Cap III : Funcții recursive

- Funcții elementare: succesor $s(x) = x+1$

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

  proiecția $\pi_k(x_1, \ldots, x_n) = x_k$

- Operații

  - compunere $f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n))$

  - recursivitate: $\begin{cases} f(0) = k & k - \text{const} \\ f(t+1) = g(t, f(t)) \end{cases}$

$$\begin{cases} f(x_1, \ldots, x_n, 0) = h(x_1, \ldots, x_n) \\ f(x_1, \ldots, x_n, t+1) = g(x_1, \ldots, x_n, t, f(x_1, \ldots, x_n, t)) \end{cases}$$

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

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

$f_+(x_1, t+1) = (x_1 + t) + 1 = f_+(x_1, t) + 1 = s(\pi_3^{(3)}(x_1, t, f_+(x_1, t)))$

2) $f_*(x_1, x_2) = x_1 * x_2$

$f_*(x_1, 0) = 0 = C_0^{(1)}(x_1)$.

$f_*(x_1, t+1) = x_1 \cdot (t+1) = (x_1 * t) + x_1 = f_*(x_1, t) + x_1 =$

$= f_+(\pi_3^{(3)}(x_1, t, f_*(x_1, t)), \pi_1^{(3)}(x_1, t, f_*(x_1, t)))$

3) $f_\wedge(x_1, x_2) = x_1^{x_2}$

$f_\wedge(x_1, 0) = x_1^0 = 1 = C_1^{(1)}(x_1)$.

$f_\wedge(x_1, t+1) = x_1^{t+1} = x_1^t \cdot x_1 = f_\wedge(x_1, t) \cdot x_1$

$= f_*(\pi_3^{(3)}(x_1, t, f_\wedge(x_1, t)), \pi_1^{(3)}(x_1, t, f_\wedge(x_1, t)))$

4) $f(x) = a^x$, $a =$ constantă

$f(0) = a^0 = 1$

$f(t+1) = a^{t+1} = a^t \cdot a = f(t) \cdot a = f_*(\pi_2^{(2)}(t, f(t)), C_a^{(2)}(t, f$

⑤ $f_!(x) = x! = 1 \cdot 2 \cdot \ldots \cdot x$.

$f_!(0) = 0! = 1$.

$f_!(t+1) = (t+1)! = t! \cdot (t+1) = f_!(t) \cdot (t+1) =$

$= f_*\left(\pi_2^{(2)}(t, f_!(t)), s\left(\pi_1^{(2)}(t, f_!(t))\right)\right)$

⑥ $f_\Sigma(x) = 1+2+\ldots+x$

$f(0) = 0$

$f_\Sigma(t+1) = 1+2+\ldots+t+t+1 = [f_\Sigma(t) + t]+1 =$

$= s\left(f_+\left(\pi_2^{(2)}(t, f_\Sigma(t)), \pi_1^{(2)}(t, f_\Sigma(t))\right)\right)$

sau

$= s(f_+(t, f_\Sigma(t)))$

⑦ $\text{pred}(x) = \begin{cases} x-1, & x > 0 \\ 0, & x = 0 \end{cases}$

$\text{pred}(0) = 0$

$\text{pred}(t+1) = t = \pi_1^{(2)}(t, \text{pred}(t))$

⑧ $f_{\dot-}(x_1, x_2) = \begin{cases} x_1 - x_2, & x_1 \geq x_2 \\ 0, & x_1 < x_2 \end{cases}$

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

$f(x_1, t+1) = x_1 - (t+1) = (x_1 - t) - 1 = \text{pred}\left(\pi_3^{(3)}(x_1, t, f_{\dot-}(x_1, t))\right)$

⑨ $f_{||}(x_1, x_2) = |x_1 - x_2|$

$f_{||}(x_1, x_2) = (x_1 \dot- x_2) + (x_2 \dot- x_1) = \cancel{f_+(f_{\dot-}(x_1, x_2), f_{\dot-}(x_2, x_1))}$

$= f_+\left(f_{\dot-}(x_1, x_2), f_{\dot-}\left(\pi_2^{(2)}(x_1, x_2), \pi_1^{(2)}(x_1, x_2)\right)\right)$

(10) $f_{max}(x_1, x_2) = \begin{cases} x_1, & x_1 \geq x_2 \\ x_2, & x_1 < x_2 \end{cases}$

$f_{max} = x_2 + (x_1 \dot{-} x_2) = f_+(\pi_2^{(2)}(x_1, x_2), f_-(x_1, x_2)$.

(11) $f_{min}(x_1, x_2) = \begin{cases} x_1, & x_1 \leq x_2 \\ x_2, & x_1 > x_2 \end{cases}$

$f_{min}(x_1, x_2) = x_1 \dot{-} (x_1 \dot{-} x_2) = f_{\dot{-}}(\pi_1^{(2)}(x_1, x_2), f_{\dot{-}}(x_1, x_2))$.

$f_{min}(x_1, x_2) = (x_1 + x_2) \dot{-} f_{max}(x_1, x_2) = f_-(f_+(x_1, x_2), f_{max}(x_1, x_2))$.

$f_{min}(x_1, x_2) = f_{\dot{-}}(f_{max}(x_1, x_2)$

(12) par, impar

Predicate.

$\boxed{13}$ $\alpha(x) = \begin{cases} 0, & x>0 \\ 1, & x=0 \end{cases} \Longleftrightarrow 1-x.$

$\alpha(0) = 1$

$\alpha(t+1) = 0 = C_0^{(2)}(t, \alpha(t)).$

$\boxed{14}$ $f_{=}(x_1, x_2) = \begin{cases} 1, & x_1 = x_2 \\ 0, & x_1 \neq x_2. \end{cases}$

$f_{=}(x_1, x_2) = 1 \dot{-} |x_1 - x_2) = \alpha(f_{||}(x_1, x_2)).$

$\boxed{15}$ $f_{\leq}(x_1, x_2) = \begin{cases} 1, & x_1 \leq x_2 \\ 0, & x_1 > x_2 \end{cases}.$

$f_{\leq}(x_1, x_2) = 1 \dot{-} (x_1 \dot{-} x_2) = \alpha(f_{\dot{-}}(x_1, x_2)).$

SEMINAR 6 – CC.

Cap II – Programe Standard

→ Instrucțiuni : $V \leftarrow V$

$V \leftarrow V+1$

$V \leftarrow V \dot{-} 1$ – $\begin{cases} V-1, & \text{dacă } V \geq 1 \\ 0, & \text{dacă } V = 0 \end{cases}$

IF $V \neq 0$ GOTO L

→ Etichete – L_1, L_2, L_3 ....

– E (exit, iese din program).

→ Variabile : – de intrare : $x_1, x_2, \ldots$

– de ieșire : Y_1, Y_2. (în general una, Y)

– aux, interne, de lucru : Z_1, Z_2 ...

} inițial, au val. 0.

→ $Y \leftarrow X$

IF $X \neq 0$ GOTO : L_1

$Z_1 \leftarrow Z_1 + 1$

IF $Z_1 \neq 0$ GOTO E

$\Big\} \Leftrightarrow$ GOTO E

[L_1]: $Y \leftarrow Y+1$

$Z_2 \leftarrow Z_2 + 1$

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

IF $X \neq 0$ GOTO L_1

[L_2]: $Z_2 \leftarrow Z_2 \dot{-} 1$

$X \leftarrow X+1$

IF $Z_2 \neq 0$ GOTO L_2

CT : $1 + \underbrace{X \cdot 4}_{L_1} + \underbrace{X \cdot 3}_{L_2} \Rightarrow O(x)$

→ ex2 $Y \leftarrow X_1 + X_2$

IF $X_1 \neq 0$ GOTO L_1

[L_1] $Z_1 \leftarrow Z_1 + 1$

P

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

$\boxed{13}$ $Y_1 \leftarrow Y + 1$

IF $X_1 \neq 0$ GOTO L_1

[L_2] IF $X_2 \neq 0$ GOTO L_3

GOTO L_4

II [L_3] $Z_2 \leftarrow Z_2 + 1$

$X_2 \leftarrow X_2 \dot{-} 1$

$Y \leftarrow Y + 1$

GOTO L_2

[L_4] IF $Z_1 \neq 0$ GOTO L_5

GOTO L_6

[L_5] $Z_1 \leftarrow Z_1 \dot{-} 1$

$X_1 \leftarrow X_1 + 1$

GOTO L_4

[L_6] IF $Z_2 \neq 0$ GOTO L_7

GOTO E

[L_7] $Z_2 \leftarrow Z_2 \dot{-} 1$

$X_2 \leftarrow X_2 + 1$

GOTO L_6

CT: $1 + \underbrace{1 \cdot 4}_{4} + \underbrace{X_2 \cdot 5}_{L_2 + L_3} + \underbrace{\cancel{X}_1 \cdot 4}_{L_4 + L_5} + \underbrace{X_2 \cdot 4}_{L_6 + L_7} = O(X_1 + X_2)$

ex3

$\longrightarrow$ $\underline{Y \leftarrow X_1 \cdot X_2}$

```
       IF X1 ≠ 0  GOTO  L1
       GOTO E
[L1]:  IF X2 ≠ 0  GOTO L2
       GOTO E
[L2]:  Z1 ← Z1 + 1
       X1 ← X1 ∸ 1
[L3]:  Y ← Y + 1
       Z2 ← Z2 + 1
       X2 ← X2 ∸ 1
       IF X2 ≠ 0 GOTO L3
[L4]:  X2 ← X2 + 1
       Z2 ← Z2 ∸ 1
       IF Z2 ≠ 0 GOTO L4
       IF X1 ≠ 0 GOTO L2
[L5]:  Z1 ← Z1 ∸ 1
       X1 ← X1 + 1
       IF Z1 ≠ 0 GOTO L5
       (GOTO E).
```

$$T: \underbrace{1+1}_{L_1} + \underbrace{x_1 \cdot \left[2 + \underbrace{x_2(4)}_{L_3} + \underbrace{x_2 \cdot 3}_{L_4} + 1\right]}_{L_2} + \underbrace{x_1 \cdot 3}_{L_5} \Rightarrow O(x_1 x_2)$$

(4) $\longrightarrow$ $y \leftarrow f_{=}(x_1, x_2) = \begin{cases} 1, & x_1 = x_2 \\ 0, & x_1 \neq x_2 \end{cases}$

```
[L0]  IF X1 ≠ 0 GOTO L1
      IF X2 ≠ 0 GOTO L3        // X1 < X2
      Y <- Y+1
      GOTO L3                  // X1 = X2
[L1]  IF X2 ≠ 0 GOTO L2
      GOTO L3                  // X1 > X2
[L2]  X1 <- X1 ∸ 1
      X2 <- X2 ∸ 1
      Z1 <- Z1 + 1
      GOTO L0
[L3]  IF Z1 ≠ 0 GOTO L4
      GOTO E
[L4]  Z1 <- Z1 ∸ 1
      X1 <- X1 + 1
      X2 <- X2 + 1
      ~~IF Z1 ≠ 0~~ GOTO L3
```

CT: $\underbrace{\min(x_1, x_2) \cdot 6}_{L_0 + L_1 + L_2} + \underbrace{\min(x_1, x_2) \cdot 5}_{L_3 + L_4} \Rightarrow O(\min(x_1, x_2))$

(5) $Y \leftarrow f_{:2}(x) = \begin{cases} 1, & x \text{ par} \\ 0, & y \text{ impar} \end{cases}$

```
[L0]  iF  X ≠ 0  GOTO L1
      Y ← Y + 1
      GOTO L2          // X par (X=0)
[L1]  X ← X ∸ 1
      Z ← Z + 1
      iF X ≠ 0  GOTO L3
        GOTO L2        // X impar
[L3]  X ← X ∸ 1
      Z ← Z + 1
      GOTO L0
[L2]  iF Z ≠ 0  GOTO L4
      GOTO E
[L4]  Z ← Z ∸ 1
      ~~Z ← Z~~
      X ← X + 1
      GOTO L2
```

CT: $\underbrace{\underbrace{\left[\frac{x}{2}\right]}_{L_0} \cdot 7}_{L_0+L_1+L_3} + \underbrace{x \cdot 4}_{L_2+L_4} \rightarrow O(x)$.

