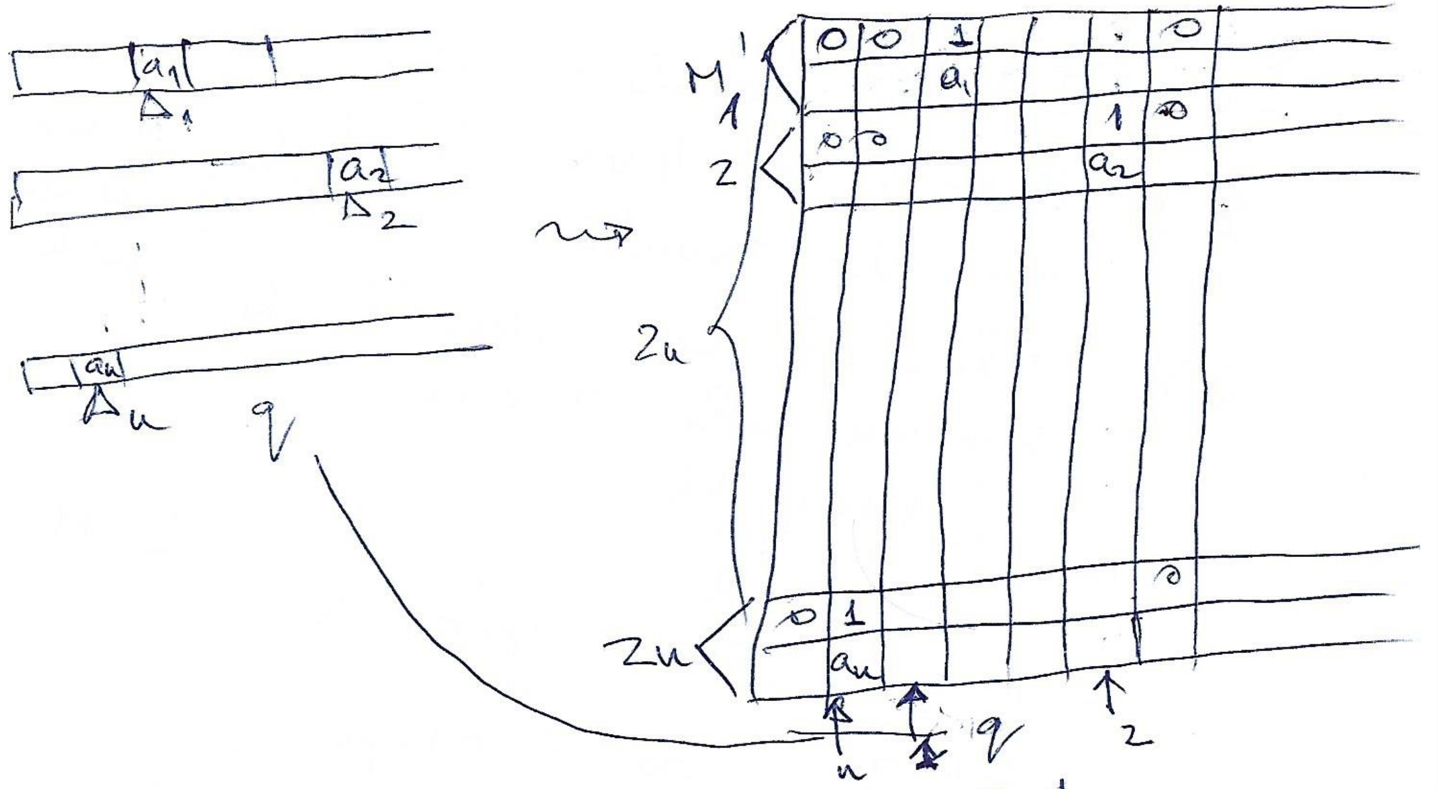
$$\begin{tikzcd} ?\,MTN \arrow[r] & \exists MTD \arrow[dl, "?"] \\ MTD \arrow[u] & \end{tikzcd}$$

<u>Teorema 2</u> Pentru orice MT cu n benzi, M, există o MT, M', cu o bandă a.î. $L(M) = L(M')$. În plus, M' este deterministă dacă M este deterministă.

