SUBIECTUL 2 :

Funcții recursive :

- funcții elementare :

$succ(n) = n+1$ (funcția succesor), $succ : N \to N$

$C_k^{(m)}(x_1, \dots, x_m) = k$ (funcția constantă), $C_k^{(m)} : N^m \to N$

$\pi_k^{(n)}(x_1, \dots, x_n) = x_k$ (funcția proiecție), $\pi_k^{(n)} : N^n \to N$

- operații :

1) compunerea funcțională : $g : N^k \to N$, $i = \overline{1,m}$

$h : N^m \to N$,

$f : N^k \to N$

f se obține prin compunerea funcțiilor din h și g;

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

2) recurența primitivă :

$f : N^{k+1} \to N$ se obține prin recurență primitivă din funcțiile $g : N^k \to N$ și $h : N^{k+2} \to N$, dacă :

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

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

3) minimizarea nemărginită :

$f : N^k \to N$ se obține din funcția $g : N^{k+1} \to N$, prin minimizare nemărginită dacă :

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

Funcții Turing calculabile:

Se numesc funcții Turing calculabile, funcțiile pentru care se poate scrie o mașină Turing.

Programe standard

Limbajul abstract $\mathcal{L}$ va calcula $f: N^k \to N$.

Variabile
- de intrare: $x_1, \dots, x_n$
- de ieșire: y
- locale: $z_1, \dots, z_n$;

} sunt inițializate cu 0

Valorile tuturor variabilelor la orice moment sunt numere naturale.

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

E → exit

Instrucțiuni:

- $V \leftarrow V$ (efect nul)
- $V \leftarrow V + 1$ (incrementare)
- $V \leftarrow V \dot{-} 1$ (efect de decrementare dacă $V \neq 0$, nul altfel)
- IF $V \neq 0$ GO TO L:
  - → $V = 0$ efect nul, se trece la următoarea instrucțiune
  - → $V > 0$ transferul se face astfel:

- se trece la instrucțiunea cu eticheta L, dacă $L \neq E$ și dacă există;

- se termină programul dacă $L = E$ sau L nu există.

Programul standard este format dintr-un set de instrucțiuni, iar terminarea se face, fie prin salt la eticheta E, fie prin salt la o etichetă inexistentă, fie transferul se face la sfârșitul instrucțiunii.

Outputul este valoarea lui Y la sfârșitul programului.

(2p) <u>TEOREMA 1</u>: Orice funcție recursivă este calculabilă cu programe standard.

→ <u>Dem</u>: <u>Pas 1</u>: Funcțiile elementare sunt calculabile cu P.S.

- $succ(x) = x + 1$:

$Y \leftarrow X$

$Y \leftarrow Y + 1$

- funcția proiecție:

$\pi_k^{(n)}(x_1, \dots, x_n) = x_k$

$Y \leftarrow X_k$

- funcția constantă:

$C_k^{(n)}(x_1, \dots, x_n) = k$

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

Pas 2: Funcțiile obținute prin aplicarea operațiilor sunt calculabile cu P.S.

- compunerea funcțională:

$$f(x_1,\dots,x_k) = h(\underbrace{g_1(x_1,\dots,x_k),\dots,g_m(x_1,\dots,x_k)}_{\text{calculabile cu P.S.}})$$

$z_{t+1} \leftarrow g_1(x_1,\dots,x_k)$

$z_{t+2} \leftarrow g_2(x_1,\dots,x_k)$

$\vdots$

$z_{t+m} \leftarrow g_m(x_1,\dots,x_k)$

$Y \leftarrow h(z_{t+1}, z_{t+2},\dots, z_{t+m})$

- recurența primitivă:

$f(x_1,\dots,x_k,0) = g(x_1,\dots,x_k) \to$ calc. cu P.S.

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

$\to$ calc. cu P.S.

$z_{p+1} \leftarrow g(x_1,\dots,x_k)$

IF $x_{t+1} \neq 0$ GO TO A_{r+1}

$Y \leftarrow z_{p+1}$

GO TO E

$$A_{r+1}: z_{p+2} \leftarrow h(x_1,\dots,x_k, \underset{\overset{\shortparallel}{0}}{z_{p+3}}, \underset{\overset{\shortparallel}{f(x_1,\dots,x_k,0)}}{z_{p+1}})$$

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

$$z_{p+3} \leftarrow z_{p+3} + 1$$

$$z_{p+1} \leftarrow z_{p+2}$$

if $x_{k+1} \neq 0$ GO TO A_{r+1}

$Y \leftarrow z_{p+2}$

- minimizarea nemărginită: calc. cu P.S.