(6) $Y \leftarrow |X_1 - X_2|$

```
]  iF  X1 ≠ 0  GOTO L1
   iF  X2 ≠ 0  GOTO L2       // Y ← X2
   GOTO L3                   // X1 = X2   Y=0

]  X2 ← X2 − 1
   Z2 ← Z2 + 1
   Y ← Y + 1
   iF X2 ≠ 0  GOTO L2
```

[L4] $x_2 \leftarrow x_2 + 1$
$z_2 \leftarrow z_2 \dot{-} 1$
IF $z_2 \neq 0$ GOTO L_4
GOTO L_3

[L1] IF $x_2 \neq 0$ GOTO L_5

[L6] ~~Y~~ $\leftarrow x_1$
$x_1 \leftarrow x_1 \dot{-} 1$
$z_1 \leftarrow z_1 + 1$
$y \leftarrow y + 1$
IF $x_1 \neq 0$ GOTO L_6

[L7] $x_1 \leftarrow x_1 + 1$
$z_1 \leftarrow z_1 \dot{-} 1$
IF $z_1 \neq 0$ GOTO L_7
GOTO L_3

[L5] $x_1 \leftarrow x_1 \dot{-} 1$
$x_2 \leftarrow x_2 \dot{-} 1$
$z_3 \leftarrow z_3 + 1$
GOTO L_0

[L3] IF $z_3 \neq 0$ GOTO L_8
GOTO E

[L8] $z_3 \leftarrow z_3 - 1$
$x_1 \leftarrow x_1 + 1$
$x_2 \leftarrow x_2 + 1$
GOTOL_3

CT: $\underbrace{\min(x_1, x_2) \cdot 6}_{L_0 + L_1 + L_5} + \underbrace{|x_1 - x_2| \cdot (4 + 3)}_{\substack{L_2 + L_4 \\ \text{sau} \\ L_6 + L_7}} + \underbrace{\min(x_1, x_2) \cdot 5}_{L_3 + L_8} =$

$= O(\min\{x_1, x_2\})$

$$f_{<}(x_1, x_2) = \begin{cases} 1, & \text{dacă } x_1 < x_2 \\ 0, & \text{dacă } x_1 \geq x_2. \end{cases}$$

// testăm întâi

$$f_{\leq}(x_1, x_2) = \begin{cases} 1, & \text{dacă } x_1 \leq x_2 \\ 0, & \text{dacă } x_1 > x_2 \end{cases}$$

Seminar 7

(9) $\longrightarrow$ $y \leftarrow x_1^{x_2}$

[L_0] if $x_1 \neq 0$ GOTO L_1

if $x_2 \neq 0$ GOTO E // $0^{x_2} = 0$, $x_2 \neq 0$

GOTO L_0 // pt. 0^0 ciclăm.

[L_1] if $x_2 \neq 0$ GOTO L_2

$y \leftarrow y+1$

GOTO E // $x_1^0 = 1$, $x_1 \neq 0$.

[L_2] $y \leftarrow y+1$.

$z_2 \leftarrow x_2$ // $O(x_2)$

[L_3] $z_1 \leftarrow y * x_1$; $y \leftarrow 0$; // $O(y * x)$

$y \leftarrow z_1$; $z_1 \leftarrow 0$

$z_2 \leftarrow z_2 - 1$

if $z_2 \neq 0$ GOTO L_3

$$CT: \underset{L_0}{1} + \underset{L_1}{1} + \underbrace{1 + x_2}_{L_2} + \underbrace{x_2 \left(x_1^{x_2} + x_1^{x_2} + 2 \right)}_{L_3} = O\left(x_2 \cdot x_1^{x_2}\right)$$

sau

$$x_1 + x_1^2 + x_1^3 + \dots + x_1^{x_2} = \cancel{x_1} \cdot \frac{x_1^{x_2} - 1}{\cancel{x_1 - 1}} = O\left(x_1^{x_2}\right)$$

(10) $\longrightarrow$ $y_1 \leftarrow x_1 / x_2$

$y_2 \leftarrow x_1 \% x_2$

[L_0] if $x_2 \neq 0$ GOTO L_1

GOTO L_0 // $x_2 = 0 \to$ ciclăm

[ ] if $x_1 \neq 0$ GOTO L_2

GOTO E. // $y_2 = y_1 = x_1 = 0$

[L2]:

$Y_2 \leftarrow X_1$ // $O(X_1)$

[L4] $z_0 \leftarrow f_{\leq}(X_2, Y_2)$ / $O(\min(X_2, Y_2))$

IF $z_0 \neq 0$ GOTO L3

GOTO E

[L3] $Y_2 \leftarrow Y_2 - X_2$ // $O(X_2)$.

$Y_1 \leftarrow Y_1 + 1$ // C

GOTO L4

CT: $\underset{L_0}{1} + \underset{L_1}{1} + X_1 + \underbrace{X_1/X_2\,(X_2 + 1 + \underbrace{X_2 + 1 + 1}_{L_3})}_{L_4} \Rightarrow \boxed{O(X_1)}$

(11)

$Y \leftarrow X!$

IF $X_0 \neq 0$ GOTO L1

$Y \leftarrow 1$

GOTO E // $0! = 1$.

[L1] $Z_1 \leftarrow X$

$Y \leftarrow Y + 1$

[L2] $Z_2 \leftarrow Y \cdot Z_1$; $Y = 0$ // $O(Y Z_1)$

$Y \leftarrow Z_2$; $Z_2 \leftarrow 0$ // $O(Z_2)$

$Z_1 \leftarrow Z_1 - 1$

IF $Z_1 \neq 0$ GOTO L2

GOTO E

CT: $\underset{L_0}{1} + \underset{L_1}{X+1} + \underbrace{X\,(X! + X! + 2)}_{L_2} \Rightarrow O(X \cdot X!)$.

(2)

(12)

$\longrightarrow$

$y \leftarrow$ patrat perfect $- \begin{cases} 1, & \text{dacă da} \\ 0, & \text{altfel.} \end{cases}$

$x = \underbrace{1+3+5+\dots+(2\sqrt{x}-1)}_{\sqrt{x} \text{ numere}}$

```
    iF x≠0 GOTO L1
    y ← y+1
    GOTO E   //0 este patrat perfect.
[L1] z0 ← x.   //O(x)
     z1 ← z1+1
     z2 ← f≤(z1, z0)   //O(min(z1,z0)).
[L3] iF z2 ≠0 GOTO L2.
     iF z0 ≠0 GOTO E;   //y=0; x≠p.p.
     y ← y+1            //y=1; x - patrat perfect.
     GOTO E
[L2] z0 ← z0 - z1   //O(z1).
     z1 ← z1+1;
     z1 ← z1+1;
     GOTO L3;
```

CT: $\underbrace{1}_{L_0} + \underbrace{x+1}_{L_1} + \underbrace{\sqrt{x}\,(2\sqrt{x}-1+\underbrace{1+2\sqrt{x}-1+3}_{L_2})}_{L_3} \Rightarrow O(x).$

(13)

$\longrightarrow$ $y \leftarrow$ al x-lea nr. din şirul lui Fibonacci.

x:	0	1	2	3	4	5	6	7	---
y:	0	1	1	2	3	5	8	13	---

```
    IF X≠0 GOTO L1
    GOTO E                      // x=y=0.

[L1]  z0 ← x
      z1 ← 0
      z2 ← 1

      z0 ← z0 ∸ 1.
[L2]  z0 ← z0 ∸ 1  ; Y ← 0.
      Y ← z1 + z2         // O(z1+z2)
      z1 ← z2; z2 ← 0     z1 ← 0    // O(z2)
      z2 ← Y
   IF z0 ≠ 0  GOTO L2          // O(Y).
   GOTO E
```

CT: $\underbrace{1}_{L_0} + \underbrace{x+3}_{L_1} + x(\underset{!}{\underline{y}} + 3y + 1) \Rightarrow O(xy)$

(4) $Y \leftarrow palindrom(x) = \begin{cases} 1, & x \text{ palindrom} \\ 0, & \text{altfel} \end{cases}$

$Z_0 \leftarrow X$ // $O(x)$

$Z_{10} \leftarrow 10$ // $O(10)$

iF $Z_0 \neq 0$ GOTO L_1

$Y \leftarrow Y+1$

GOTO E // 0 e palindrom

[L_1]: $Z_1 \leftarrow Z_0 / Z_{10}$;

$Z_2 \leftarrow Z_0 \% Z_{10}$; $Z_0 \leftarrow 0$ // $O(Z_0)$

$Z_0 \leftarrow Z_1$; $Z_1 \leftarrow 0$. // $O(Z_1)$

$Z_3 \leftarrow Z_4 \cdot Z_{10}$; $Z_4 \leftarrow 0$; $O(Z_4 \cdot Z_{10})$

$Z_4 \leftarrow Z_3 + Z_2$; // $O(Z_3 + Z_2)$

$Z_3 \leftarrow 0$; $Z_2 \leftarrow 0$

iF $Z_0 \neq 0$ GOTO L_1

$Y \leftarrow f_=(X, Z_4)$. // $O(\min(X, Z_4))$.

CT: $X + 10 + 1 + \log_{10} X (X + \cancel{X} + X + \cancel{X} + 1) + X \rightarrow O(x \log x)$

(5) ..., ..., x_n, t)] = 1

CC SEMINAR 9

$\neg P = 1 \dot{-} p = \alpha(p)$

$P \wedge Q = p * Q \overset{sau}{=} \min(p, q)$

$P \vee Q = \max(p, q) \overset{sau}{=} \neg(\neg P \wedge \neg Q) = \alpha(\alpha(p) * \alpha(Q))$.

→ Dacă g și h sunt funcții calculabile și p e predicat calculabil, atunci f e fct. calculabilă.

$$f(x_1, \dots, x_n) = \begin{cases} g(x_1, \dots, x_n) & \text{dacă } p(x_1, \dots, x_n) = 1 \\ h(x_1, \dots, x_n) & \text{dacă } p(x_1, \dots, x_n) = 0 \end{cases}.$$

$$\boxed{f = g \cdot P + h(1 \dot{-} p) = g P + h(\alpha(p))}$$

→ Dacă f e fct. calculabilă, atunci g și h sunt fct. calculabile

$$g(x_1, \dots, x_n, x_{n+1}) = \sum_{t=0}^{x_{n+1}} f(x_1, \dots, x_n, t).$$

$$h(x_1, \dots, x_n, x_{n+1}) = \prod_{t=0}^{x_{n+1}} f(x_1, \dots, x_n, t).$$

$g(x_1, \dots, x_n, 0) = f(x_1, \dots, x_n, 0)$.