<u>Dem.</u> Fie $M = (n, Q, V, U, \delta, q_0, B, F)$

$$\Downarrow$$

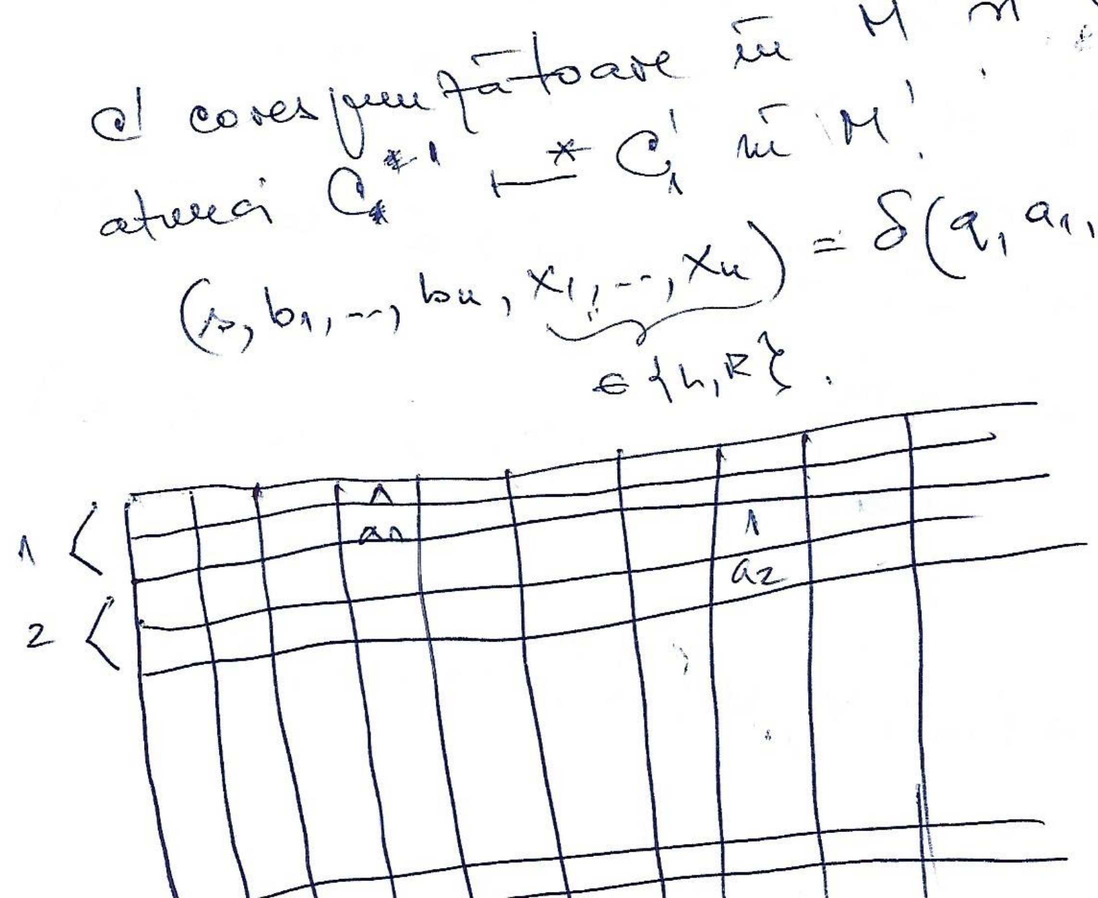
$$M' = (Q', V, U', \delta', q_0, B, F)$$



- Pe pistele de ordin impar am 0, 1.
- Pe pistele de ordin par am simbolurile din M.

Dacă o configurație din M are o configurație C corespunzătoare în M' și $C_* \vdash C_1$ atunci $C_*' \vdash^* C_1'$ în M'.

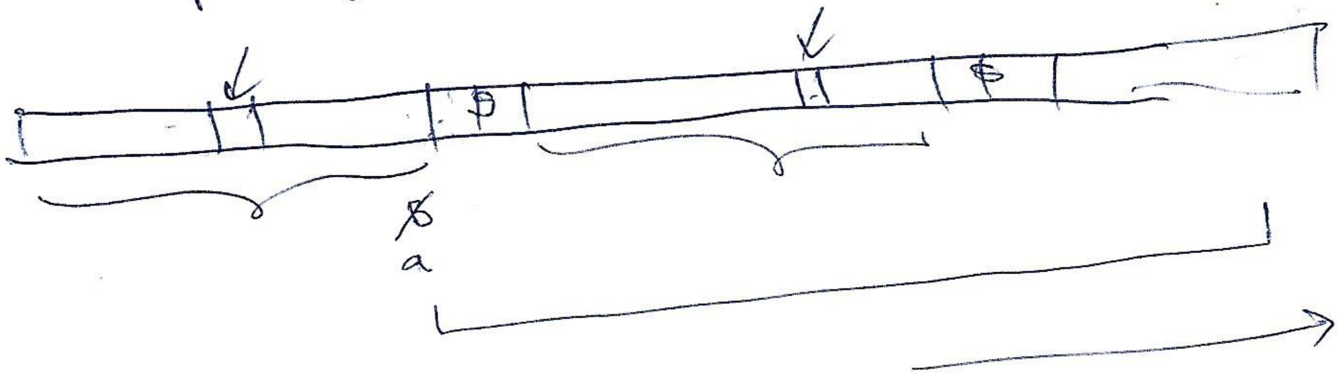
$(p, b_1, \dots, b_n, \underbrace{x_1, \dots, x_n}_{\in\{L, R\}}) = \delta(q, a_1, \dots, a_n)$