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

$A_{r+2}: z_{p+1} \leftarrow g(x_1,\dots,x_k,z_{p+2})$

if $z_{p+1} \neq 0$ GO TO A_{r+1}

$Y \leftarrow z_{p+2}$

GO TO E

$A_{r+1}: z_{p+2} \leftarrow z_{p+2} + 1$

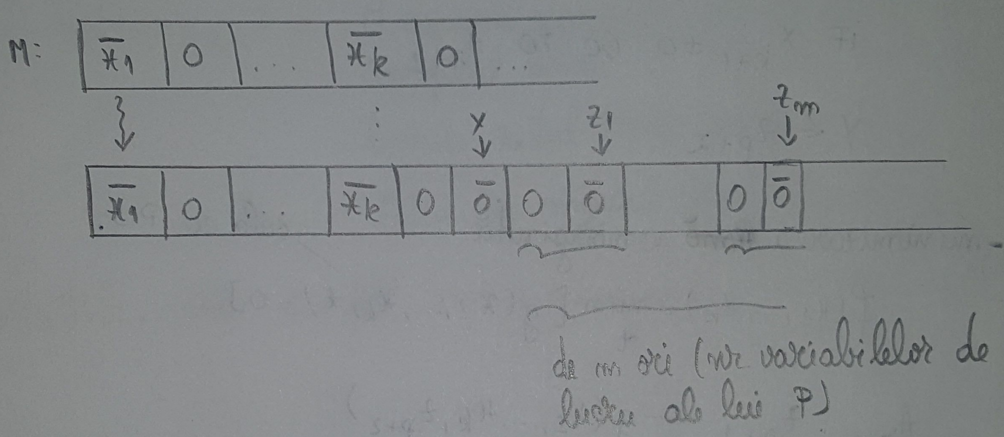
GO TO A_{r+2}

$$\begin{tikzcd} M.T. & & F.R. \arrow[ll] \arrow[dl] \\ & P.S. \arrow[ul, dashed] & \end{tikzcd}$$

(2p) TEOREMA 2: Orice funcție calculabilă cu programe standard este Turing calculabilă.

Dem: Fie P un program standard care calculează funcția $f(x_1, \dots, x_k)$.

P are instrucțiunile $\dot{I}_1, \dot{I}_2, \dots, \dot{I}_n$, $n \geq 1$



Starea curentă $\langle \dot{I}_j \rangle$, $1 \leq j \leq n$ reține o instrucțiune. Pentru fiecare stare curentă $\langle \dot{I}_j \rangle$ M execută următorul program:

- dacă $\dot{I}_j$ este $V \leftarrow V$, M trece în starea $\dot{I}_{j+1}$;
- dacă $\dot{I}_j$ este $V \leftarrow V+1$, M crește segmentul de pe bandă, asociat lui V, cu un 1 și trece în starea $\dot{I}_{j+1}$;
- dacă $\dot{I}_j$ este $V \leftarrow V \dot{-} 1$, M verifică dacă segmentul asociat lui V are un singur 1, iar dacă DA atunci trece în starea $\dot{I}_{j+1}$, altfel, șterge un 1 din segment și trece în starea $\dot{I}_{j+1}$;

- dacă $\dot{I}_j$ este IF $V \neq 0$ GO TO L, M verifică dacă segmentul asociat lui V are un singur 1, iar dacă da atunci trece la $\langle \dot{I}_{j+1} \rangle$, altfel trece la $\langle \dot{I}_s \rangle$, unde:

$$s = \begin{cases} \min_r (\, \dot{I}_r \text{ are eticheta } L) \text{ și } L \neq E \\ n+1, \text{ altfel} \end{cases}$$

În starea $\langle \dot{I}_{n+1} \rangle$: - M copiază segmentul asociat lui Y la începutul benzii;
- rescrie restul benzii cu 0;
- intră în singura stare finală q_f.

(4p) <u>TEOREMA 3</u>: Orice funcție Turing calculabilă este recursivă.

↳ puteți alege să scrieți dem. la asta și doar atât