$g(x_1, \dots, x_n, t+1) = g(x_1, \dots, x_n, t) + f(x_1, \dots, x_n, t+1)$.

$h(x_1, \dots, x_n, t+1) = h(x_1, \dots, x_n, t) * f(x_1, \dots, x_n, t+1)$.

$h(x_1, \dots, x_n, 0) \overset{proiectie}{=} f(x_1, \dots, x_n, 0)$

→ Dacă P e predicat calculabil atunci și următoarele predicate sunt calculabile:

$(\forall t)_{\leq x_{n+1}} \; P(x_1, \dots, x_n, t)$

$(\exists t)_{\leq x_{n+1}} \; P(x_1, \dots, x_n, t)$

$(\forall t)_{< x_{n+1}} \; P(x_1, \dots, x_n, t)$

$(\exists t)_{< x_{n+1}} \; p(x_1, \dots, x_n, t)$.

$(\forall t)_{\leq x_{n+1}} \; p(x_1, \dots, x_n, t) \iff \left[\prod_{t=0}^{x_{n+1}} P(x_1, \dots, x_n, t)\right] = 1$

$$(\exists t)_{\leq x_{n+1}} \; p(x_1, \dots, x_n, t) \Leftrightarrow \left[\sum_{t=0}^{x_{n+1}} p(x_1, \dots, x_n, t) \right] \neq 0$$

$$(\forall t)_{< x_{n+1}} \; p(x_1, \dots, x_n, t) \Leftrightarrow (\forall t)_{\leq x_{n+1}} \left[p(x_1, \dots, x_n, t) \vee (t =$$

$$(\exists t)_{< x_{n+1}} \; p(x_1, \dots, x_n, t) \Leftrightarrow (\exists t)_{\leq x_{n+1}} \left[p(x_1, \dots, x_n, t) \wedge (t \neq x_{n+1}$$

→ Dacă P predicat calculabil, atunci următoarea fct. este calculabilă:

$$\min_{t \leq x_{n+1}} p(x_1, \dots, x_n, t)$$

EX1 $P_{:}(x_1, x_2) = \begin{cases} 1, & x_1 \vdots x_2 \\ 0, & x_1 \not\vdots x_2 \end{cases}$

nu e indice

$$P_{:}(x_1, x_2) \Leftrightarrow (\exists t)_{\leq x_1} [x_1 = t * x_2]$$

EX2 $P_{|}(x_1, x_2) = \begin{cases} 1, & x_1 \mid x_2 \\ 0, & x_1 \mid x_2 \end{cases}$

$$P_{|}(x_1, x_2) \Leftrightarrow (\exists t)_{\leq x_2} [x_2 = t * x_1]$$

$\boxed{EX3}$ $\text{prim}(x) = \begin{cases} 1, & x \text{ prim} \\ 0, & \text{altfel.} \end{cases}$

$\text{prim}(x) \Leftrightarrow (x>1) \wedge (\forall t)_{<x} \left[\alpha(P_{\vdots}(x,t)) \vee (t \leq 1)\right]$

$\boxed{EX4}$ $P_p(x) \Leftrightarrow \begin{cases} 1, & x \text{ pătrat perfect} \\ 0, & \text{altfel} \end{cases}$

$P_p(x) \Leftrightarrow (\exists t)_{\leq x} \left[x = t \cdot t\right]$

$\boxed{EX6}$ $\text{Nr.perfect}(x) = \begin{cases} 1, & x = \text{suma divizorilor lui } x \text{ diferiți de } x \\ 0, & \text{altfel.} \end{cases}$

$\text{Nr.Perfect}(x) \Leftrightarrow x = \sum_{t=0} t \cdot \left[P_{\vdots}(x,t) \wedge (t \neq x)\right]$

$\boxed{EX7}$

$f_/(x_1,x_2) = \left[\frac{x_1}{x_2}\right]$

$f_/(x_1,x_2) = \min_{t \leq x_1} \left[x_2 * (t+1) > x_1\right]$

$f_{\%}(x_1,x_2) = x_1 \% x_2$

$f_{\%}(x_1,x_2) = x_1 \dot{-} x_2 \cdot f_/(x_1,x_2)$

sau $f_{\%}(x_1,x_2) = \min_{t < x_2} \left[P_{\vdots}(x_1 - t, x_2)\right]$

<u>Test</u> - Model

① $f(x_1,x_2) = \begin{cases} 2 * x_1 - x_2, & \text{dacă } x_1 \leq x_2 \\ \frac{x_2+3}{x_1}, & \text{dacă } x_1 > x_2 \end{cases}$

$f(x_1,x_2) = f_{\dot{-}}\left(f_*\left(C_2^2(x_1,x_2), \Pi_1^{(2)}(x_1,x_2)\right), \Pi_2^{(2)}(x_1,x_2)\right) * P_{\leq}(x_1,x_2)$

$+ f_{\wedge}\left(\Pi_1^{(2)}(x_1,x_2), f_+\left(\Pi_2^{(2)}(x_1,x_2), C_3^{(2)}(x_1,x_2)\right)\right) * \alpha\left(P_{\leq}(x_1,x_2)\right)$

1.

(2) $f(x) = \sum_{k=0}^{x} (3 * k \dot{-} 2)$

$f(0) = 3 * 0 \dot{-} 2 = 0;$

$f(t+1) = f(t) + (3 * (t+1) \dot{-} 2) =$

$= f_+ \left(\pi_2^{(2)}(t, f(t)), s(f_* (c_3^{(2)}(t, f(t)), \pi_1^{(2)}(t, f(t)))) \right)$

CC - Seminar 8

$\longrightarrow Y \leftarrow cmmdc(x_1, x_2)$

$x_1 = 3 ; \quad x_2 = 5$

```
[L0]: IF x1 ≠ 0 GOTO L1
      Y ← x2              // O(x2)
      GOTO E

[L1]: IF x2 ≠ 0 GOTO L2
      Y ← x1              // O(x1)
      GOTO E

[L2]: z1 ← x1             // O(x1)
      z2 ← x2             // O(x2)
[L4]  z0 ← f<(z1, z2)
      IF z0 ≠ 0 GOTO L3
      z1 ← z1 - z2        // O(z2)
      IF z1 ≠ 0 GOTO L4
      Y ← z2              // O(z2)
      GOTO E

[L3]: z2 ← z2 - z1        // O(z1).
      IF z2 ≠ 0 GOTO L4
      Y ← z1              // O(z1)
      GOTO E.
```

$$T = \underbrace{1+}_{L_0} \underbrace{1+}_{L_1} \underbrace{x_1 + x_2}_{L_2} + \underbrace{max(x_1, x_2)\left[min(x_1, x_2) + 1 + min(x_1, x_2) + 1\right]}_{L_4 + L_3}$$

$$+ \; min(x_1, x_2) = O(x_1 x_2).$$

$\longrightarrow Y \leftarrow [x_1, x_2]$

$[L_0]$: if $x_1 \neq 0$ GOTO L_1
GOTO E

(2)

$[L_1]$: if $x_2 \neq 0$ GOTO L_2
GOTO E

$[L_2]$: $z_0 \leftarrow f_<(x_1, x_2)$
if $z_0 \neq 0$ GOTO L_3
$z_1 \leftarrow x_1$ // $O(x_1)$, in z_1 avem max
$z_2 \leftarrow x_2$ // $O(x_2)$, in z_2 avem min
GOTO L_4

$[L_3]$: $z_1 \leftarrow x_2$
$z_2 \leftarrow x_1$ // $O(x_2)$, z_1 = maxim.
// $O(x_1)$, z_2 avem minim.

$[L_4]$: $Y \leftarrow Y + z_1$ // $O(z_1)$
$z_3 \leftarrow \cancel{Y} \,\%\, z_2$ // $O(z_1)$
if $z_3 \neq 0$ GOTO L_4

$$CT = \underbrace{1}_{L_0} + \underbrace{1}_{L_1} + \underbrace{\min(x_1, x_2) + 1 + x_1 + x_2}_{L_2 + L_3} + \underbrace{\min(x_1, x_2)\left(\max(x_1, x_2)\right)}_{L_4}$$

$\underbrace{x_1 x_2 + 1)}$

$O(x_1 x_2 \min(x_1 x_2))$

$\longrightarrow Y \leftarrow [\log_2 x]$

$\boxed{\text{Varianta 1}}$

[L_0] if $x \neq 0$ GOTO L_1
GOTO L_0 // pentru $x=0$ ciclăm.

[L_1] $z_1 \leftarrow x$ // $O(x)$
$z_2 \leftarrow 2$ // $O(2)$

$z_3 \leftarrow z_1 / z_2$; $z_1 \leftarrow 0$. // $O(z_1)$
$z_1 \leftarrow z_3$; $z_3 \leftarrow 0$: // $O(z_3)$
$Y \leftarrow Y+1$
if $z_1 \neq 0$ GOTO L_2
$Y \leftarrow Y-1$

CT: $\underset{L_0}{1} + \underset{L_1}{x \cdot 2} + \underbrace{\log_2 x\left(x + \frac{x}{2} + 1 + 1\right)}_{L_2} + 1 \Rightarrow O(x \log x)$.

$\boxed{\text{Varianta 2}}$

[L_0]: if $x \neq 0$ GOTO L_1
GOTO L_0 // Ciclăm

[L_1]: $z_1 \leftarrow 1$
$z_2 \leftarrow 2$

[L_2]: $z_3 \leftarrow z_2 \cdot z_1$; $z_1 \leftarrow 0$; $O(z_1 z_2)$
$z_1 \leftarrow z_3$; $z_3 \leftarrow 0$ // $O(z_3)$
$Y \leftarrow Y+1$
$z_0 \leftarrow f_{\leq}(z_1, x)$ // $O(\min(z_1, x))$.
if $z_0 \neq 0$ GOTO L_2
$Y \leftarrow Y-1$

CT: $\underset{L_0}{1} + \underset{L_1}{3} + \underbrace{\log_2 x\left(2x + 2x + 1 + \cancel{\min}\ x + 1\right)}_{L_2} + 1 \Rightarrow O(x \log x)$

②

$\rightarrow y \leftarrow prim(x) = \begin{cases} 1 & da \\ 0 & nu \end{cases}$

$\not{f}$

$\not{f}$ [L_0]: iF $x \neq 0$ GOTO L_1

$\not{f}$ GOTO E.

[L_1]: $z_2 \leftarrow 2$ // $O(2)$

$z_1 \leftarrow x / z_2$; // $O(x)$, z_1 = posibilul divizor.

[L_3] $z_3 \leftarrow f_<(z_1, z_2)$ // $O(\min(z_1, z_2))$

$y \leftarrow y+1$

iF $z_3 \neq 0$ GOTO E // x = prim.

$y \leftarrow y-1$

$z_4 \leftarrow x \% z_1$ // $O(x)$

iF $z_4 \neq 0$ GOTO L_2

GOTO E // $x \neq$ prim.