Etapa 1 M' scanează întreaga bandă și identifică simbolurile $a_1, a_2, \dots, a_n$ citite de M.

Etapa 2 M' scanează banda de la dreapta la stânga și actualizează simbolurile $a_1, a_2, \dots, a_n$ cu $b_1, \dots, b_n$.

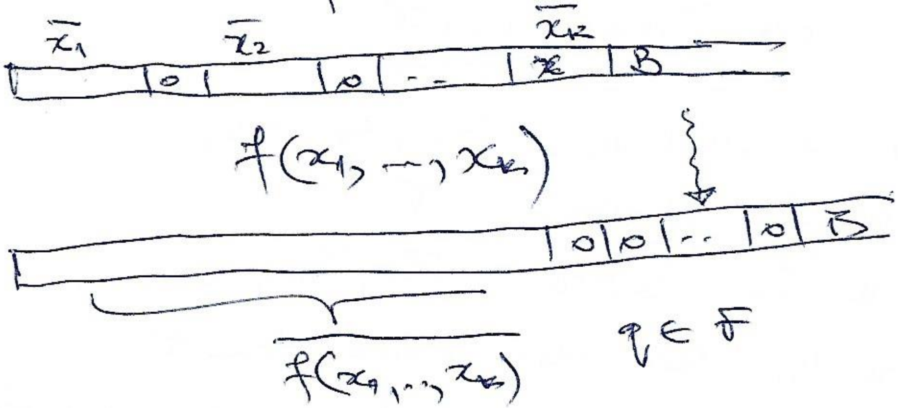
Etapa 3 M' scanează banda de la stg la dreapta și actualizează pozițiile 1 pe pistele impare.



= CURS 4 =

MT ca dispozitiv de calcul al funcţiilor

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



$\iff f(x_1, \dots, x_k)$ este definită

~~Dacă~~ $f(x_1, \dots, x_k)$ nu este definită $\iff$ M nu se opreşte pe intrarea $x_1, \dots, x_k$

Discuţie: $f : \mathbb{N}^k \to \mathbb{N}$ restrictiv?

(i) $f : \mathbb{N}^k \to \mathbb{N}^t$ $\quad f(x_1, \dots, x_k) = \underbrace{(v_1, \dots, v_t)}$

$$\downarrow$$

$$\underbrace{2^{v_1} \cdot 3^{v_2} \cdots p_t^{v_t}}_{\in \mathbb{N}}$$

(ii) $f : \mathbb{Z}^k \to \mathbb{N}$

$\downarrow$

$\mathbb{N}^{2k}$ $\quad (x_1, \dots, x_k) \in \mathbb{Z}^k$

$$\downarrow$$

$$(\substack{1\\0}, |x_1|, \substack{1\\0}, |x_2|, \dots, \substack{1\\0}, |x_k|)$$

(iii) $f : \mathbb{Q}^k \to \mathbb{N}$

$\downarrow$

$(\underbrace{\mathbb{Z} \times \mathbb{Z}}_{\mathbb{N}^3})^k$

(iv) $f : \mathbb{R}^k \to \mathbb{N}$

Ce funcții putem calcula cu MT?
, sunt Turing calculabile?

Limbaj de programare abstract

Sintaxa: { variabile de intrare: $X_1, X_2, X_3, \dots$
variabila de ieșire: Y
variabile de lucru: $Z_1, Z_2, Z_3, \dots$

Variabilele de lucru și cea de ieșire sunt inițializate cu 0. Orice variabila memorează o valoare naturala

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

Instrucțiuni (1) $V \leftarrow V$, unde V este o variabilă

Efect: Nul.

(2) $V \leftarrow V+1$ Efect

(3) $V \leftarrow V \dot{-} 1$ Efect: decrementare dacă val lui $V > 0$
Nul, altfel.