[L_2] $z_1 \leftarrow z_1 - 1$

GOTO L_3

CT: $\underbrace{1}_{L_0} + \underbrace{2 + x}_{L_1} + \frac{x}{2} \underbrace{(2 + 1 + 1 + 1 + x + 3)}_{L_3 + L_2} \Rightarrow O(x^2)$

① h și g fcții calculabile
p predicat calculabil $\Big\} \Rightarrow f$ fct. calculabilă

$$\boxed{f = g \cdot p + h(1-p) = g \cdot p + h(\alpha(p))}$$

$$f(x_1,\dots,x_n) = \begin{cases} g(x_1,\dots,x_n) & \text{dacă } p(x_1,\dots,x_n)=1 \\ h(x_1,\dots,x_n) & \text{, altfel} \end{cases}$$

② dacă f calculabilă $\Rightarrow \begin{cases} g \\ h \end{cases}$ calculabile

$$g(x_1,\dots,x_n,x_{n+1}) = \sum_{t=0}^{x_{n+1}} f(x_1,\dots,x_n,t).$$

$$h(x_1,\dots,x_n,x_{n+1}) = \prod_{t=0}^{x_{n+1}} f(x_1,\dots,x_n,t).$$

③ p predicat calculabil, atunci urm. predicate sunt calculabile:

$$\begin{cases} (\forall t)_{\leq x_{n+1}} \; p(x_1,\dots,x_n,t) \\ (\exists t)_{\leq x_{n+1}} \; p(x_1,\dots,x_n,t) \end{cases}.$$

$\Big\{$ idem. pe relație de inegalitate strictă.

④ p predicat calculabil, atunci $\boxed{\min_{t \leq x_{n+1}} p(x_1,\dots,x_n,t)}$ calculabil

CC - Seminar 2

Să se accepte limbajul $N = \{w = a^n b^k c^{2n} \mid n > k \geq 1\}$

$n=3$
$k=2$

Inainte :

...	B	a	a	a	b	b	c	c	c	c	c	c	B ...

După :

...	B	a'	a	a	b'	b	c'	c'	c	c	c	c	B

Pas 1: Marcăm câte un a, un b și 2 de c.

a) Citim un a, scriem a', pas →

b) Cât timp citim a sau b', nu modificăm banda, pas →

c) Citim un b, scriem b', pas →

d) Cât timp citim b sau c', nu modif. banda, pas →

e) Citim un c, scriem c', pas →

f) Citim un c, scriem c', pas ←

g) Cât timp citim c', b, b' sau a, nu modificăm, pas ←

h) Citim a', scriem a', pas →. Reluăm pasul 1a.

Pas 2

a) Citim c', scriem c', pas →

b) Cât timp citim c', ~~nu mo~~ scriem c', pas →.

c) Citim un c, scriem c', pas →

d) Citim un c, scriem c', pas ←

Pas 3 (Marcăm câte un a & 2 de c)

a) Cât timp citim c', b' și a, nu modificăm banda, pas ←

b) Citim a', scriem a', pas →

c) ~~Scriem a'~~ Citim a, scriem a', pas →

d) Cât timp citim a, b', c' nu modificăm, pas →

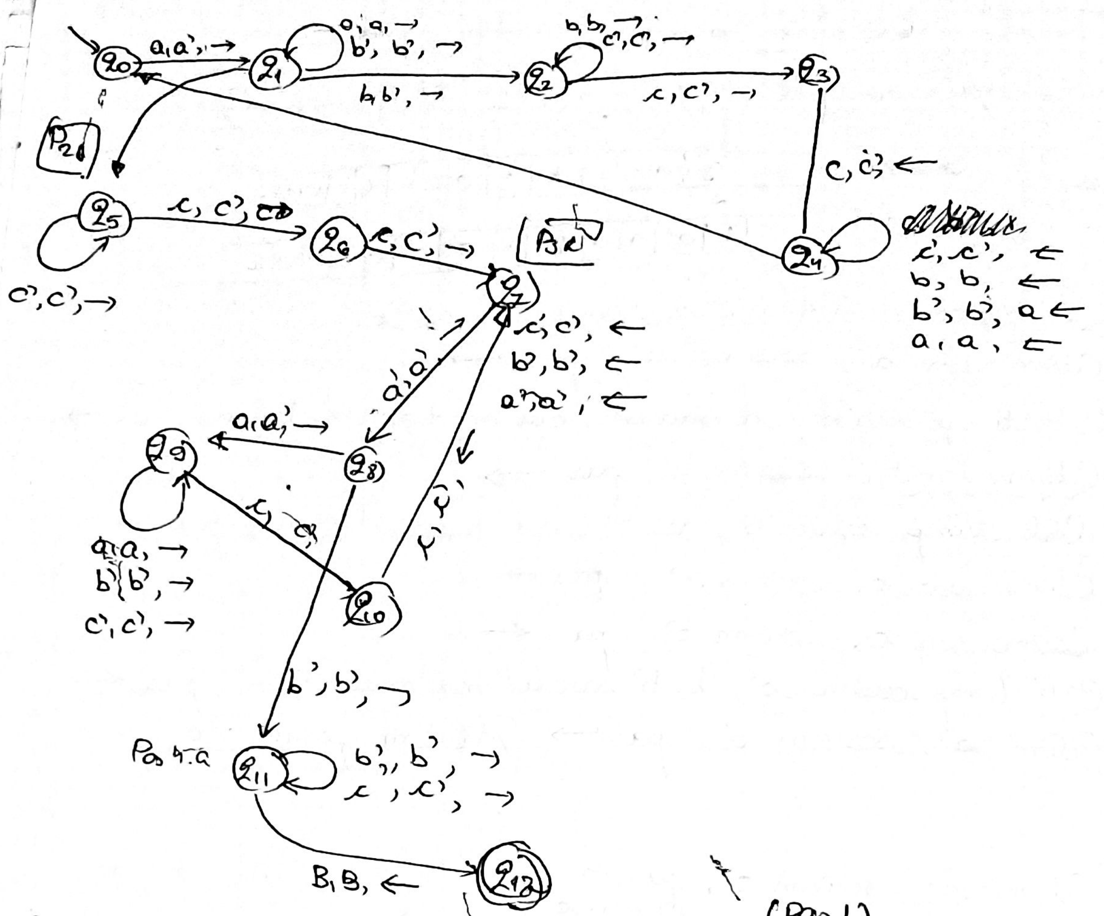
e) Citim c, scriem c', pas →

f) Citim c, scriem c', pas ←; Reluăm pas 3a.

Pas 4: a) Citim b', scriem b', pas →

b) Cât timp citesc b', c', nu modific, pas →

c) Citim B, scriu B, pas ←, stare finală



(Pas 1)

$$CS = |w| \; ; \quad CT = \min\left(|w|_a, |w|_b, \frac{|w|_c}{2}\right)\left(|w|_a - |w|_b + |w|_b + 2|w|_b\right) +$$

$$+ \underbrace{2|w|_b + 2}_{P2} + \underbrace{\min\left(|w|_a - |w_b|, \frac{w_c - 2w_b}{2}\right)}_{P3} \cdot (w_b + 2w_a)$$

$$+ w_b + 2w_a + 1$$

→ Se dau x și y în baza 1, delimitate de un 0. Să se calculeze $x \cdot y$.

$x=3$

$y=2$

Înainte

B	1	1	1	1	0	1	1	1	B	...

După

B	1'	1	1	1	0	1''	1''	1	2	1	1	1	1	1	1	1	B	..

Pas 1:

a) Citim un 1 din x, scriem 1', pas →

b) Cât timp citim 1 din x, scriem 1, pas →

c) Citim 0, scriem 0, ~~sch~~ pas →, sch. starea.

d) Citim un 1 din y, scriem 1'', pas →

e) Cât timp citim 1 din y, scriem 1, pas →.

f) Citim B, scriem 2, pas →

g) Citim B, scriem 1, pas ←

Pas 2) De x ori îl copiem pe y la finalul benzii.

a) Cât timp citim 2, 1, 1'' sau 0, nu modif, pas ←

b) Citim 1' din x, scriem 1', pas →

c) Citim un 1 din x, scriem 1', pas →

d) Cât timp citim 1 sau 0, nu modif. pas →

e) Cât timp citim 1'', scriem 1'', pas →.

Pas 3: Copiem pe y la final

a) Citim un 1 din y, scriem 1', pas →

b) Cât timp citim 1 sau 2, nu modif. banda, pas →

c) Citim B, scriem 1, pas ←

d) Cât timp citim 1 sau 2; nu modif. banda, pas ←

e) Citim 1' din y, scriem 1', pas →; Reiau pasul 3a

Pas 4 a) Citim 2, scriem 2, pas ←

b) (Demarcăm tot y)

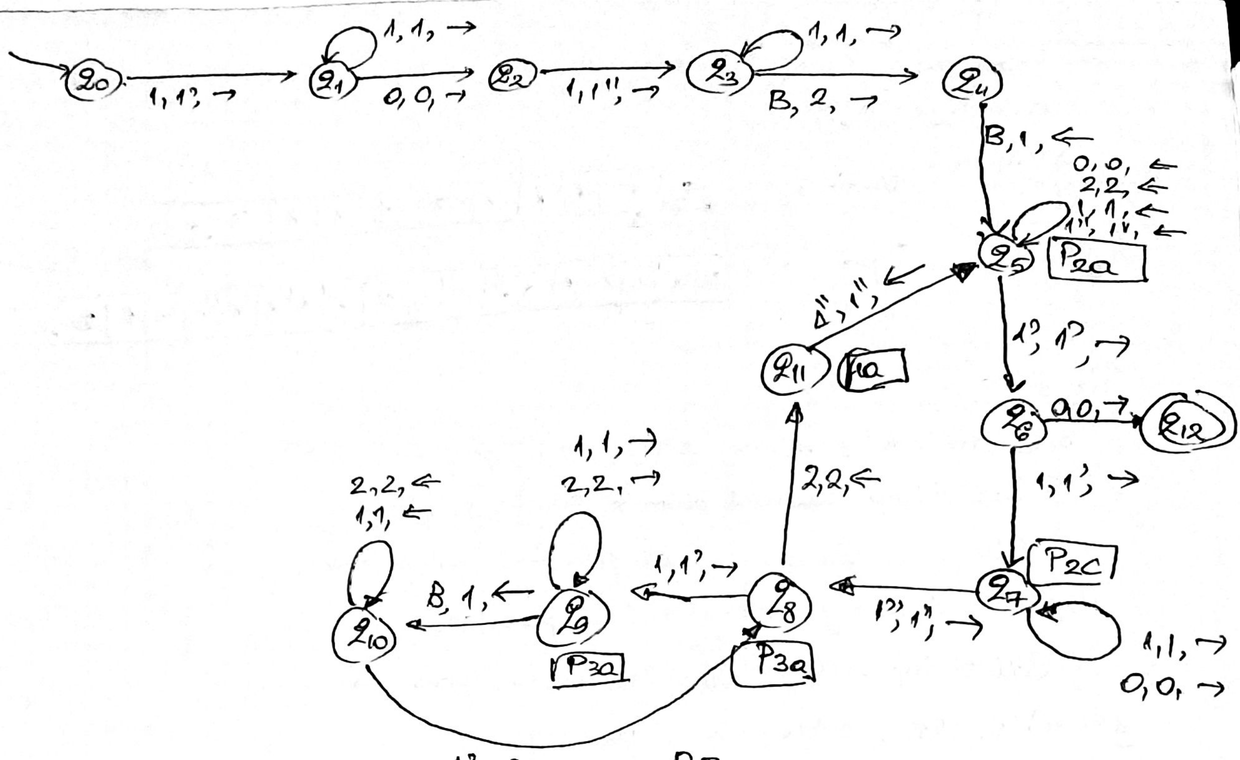
Cât timp citim 1', scriem 1'', pas ←

c) Citim 1'', scriem 1'', pas ←; schimbăm starea;

Reluăm pas 2a

~~d) Cât timp~~

f) Citim 0, scriem 0, pas →, Stare finală.



$$CT:\ \underbrace{x+y}_{\text{pas 1}} + x\cdot\Big[\underbrace{y+xy}_{2a} + 2 + \underbrace{x}_{2d} + \overbrace{y(1+\underbrace{xy}_{3b}+1+\underbrace{xy}_{3d}+1)}^{P.3} + 1 + \underbrace{y}_{p.4b} + 1\Big] + 1$$

$O((xy)^2)$

COMPLEXITĂȚI

FUNCTIE	COMPLEXITATE
$y \leftarrow x$	$O(x)$
$y \leftarrow x_1 + x_2$	$O(x_1 + x_2)$
$y \leftarrow x_1 x_2$	$O(x_1 x_2)$
$y \leftarrow f_{=}(x_1, x_2)$	$O(\min\{x_1, x_2\})$
$y \leftarrow f_{:2}(x_1, x_2)$	$O(x)$
$y \leftarrow \lvert x_1 - x_2 \rvert$	$O($
$y \leftarrow f_{<}(x_1, x_2)$	$O(\min(x_1, x_2))$
$y \leftarrow x_1^{x_2}$	$O(x_2 \cdot x_1^{x_2})$
$\begin{cases} y_1 \leftarrow x_1 / x_2 \\ y \leftarrow x_1 \% x_2 \end{cases}$	$O(x_1)$
$y \leftarrow x!$	$O(x \cdot x!)$

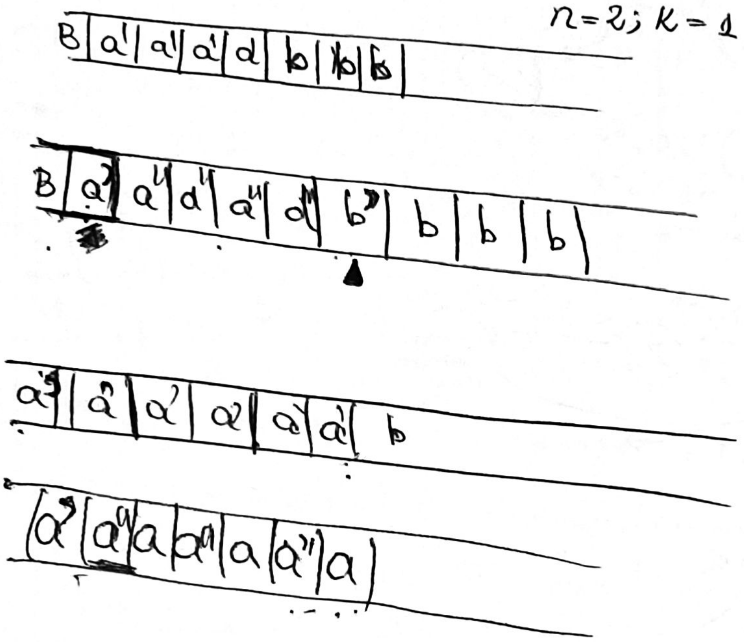
TEST SEMINAR

1. $w = a^* b^*$

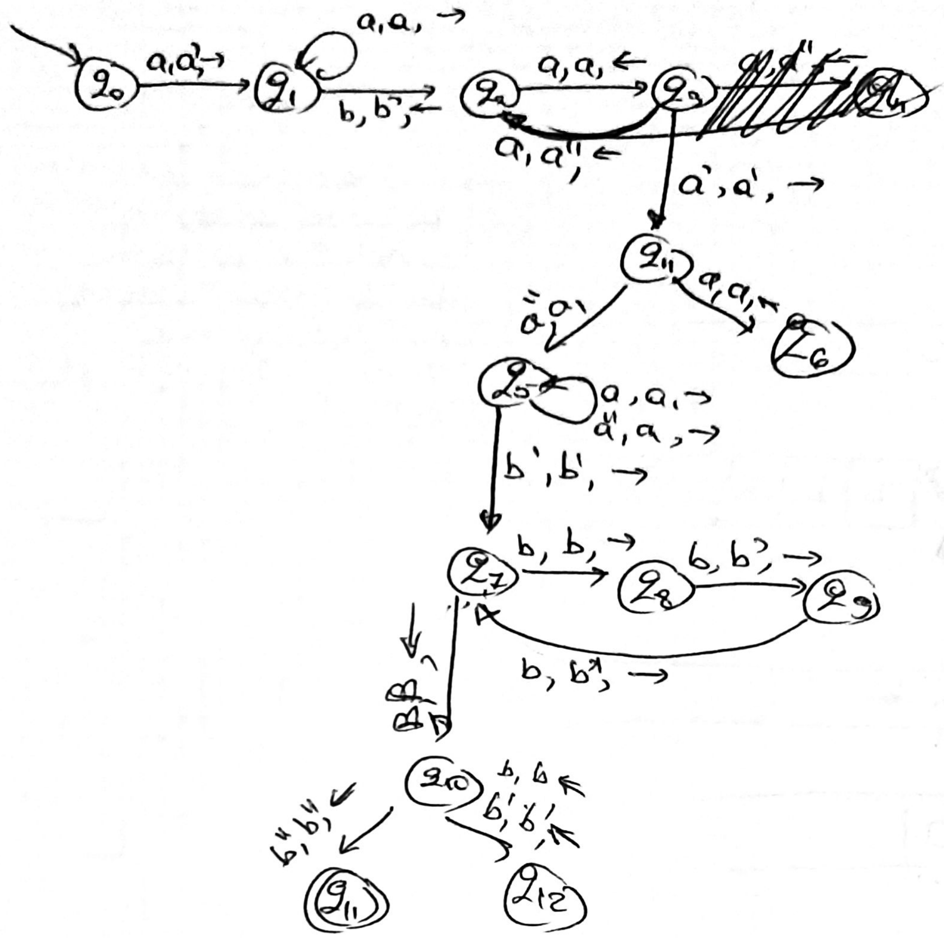
   $w = a^{2n} b^{3k}$

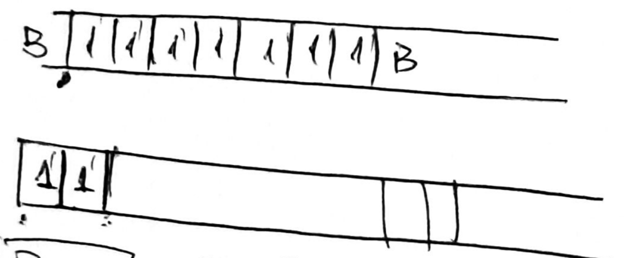
   $n, k > 1$

$n = 2;\ k = 1$



→ $n, k \neq 0$; vom avea pe bandă, în caz că cuvântul e de forma dorită măcar 2 de a și 3 de b; nu mai adaugăm un a/b în plus pe bandă





$x = 7$

$\boxed{\text{Pas 1}}$ Pe B2, scriem 1,

$\boxed{\text{Pas 1}}$ Pe B_2, citim B, scriem 1, pas ←.
Pe B_2, citim B, scriem 1, ~~pas~~ stăm. // y=2

$\boxed{\text{Pas 2}}$ $\boxed{\text{Pas 1}^3}$ – test pt divizibilitate cu 2

a) Cât timp pe ambele benzi citim 1, scriem 1', pas →.

b) Dacă pe B_1 citim B iar pe B_2 citim 1, nu modif, stăm, stare finală

c) Dacă pe B_1 citim 1. iar pe B_2 citim B, ~~stăm~~ nu modif, stăm, sari la pas 3.

d) Dacă pe B_1 citim B și pe B_2 citim B, nu modif, stăm, sari la pas 4

$\boxed{\text{Pas 3}}$ (Demarcăm tot y)

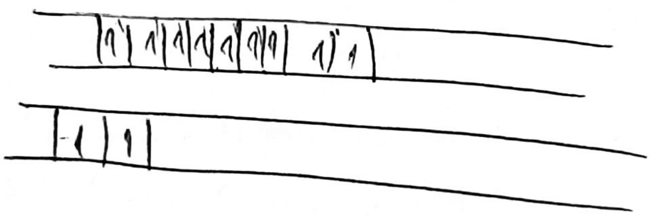
a) Pe B_2, cât timp citim 1', scriem 1, pas ←.

~~b) Pe B_2, citim B, scriem 1, pas ←~~

~~c) Pe B_2, citim B, scriem 1, pas stăm~~ ~~// y = y+2;~~

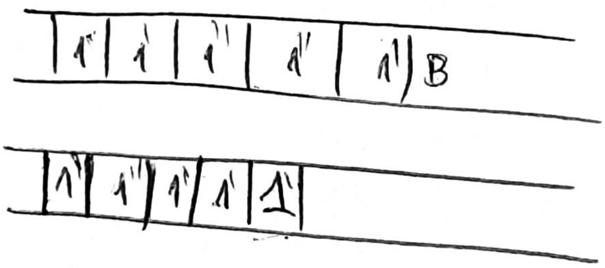
b) Citim B, scriem B, pas →. ~~y impar~~.

Reluăm pas 2a.



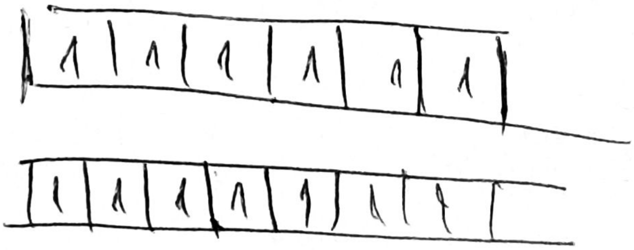
→ ~~Văd dacă se împ~~
Încerc de câte ori pot să-l cuprind pe y în x (demarcând y)

→



→ Test pt 1 și 2.

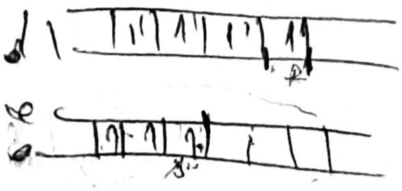
→ Dacă se cuprinde



test egalitate

DA → prim

NU → neprim



CC. - Seminar 1. 04.10.2018

Mașini Turing.

MT = $(Q, \Sigma, \Gamma, B, q_0, F, \delta)$

Q = mulț. stărilor.

Σ = alfabetul de intrare

Γ = alfabetul benzii

$\Sigma \subseteq (\Gamma \setminus \{B\})$

$B \in (\Gamma \setminus \Sigma)$ simbolul "blank".

$q_0 \in Q$ starea inițială.

$F \in Q$ mulț. stări finale

$\delta : (Q \setminus F) \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$

fct. de tranziție.

→ Se dă x în baza 1. Să se deplaseze cu 4 poziții spre dreapta.

$\boxed{x=5}$

Înainte:

...	B	1	1	1	1	1	1	B	...

După:

B	0	0	0	0	1	1	1	1	1	1	B	...