Obs Orice instrucțiune poate fi etichetată.

(4) IF $V \neq 0$ GOTO L

Efect: Dacă val lui V este 0 se trece la instrucțiunea următoare Altfel, se face transfer la prima instrucțiune cu eticheta L. Dacă nu există instrucțiune cu eticheta L, programul se oprește.

Program standard : o secvență finită de instrucțiuni.

Oprirea unui program standard:

- se termină instrucțiunile
- transfer la o instrucțiune care ~~a etic~~ nu există. (Vezi instrucțiunea IF $V \neq 0$ GOTO L)
- transfer la eticheta E (Exit)

Cum calculează ? $f : \mathbb{N}^k \to \mathbb{N}$

$f(x_1, \dots, x_k)$

$X_1 \leftarrow x_1$
$X_2 \leftarrow x_2$
$\dots$
$X_k \leftarrow x_k$

} nu apar în program. (sunt inițializate)

$f(x_1, \dots, x_k)$ este definită $\Leftrightarrow$ Programul se oprește și valoarea lui y este $f(x_1, \dots, x_k)$.

$f(x_1, \dots, x_k)$ nu este definită $\Leftrightarrow$ Programul nu se oprește pe intrarea $X_1, \dots, X_k$.

Ex 1. $f(x) = x$.

A_2: IF $X_1 \neq 0$ GOTO A_1
$Z_1 \leftarrow Z_1 + 1$
IF $Z_1 \neq 0$ GOTO E
A_1: $X_1 \leftarrow X \dot{-} 1$
$Y \leftarrow Y + 1$
IF $X \neq 0$ GOTO A_2.

Obs 1) GOTO L (Macro instrucțiune)
2) $V \leftarrow V'$ (— " —)

A_2: IF $X_1 \neq 0$ GOTO A_1

GOTO ~~A_3~~ E

A_1: $X_1 \leftarrow X_1 \dot{-} 1$

$Y \leftarrow Y+1$

$Z_2 \leftarrow Z_2+1$.

IF $X_1 \neq 0$ GOTO A_1

A_3: $X_1 \leftarrow X_1+1$

$Z_2 \leftarrow Z_2 \dot{-} 1$.

IF $Z_2 \neq 0$ GOTO A_3

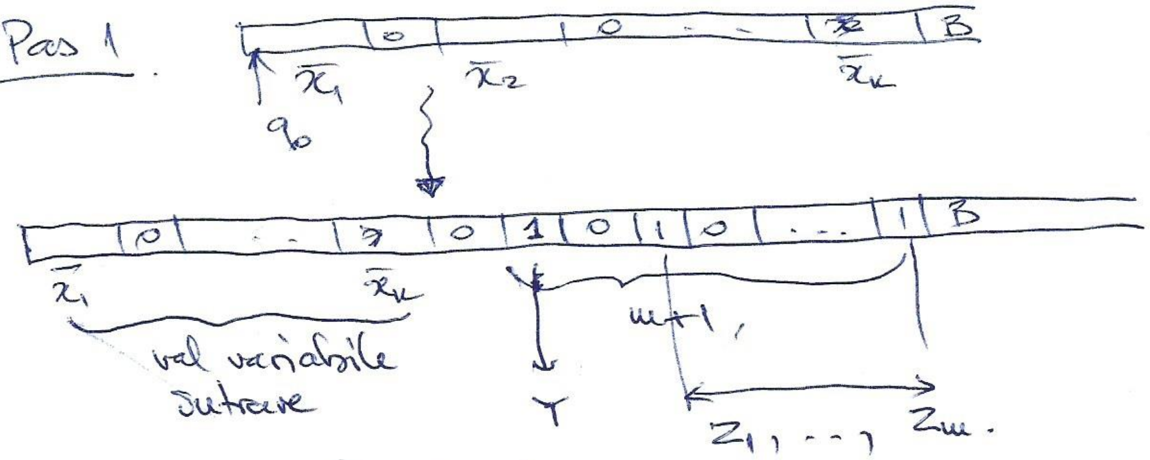
(T) Funcție calculabilă cu PS. $\Rightarrow$ Funcție Turing calculabilă.

Dem. Fie $f: \mathbb{N}^k \to \mathbb{N}$ o funcție calculabilă cu PS.
Fie P programul standard care calculează f.

P < n instrucțiuni: $I_1, I_2, \dots, I_n$.
$Z_1, \dots, Z_m$ variabile de lucru.