Pas 1: Cât timp citim 1, scriem 1, pas →

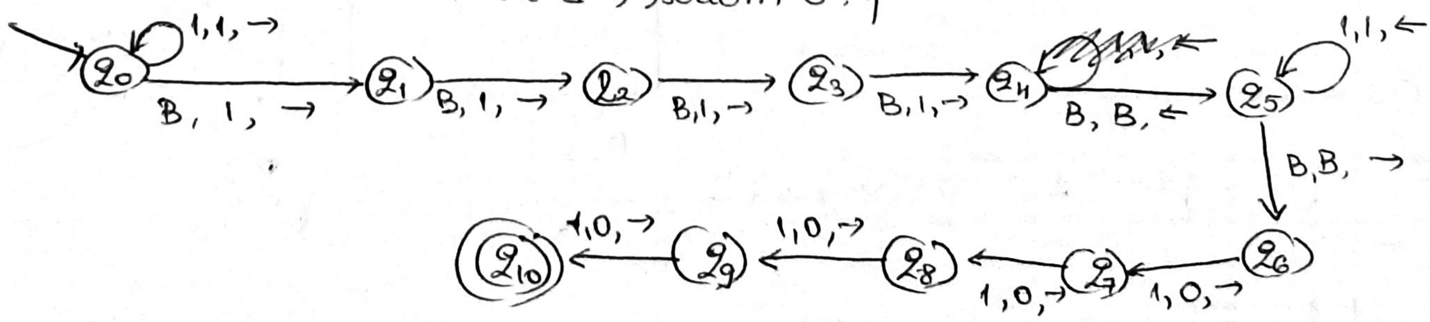
Pas 2: De 4 ori citim B, scriem 1, pas →

Pas 3: Citim B, scriem B, pas ←

Pas 4: Cât timp citim 1, scriem 1, pas ←

Pas 5: Citim B, scriem B, pas →

Pas 6: Citim de 4 ori 1, scriem 0, pas →



→ Se dau x și y în baza 1, delimitate de un 0. Să se calculeze $x+y$.

$x=3$, $y=2$

Înainte

B	1	1	1	1	0	1	1	1	B

După

B	1'	1	1	1	0	1	1	1	2	1	1	1	1	1	1	B

Pas 1: Citim un 1 din x, scriem 1', pas →

Pas 1: Cât timp citim 1 sau 0, nu modificăm banda

Pas 3: Citim blank, scriem 2, pas ←

Pas 4 a: Cât timp citim 0/1/2, nu modificăm banda, pas ←

Copiem x la finalul benzii

b: Citim 1' din x, scriem 1', pas →

c: Citim un 1 din x, scriem 1', pas →

d: Cât timp citim 1/0/2, nu modificăm banda, pas →

e: Citim B, scriem 1, pas ←. Reluam pas 4a.

Pas 5: Citim 0, scriem 0, pas →

Pas 6 a: Citim un 1 din y, scriem 1', pas →

Copiem y la fin. benzii

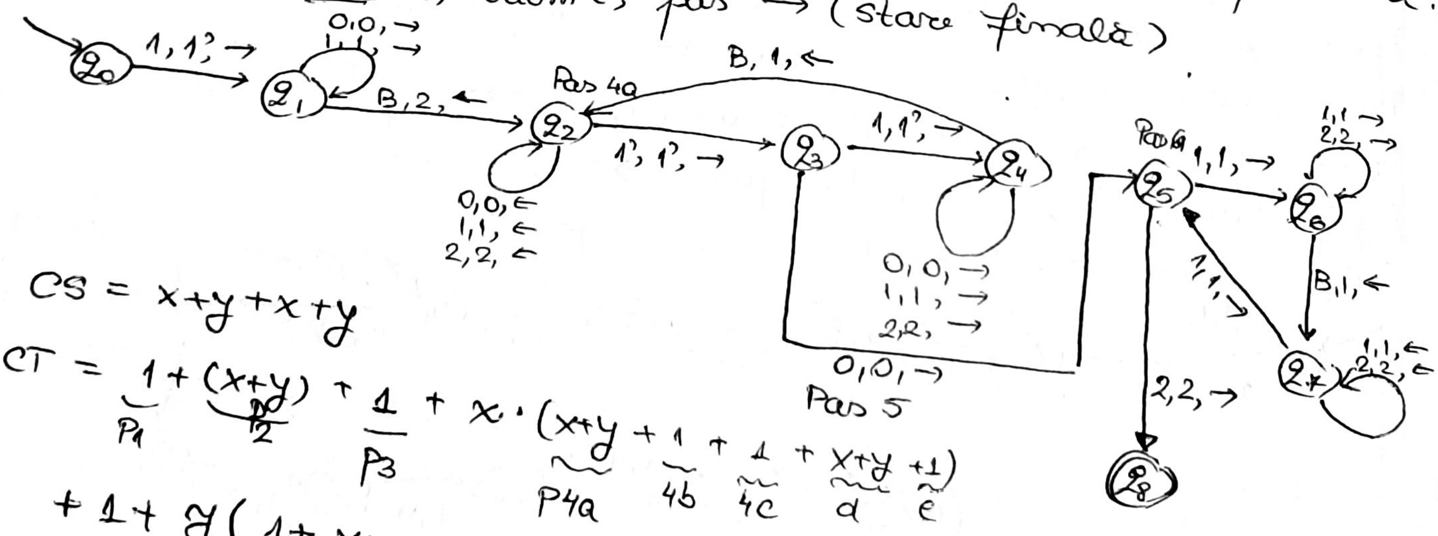
b: Cât timp citim 1 sau 2, nu modificăm banda, pas →

c: Citim B, scriem 1, pas ←

d: Cât timp citim 1 sau 2, nu modif. banda, pas ←

e: Citim 1', scriem 1', pas →, Reluăm pas 6 a.

Pas 7: Citim 2, scriem 2, pas → (stare finală)



$CS = x+y+x+y$

$CT = \underbrace{1}_{P1} + \underbrace{(x+y)}_{P2} + \underbrace{1}_{P3} + x \cdot (\underbrace{x+y}_{P4a} + \underbrace{1}_{4b} + \underbrace{1}_{4c} + \underbrace{x+y}_{d} + \underbrace{1}_{e})$

$+ 1 + y(\underbrace{1}_{6a} + \underbrace{x+y}_{6b} + 1 + x+y + 1) + 1$

$= O((x+y)^2)$

(3) → Se dau x și y în baza 1, delimitate de un 0. Să se calculeze modulul diferenței.

x=3

y=2

Înainte

B	1	1	1	1	0	1	1	1	B

După

B	1'	1	1	1	0	1	1	1	2	1	1	B

Pas 1 a; Citim un 1 din x, scriem 1', pas →
b) Cât timp citim 1 din x, scriem 1, pas →
c) Citim 0, scriem 0, pas →, schimbăm starea
d) Cât timp citim 1' din y, scriem 1', pas →
e) Citim un 1 din y, scriem 1', pas ←
f) Cât timp citim 1' din y, scriem 1', pas ←
g) Citim 0, scriem 0, pas ←, schimbăm starea
h) Cât timp citim 1 din x, scriem 1, pas ←
i) Citim un 1' din x, scriem 1', pas →, reluăm pas 1 a

Pas 2a: Citim 0, scriem 0, pas → ; pas 3 (x ≤ y)
b) Citim B, scriem 2, pas → ; citim B, scriem 1, pas →, citim B, scriem 1, pas ← ; pas 4 (x>y)

Pas 3 a: Cât timp citim 1 sau 1', nu modif. banda, pas →
b: Citim B, scriem 2; pas → Citim B, scriem 1, pas ←
c: Cât timp citim 1 sau 2, nu modif, pas ←.
d: Citim 1' din y, scriem 1', pas →
e: Citim un 1 din y, scriem 1', pas →
f: Cât timp citim 1 sau 2, nu modif, pas →
g: Citim B, scriem 1, pas ←, reluam Pas 3 C

Pas 4 Ne întoarcem în stânga până la partea nemarcată din x și o copiem la finalul benzii

Pas 5: Citim 2, scriem 2, pas →, stare finală

Când nu se mai poate îndeplini 3 c.

Sub1 Mașini Turing & Gramatici oarecare

Def Mașina Turing = un 7-tuplu $M = (Q, V, U, \delta, q_0, B, F)$ unde:

- Q = mulțimea stărilor
- V = alfabetul de input
- U = alfabetul benzii, include V (mașina poate modifica inputul)
- δ = funcția de tranziție
- q_0 = starea inițială
- B = simbolul blank
- F = mulțimea stărilor finale

→ MT cu o sg bandă: $\delta : Q \times U \to P(Q \times U \times \{L, R\})$ (left, right)

(q, a) conține (s, b, x) – dacă mașina se află în starea q și citește pe bandă simbolul a, atunci:

1) înlocuiește simbolul a cu simbolul b
2) deplasează capul de citire în direcția dată de x
3) schimbă starea q în starea s

→ deterministă = o MT se numește deterministă dacă $|\delta(q,a)| \leq 1$ $\forall q \in Q$ și $a \in U$. Altfel, mașina se numește nedeterministă.

→ MT cu mai multe benzi

$\delta : Q \times U^n \to P(Q \times U^n \times \{L, R\}^n)$

$(q, a_1, a_2, \ldots, a_n)$ conține $(s, b_1, b_2 \ldots, b_n, x_1, x_2, \ldots, x_n)$; dacă mașina se află în starea q și citește simbolul a_1 pe banda 1, a_2 pe banda 2, ..., a_n pe banda n, atunci:

1. Scrie b_1 în locul lui a_1, .., b_n în locul lui a_n
2. Deplasează capul de citire coresp. benzii i în direcția x_i.
3. Schimbă starea q în starea s

→ deterministă = o MT s.n deterministă dacă $|\delta(q, a_1, .., a_n)| \leq 1$ pt $\forall q$ și $\forall a_i$, $i = \overline{1,n}$. Altfel, mașina Turing s.n. nedeterministă.

Mașina Turing poate fi:

→ dispozitiv de acceptare

$L(M) = \{w \in V^* \mid w$ acceptat de $M\}$

→ dispozitiv de calcul

INTRARE: $1^{x_1+1} 0 1^{x_2+1} 0 \ldots 0 1^{x_k+1}$

IEȘIRE: $1^{f(x_1, x_2, \ldots, x_k)+1}$; $f : N^* \to N$

Dacă se poate construi o astfel de mașină pt fct f, atunci f s.n TURING CALCULABILĂ.

Def

Gramatică = un sistem (T, N, S, P) unde:
T = alfabetul terminalelor
N = alfabetul neterminalelor
S - simbolul de start
P - mulțimea producțiilor

$L(G) = \{ w \mid w \in T^*, S \xrightarrow{*} w \}$

TEOREME

1) O mașină Turing cu $\underline{n}$ benzi este echivalentă cu o m
cu o singură bandă.

2) O mașină Turing nedeterministă cu o bandă este ec
cu o mașină deterministă cu 3 benzi.

3) Pentru orice gramatică G există o mașină Turing M
$L(M) = L(G)$

4) Pentru orice mașină Turing M există o gramatică G aî
$L(G) = L(M)$

Dem

→ Pt orice MT M cu k benzi, există o MT M' cu o singură
banda echivalentă cu M.

În plus, dacă M deterministă, M' este și ea deterministă.

→ Construim M' o mașină Turing aî:

- M' are o singură bandă, pe care o împărțim în piste. Fiecare set de două piste corespund unei benzi din mașina M.
- pe pista $2*i-1$ se află conținutul benzii i a mașinii M.
- pista $2*i$ conține 0-uri mai puțin pe o poziție, care are 1 unde se află capul de citire al benzii i al mașinii M.
- mașina M' citește conținutul benzii de la stg la dreapta și memorează simbolurile de pe pistele impare aflate imediat deasupra simbolurilor de 1 de pe pistele pare.
- când ajunge la finalul, simulează mișcările pe care le-ar fi făcut mașina M. Parcurge din nou banda, dar de la dreapta la stânga și actualizează conținutul pistelor impare în conformitate cu simbolurile care ar fi fost scrise de mașina M și conținutul pistelor pare în conformitate cu direcția în care s-ar fi deplasa fiecare cap de citire.

3 22 11 31

→ Pentru orice mașină Turing M nedeterministă cu o bandă, există o mașină Turing cu 3 benzi, M', deterministă, echivalentă cu M.

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

Fie $E = \{(q, a, s, b, x)\}$ unde
- → q, s sunt stari $\in Q$
- → $q \notin F$
- → $a, b \in U$ și $b \notin B$
- → $x \in \{L, R\}$ dir. de deplasare

Daca mașina se află în starea q și citește simbolul a, atunci scrie b, deplasează capul de citire în direcția indicată de x și trece în starea s.

Mașina M' are 3 benzi și lucrează astfel:

1. Copiază w pe banda 3
2. Generează pe banda 2 succesorul cuvântului din E^* înscris pe bandă
3. Citește simbolul de pe banda 2 (q, a, s, b, x)

   a) Verifică dacă mașina se află în starea q. Dacă nu, merge la pas 5.

   b) Verifică dacă simbolul citit de pe banda 1 este a. Dacă nu, merge la pas 5.

   c) Scrie b peste a.

   d) Schimbă starea în S

   e) Deplasează capul de citire/scriere pe banda 1 în direcția x.

4. Verificăm dacă starea obținută e finală; dacă da ne oprim. Altfel, mergem la pasul 5.

5. Copiază w de pe banda 3 pe banda 1. Merge în starea q_0 și merge la pasul 2.

Sub2 Funcții recursive, Turing calculabile, Calculabile cu programe standard

Def

1. Funcții recursive = o funcție se obține din funcții elementare prin aplicarea operațiilor de compunere, recurență primitivă și minimizare nemărginită.

FUNCȚII ELEMENTARE
- succesor : $succ(x) = x+1$
- constante: $C_K^{(n)}(x_1, \dots, x_n) = k$
- proiecții : $\Pi_K^{(n)}(x_1, \dots, x_n) = x_K$.

COMPUNEREA FUNCȚIONALĂ $g_i : \mathbb{N}^K \to \mathbb{N}$; $h : \mathbb{N}^n \to \mathbb{N}$ primitiv recursive,

atunci $f : \mathbb{N}^K \to \mathbb{N}$ primitiv recursive $f(x_1, \dots x_K) = h(g_1(x_1, \dots, x_K), \dots, g_n(x_1, \dots, x_K))$

RECURENȚĂ PRIMITIVĂ

$f : \mathbb{N}^{K+1} \to \mathbb{N}$ este obținută prin recurență primitivă din fct $g : \mathbb{N}^K \to \mathbb{N}$ și $h : \mathbb{N}^{K+2} \to \mathbb{N}$ dacă:

a) $f(x_1, x_2, \dots, x_K, 0) = g(x_1, \dots, x_K)$

b) $f(x_1, \dots, x_K, t+1) = h(x_1, \dots, x_K, t, f(x_1, \dots, x_K, t))$

MINIMIZARE NEMĂRGINITĂ

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

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

2. Funcții Turing Calculabile = o funcție s.n Turing calculabilă dacă există o mașină Turing ce are ca input $1^{x_1+1} 0 \, 1^{x_2+1} 0 \dots 0 \, 1^{x_K+1}$ și ca output $1^{f(x_1, x_2, \dots, x_K)+1}$

3. Programe standard

Limbajul abstract $\mathcal{S}$ va calcula $f : \mathbb{N}^K \to \mathbb{N}$

variabile
- de intrare $x_1, \dots, x_n$
- de ieșire y
- locale $z_1, z_2, \dots, z_m$

variabilele de lucru și cea de ieșire sunt inițializate cu $\underline{\underline{0}}$

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

INSTRUCȚIUNI

- $V \leftarrow V$ (efect nul)
- $V \leftarrow V+1$ (incrementare)
- $V \leftarrow V \dot{-} 1$ (decrementare dacă $V \neq 0$, nul altfel)

IF $V \neq 0$ GOTO L ($V>0$ transferul se face astfel:
→ la prima instrucțiune cu eticheta L, d
$L \neq E$ și există L
→ se termină programul dacă $L = E$ sau

Programul standard este format dintr-un set de instrucțiu
iar terminarea se face printr-un salt la eticheta E, fie pri
salt la o etichetă inexistentă, fie transferul se face la
sfârșitul instrucțiunilor.

Outputul e valoarea lui y de la sfârșitul programului.

TEOREME

1) Orice funcție calculabilă cu programe standard este Turing calculabilă
2) Orice funcție recursivă este calculabilă cu programe stand
3) Orice funcție Turing calculabilă este recursivă.

Dem) Orice funcție Turing calculabilă este recursivă.

$\overline{x_1}$	0	$\overline{x_2}$	0	$\overline{x_3}$	...		0	$\overline{x_k}$	0	

↓

$\overline{f(x_1, x_2, \dots, x_k)}$	0 0 ... 0 B

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

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

Configurație: $\alpha\, q\, \beta$

→ numerotăm celulele benzii 0, 1, 2...
→ numerotăm stările din Q $q_0 = 0, q_1 = 1, \dots, q_i = i$.
→ numerotăm simbolurile din U: $0 = 0$, $1 = 1$. Numărul atașat unei configurări

$q \to i$

$|\alpha| = j$ → poziția pe ~~cap~~ bandă a capului este j

→ Conținutul benzii : $\alpha\beta = s_1 s_2 \dots s_n$ → nr. atașat este

$p_1^{\#s_1} p_2^{\#s_2} \dots p_n^{\#s_n}$ unde $\#s_k$ este nr. simbolurilor lui s_k.

p - poziția capului de citire

→ Configurația $\langle x, \langle y, z \rangle \rangle$ → x = nr. de stări
→ y = poziția capului I/O
→ z = nr. atașat conținutului benzii

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

$C_M(x, 0)$, $x = (x_1, \dots, x_m)$ = nr. atașat configurației la pasul 0 al mașinii M pe intrarea(x).

Ne propunem să demonstrăm că $C_M(x, n)$ este fct. recursivă

$$C_M(x, 0) = \langle 0, \langle 0, p_1 p_2 p_{x_1+1} p_{x_1+2} \dots p_{x_1+x_2+2} \dots p_{x_1+\dots+x_k} \rangle \rangle$$

$h_1(z) = \begin{cases} \text{nr. atașat în care trece mT din configurația curentă, cu nr. } z\text{, dacă } z \text{ este nr. unei configurații valide} \\ \uparrow\text{, altfel (nedefinită)} \end{cases}$

$h_2(z) = \begin{cases} \text{poziția capului de citire/scriere în care ajunge mT din config. curentă dacă } z \text{ este nr. unei config. valide} \\ \uparrow \text{altfel} \end{cases}$

$h_3(z) = \begin{cases} \text{nr. atașat conținutului benzii în care trece mT din config cu nr. } z\text{, dacă } z \text{ este nr. unei config. valide} \\ \uparrow \text{altfel} \end{cases}$

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

h_1, h_2, h_3 recursive?

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

$g_2(a,b) = \begin{cases} 0\text{, M se deplasează la stg. din starea cu nr. a citind simbolul cu nr b} \\ 2\text{, M " — " dreapta " — "} \end{cases}$

$g_3(a,b) = \begin{cases} \text{nr de simboluri scris de M din starea cu nr. a citind simbolul cu nr. b} \end{cases}$

g_1, g_2, g_3 – funcții de suport finit ⇒ g_1, g_2, g_3 recursive

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

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

$$h_3(z) = \frac{r(r(z))}{P_{\ell(r(z))}^{(r(r(z)))_{\ell(r(z))}}} \cdot P_{\ell(r(z))}^{g_3(\ell(z),\, r(r(z)))}$$

h_1, h_2, h_3 funcții recursive $\rightarrow C_M(x, n)$ recursivă

$C_M(x, n_0) \rightarrow M$ se oprește după n_0 pași

$C_M(x, n)$ $\forall n > 0$

$\mu(x) = \min\limits_{n} \left[C_M(x, n) = C_M(x, n+1) \right]$ = nr de pași după care M se oprește pe intrarea x

$f(x) = L_t\left(r(r(C_M(x, \mu_x)))\right) \dot{-} 1$

$f(x_1, \dots, x_k) \Rightarrow f$ fct. recursivă

Orice fct. TURING calc. e recursivă

$<x,y> = 2^x(2y+1)-1$

- $l(z) = x$ dacă $\exists y$ cu prop $<x,y> = z$
- $r(z) = y$ dacă $\exists x$ cu prop $<x,y> = z$
- p_n – al n-lea nr prim.

$[a_1, a_2, \dots, a_k] = p_1^{a_1} \cdot p_2^{a_2} \cdot \dots \cdot p_k^{a_k}$

→ $f: \mathbb{N}^k \to \mathbb{N}$ Turing calculabilă ⇒ $\exists$ mT M a.î.

$M = (Q, V, U, \delta, q_0, B, F)$ care calculează fct f

→ Renumerotăm stările $Q = \{q_0, \dots, q_r\} = \{0, 1, \dots, r\}$

Renumerotăm alfabetul

$V = \{0, 1, B, \dots\}$ cu $|U| = n \Rightarrow U = \{s_0, s_1, \dots, s_n\}$ →

→ $U = \{0, 1, \dots, n\}$

→ O config M e formată din:
- starea q
- poziția capului de citire/scriere
- conținutul benzii: $s_{i_1}, s_{i_2}, \dots, s_{i_k}$

Deci vom numi configurație:

$z = <\#q, <p, [s_1, s_2, \dots, s_k]>>$

Definim $C_M: \mathbb{N}^{k+1} \to \mathbb{N}$, $C_M(x,n)$ – nr. atașat config. MT M pe intrarea x la pasul n.

$C_M(x,0) = <0, <0, [p_1, p_2, \dots, p(x_1+1) \dots, p(x_1+x_2+2), \dots]>>$

↳ starea 0, poziția capului la începutul benzii, iar conținutul benzii e chiar inputul

Ne propunem să dem. că $C_M(x,n)$ e fct. recursivă.

$h_1(z) = \begin{cases} \text{nr atașat } \underline{\text{stării}} \text{ în care trece mT din configurația curentă, cu nr. } z, \text{ dacă } z = \text{nr. unei config. valide} \\ \uparrow \text{ altfel} \end{cases}$

$h_2(z) = \begin{cases} \text{nr. atașat } \underline{\text{poz capului de citire/scriere}} \text{ în care trece MT din config crt., dacă } z = \text{nr. config valid} \\ \uparrow \text{ altfel} \end{cases}$

$h_3(z) = \begin{cases} \text{nr. atașat } \underline{\text{conținutului benzii}} \text{ în care trece MT din config. } z, \text{ dacă } z = \text{nr. unei config valide} \\ \uparrow \text{ altfel} \end{cases}$

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

h_1, h_2, h_3 recursive = ?