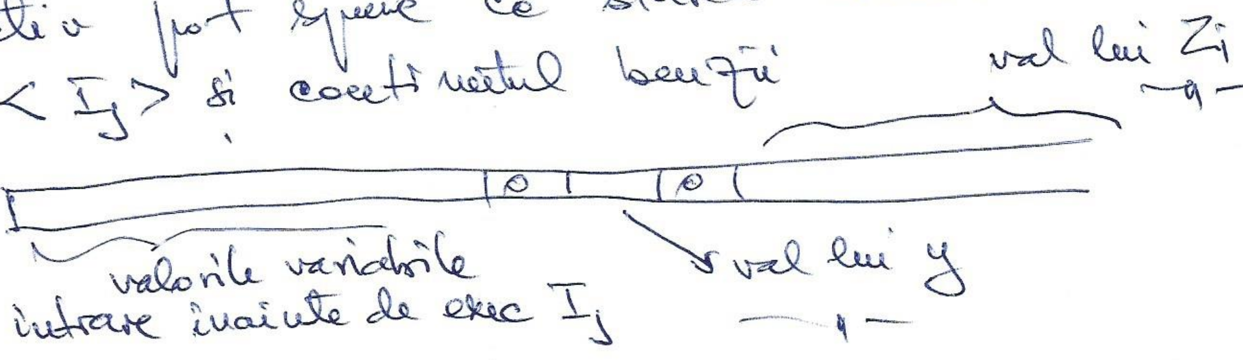
Cum calculează M o MT:

Pas 1.



Starea va fi $\langle I_1 \rangle$

Inductiv pot spune că starea curentă a MT este $\langle I_j \rangle$ și conținutul benzii



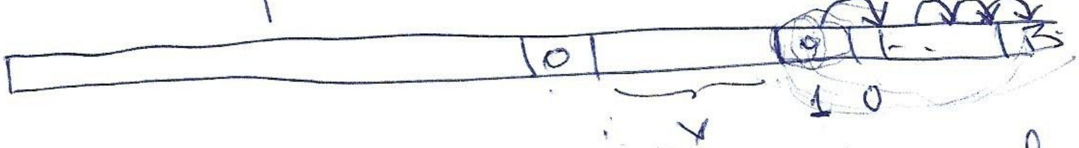
<u>Pas 2</u>. $I_j : X \leftarrow V$.

MT nu efectuează nicio modificare a benzii și schimbă starea $\langle I_{j+1} \rangle$.

$I_j : V \leftarrow V+1$.

(i) MT identifică porțiunea de bandă corespunzătoare lui V

(ii) Deplasează la dreapta toate porțiunile de la dreapta zonei coresp. lui V.



(iii) ~~...~~ Mărește porțiunea coresp lui V cu încă un 1.

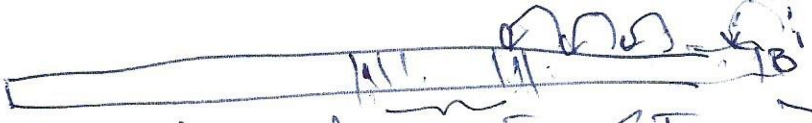
(iv) Schimbă starea în $\langle I_{j+1} \rangle$.

$I_j : V \leftarrow V \dot{-} 1$.

(i) idem (i) anterior.

(ii) Dacă porțiunea coresp lui V are doar un 1, schimbă starea în $\langle I_{j+1} \rangle$.

(iii) Altfel, șterge un 1 și compactează porțiunea lui V.



schimbă starea în $\langle I_{j+1} \rangle$.

101 ⟶ 101

1111

I_j : IF $V \neq 0$ GOTO L.

(i) Idem (i) anterior.

(ii) Idem (ii) anterior.

(iii) Altfel. lasă banda neschimbată.

<u>starea curentă</u> ⟶ $\langle I_p \rangle$ unde p este cel mai mic indice cu instrucțiunea I_p având eticheta L.

dacă L este o etichetă, diferită de E și care există în P.

Dacă L = E sau nu există starea devine $\langle I_{n+1} \rangle$ / eticheta L,

Mulțimea stări finale: $\{\langle I_{n+1} \rangle\}$.
intermediare.

<u>Pas final (3)</u>: Șterge tot de pe bandă cu excepția lui y;

Întrebare: Există funcții $f: \mathbb{N} \to \mathbb{N}$ care nu pot fi calculate cu PS?

$$\text{card}(\{f: \mathbb{N} \to \mathbb{N}\}) = \aleph_0^{\aleph_0}$$
$$\text{card}(\{\text{toate PS}\}) = \aleph_0$$