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

$g_2(a,b) = \begin{cases} 0, & \text{M se deplasează la stg. din starea cu nr a citind simbolul cu nr b.} \\ 2, & \text{M se deplasează la dreapta " "} \end{cases}$

$g_3(a,b)$ = nr. de simboluri scris de M din starea cu nr a citind simbolul cu nr b

g_1, g_2, g_3 funcții de suport finit ⟹ g_1, g_2, g_3 recursive

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

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

$$h_3(z) = \frac{r(r(z))}{P_{\ell(r(z))}^{r(r(z))_{\ell(r(z))}}} \cdot P_{\ell(r(z))}^{g_3(\ell(z), r(r(z)))}$$

h_1, h_2, h_3 fcții recursive → $C_M(x,n)$ recursivă

$C_M(x, n_0)$ → M se oprește după n_0 pași

$C_M(x,n) \ \forall n > 0$

$\mu(x) = \min\limits_n \left[C_M(x,n) = C_M(x, n+1)\right]$ = nr. de pași după care M se oprește pe x.

$$f(x) = L_t\left(r(r(C_M(x, \mu x)))\right) - 1$$

$\shortparallel$

$f(x_1, ..., x_k)$

Sub 3 MULȚIMI RECURSIVE

Def

→ Spunem că limbajul L este recursiv (REC) dacă una din următoarele condiții este îndeplinită

- fie $f(x) = (x \in L)\ ?\ 1 : 0$ → f e recursivă
  → f e calculabilă cu P.S.

- există o mașină Turing M care se oprește pe fiecare intrare și acceptă L.

→ Spunem că L este recursiv enumerabil (RE) dacă ∃ o mașină Turing care acceptă L (dar care poate să nu se oprească pe fiecare intrare).

→ Spunem că L este nerecursiv enumerabil (NRE) dacă ∄ nicio mașină Turing care accepta L.

Atunci, putem defini cele 3 mulțimi:

REC = { L / L limbaj recursiv }

RE = REC ∪ { L limbaj recursiv enumerabil }

NRE = RE ∪ { L limbaj <u>nerecursiv</u> enumerabil }

→ O mulțime X s.n recursiv enumerabilă dacă fct. ei caracteristică este Turing calculabilă

→ O mulțime X s.n recursivă dacă fct. ei caracteristică este Turing calculabilă, dar mașina Turing se oprește pe fiecare intrare.

Th
(1) Dacă A - mulț recursivă ⇒ A recursiv enumerabilă

(2) a) X mulțime recursivă ⇒ X și C_X sunt rec. enumerabile

b) X & Y recursive sau rec. enumerabile ⇒ X∪Y sau X∩Y sunt recursive sau recursive enumerabile

(3) Orice prop. netrivială pt RE este nedecidabilă (Rice)

Prop & Relații :
1. REC închisă la ∩, ∪, C
2. RE închisă la ∩, ∪
3. RE nu e închisă la C

$L, C(L) \in RE \Leftrightarrow L \in REC$

$REC \subset RE \subset NRE$

(Dem)

(1) REC închisă la $\cap, \cup, c$

- Fie $L_1, L_2 \in REC$. Atunci $\exists$ mașina turing M_i cu $L(M_i)$, de care se oprește pe fiecare input, unde $i = \{1, 2\}$

1. Construim mașina Turing $M\cap$:
   - input: w
   - simulează mașina M_1 pe intrarea w:
     - M_1 respinge $\Rightarrow$ $M\cap$ respinge
     - M_1 acceptă $\Rightarrow$ simulează M_2 pe intrarea w:
       - M_2 acceptă $\Rightarrow$ $M\cap$ acceptă
       - M_2 respinge $\Rightarrow$ $M\cap$ respinge.

- Fie $w \in L(M\cap) \Leftrightarrow$ w e acceptat și de M_1 și de $M_2 \Leftrightarrow$ $w \in L(M_1) \cap L(M_2) \Leftrightarrow w \in L_1 \cap L_2$ deci $L_1 \cap L_2$ este limbaj recursiv.

2. Construim mașina Turing $M_\cup$:
   - input: w
   - simulează M_1 pe intrarea w:
     - M_1 acceptă $\Rightarrow$ $M\cup$ acceptă
     - M_1 respinge $\Rightarrow$ simulează M_2 pe intrarea w:
       - M_2 acceptă $\Rightarrow$ $M_\cup$ acceptă
       - M_2 respinge $\Rightarrow$ $M_\cup$ respinge.

- Fie $w \in L(M_\cup) \Leftrightarrow$ w e acceptat de M_1 sau de $M_2 \Leftrightarrow$ $w \in L(M_1) \cup L(M_2) \Leftrightarrow w \in L_1 \cup L_2$. Deci $L_1 \cup L_2$ este limbaj recursiv

3. Construim mașina Turing $MC(L_1)$:
   - input: w
   - simulează mașina M_1 pe intrarea w:
     - M_1 acceptă $\Rightarrow$ $MC(L_1)$ respinge
     - M_1 respinge $\Rightarrow$ $MC(L_1)$ acceptă

Fie $w \in L(M_C(L_1)) \Leftrightarrow$ w nu e acceptat de $M_1 \Leftrightarrow$ $w \notin L(M_1) \Leftrightarrow w \notin L_1 \Leftrightarrow w \in C(L_1)$. Deci $C(L_1)$ este limbaj recursiv.

② Th. Rice : Orice proprietate netrivială pe RE e nedecidabilă

M_i) decidabil = $\exists M$ maşină Turing care se opreşte pe fiecare intrare şi rezolvă problema P. (P e decidabilă dacă L_p e recursiv)

RA : Pp că L_p e recursiv $\Rightarrow \exists M_p$ aî $\begin{cases} L(M_p) = L_p \\ M_p \text{ se opreşte} \end{cases}$

$\boxed{\text{CAZUL 1}}$

$\emptyset \notin P$.

Fie $L \in P$ aî $L = M_L$

Fie M şi w arbitrar, fixate.

Construim M' aî $L(M') = \begin{cases} L, \text{ dacă } w \in L(M) \\ \emptyset \text{ altfel} \end{cases}$

- INPUT: x
- ignorăm intrarea x şi simulează M pe intrarea w.
  Dacă M acceptă, M' revine la începutul său adică x.
- Simulează pe M_L pe intrarea x
  Acceptă $\Leftrightarrow$ M_L acceptă.

2

Construim M_0:

- INPUT: $\#(M)\$w$
- CALCULEAZĂ: $\#\widehat{(M')}$, M' e maşina construită anterior pt. perechea M, w (de pe rândul anterior).
- SIMULEAZĂ M_p pe intrarea $\#(M')$
  Acceptă $\Leftrightarrow$ M_p acceptă

M_0 se opreşte pe fiecare intrare

$$L(M_0) = \{ \#(M)\$w \mid M_p \text{ acceptă } \#(M') \} =$$
$$= \{ \#(M)\$w \mid \#(M') \in L_p \}$$
$$= \{ \#(M)\$w \mid L(M') \in P \}$$
$$\Updownarrow$$
$$L(M') = L \Leftrightarrow w \in L(M)$$

$$L(M_0) = \{ \#(M)\$w \mid w \in L(M) \} = L_U \text{ ⨯}$$

$\boxed{\text{CAZUL 2}}$

$\emptyset \in P \Rightarrow \emptyset \notin C_p$ şi se aplică cazul 1 (iau complementul lui P) $\Rightarrow \emptyset$ nu are proprietatea $\overline{P}$

Sub 4: Funcția universală / Program universal. Problema opririi

→ Funcția universală. Program universal

$\phi^{(n)}(x_1, ..., x_n, t)$ - funcția universală de n variabile

$\forall n \geq 1$, această funcție este calculabilă

→ această funcție primește ca argumente variabilele de intrare ale unui program standard $(x_1, ..., x_n)$ și codificarea acestui program t.

→ simbolul "#" înseamnă codificare. Codificarea unui program P este de forma $\#(P) = [\#(I_1), \#(I_2), ..., \#(I_n)]$

$\#(P)$ este argumentul "t" din prototipul funcției ϕ

$\#(I)$ reprezintă codificarea unei instrucțiuni și este de forma:

$\#(I) = \langle a, \langle b, c \rangle \rangle$ unde
- a - eticheta
- b - nr. variabilei in S
- c - tipul instrucțiunii

→ a = 0 dacă instr. I nu are etichetă; $\#(L)$ dacă are etichetă

Etichetele sunt codificate:
E = 1, $A_1 = 2$, $A_2 = 3$ .... Deci, de exemplu, $\#(A_3) = 4$.

→ b = nr. variabilei care apare in I

→ c = 0 corespunde lui $V \leftarrow V$

c = 1 " ——— " $V \leftarrow V+1$

c = 2 " ——— " $V \leftarrow V-1$

$c = \#(L) + 2$; este necesar "+2" pt. a se evita confuzia cu celelalte instrucțiuni;
→ coresp. lui IF V != 0 GOTO L

Teoremă: $\phi^{(n+1)}$ e calculabilă cu programe standard

S - produs al tuturor variabilelor din progr. simulat; fol. pt a stabili că V = 0

V - variabila instr. curente.

L (T) - return. nr. instrucțiuni de rulat

(T) K - return. nr. instrucțiunii car trb. rulată

J - instr. de rulat

U - tipul instrucțiunii care trb. rulată

P_V - al v-lea nr prim

r și l - funcții rec. care returnează $\langle a, \langle b, c \rangle \rangle$ adică codificarea instr.

$T \leftarrow x_{n+1} + 1$

$S \leftarrow \prod_{i=1}^{m} P_{2i}^{x_i}$ // memorează val. tuturor variabilelor pr

```
C:  IF (K = 0) || (K > L(T)) GOTO E1
    J = r(r((T)_K))
    V ← l(r((T)_K))
    IF i = 0  GOTO N
    IF i = 1  GOTO A
    IF i = 2  GOTO M
    IF (P_V ∤ S) GOTO N          // P_V nu divide S
    K ← min_r [(l((T)_r) = J-2) & (J-2) > 1]
    GOTO C
A:  S ← S * P_V
    GOTO N
M:  IF (P_V ∤ S) GOTO N
    S ← S / P_V
N:  K ← K+1
    GOTO C
E1: Y ← (S)_1
GOTO E
```

→ Problema opririi.

ENUNT: Dată codificarea unui program #(P), notată cu y și o intrare, pentru acest program, notată cu x, se poate calcula funcția / predicatul HALT(x,y)?

unde HALT(x,y) = 1 dacă programul se termină, 0 dacă programul nu se termină

→ predicat = fct. care returnează 0 sau 1

Predicatul HALT nu se poate calcula.

Dem RA: Pp. ca HALT este calculabil.

Consider următorul program, P1: Z[r+1] ← HALT(x,x)

A: IF Z[r+1] ≠ 0 GOTO A

Z[r+1] este în principiu prima variabilă de lucru liberă.

Programul ciclează la infinit.

Codific P1 cu #(P1) = t.

Rulăm HALT cu argumentele x și t

Se va obține contradicție: Z va fi 0 dacă programul x cu intrarea x nu se oprește; dar cum x este t, obțin

HALT(t, t) ⟺ not(HALT(t, t))

/

Sub 5: Complexitate timp

→ Pentru calculul complexitatii timp se foloseste modelul masinii Turing cu k benzi infinite la ambele capete.

Poate să nu miște capul de citire - scriere la un moment dat. Masinile considerate se opresc la fiecare input.

Def

1. $TIME_M(n)$ = nr. maxim de pași pe care - l face masina M pentru a decide o intrare de lungime n

2. $(N)(D)TIME_K(f(n)) = \{ L \mid \exists M$ o mașină Turing nedeterministă sau deterministă cu k benzi a.î. $L(M) = L$ și $TIME_M(n) \leq f(n)\ \forall n \geq n_0 \}$

↓

mulțimea limbajelor care au un algoritm care rezolvă problema și a cărei timp e majorat de funcția f

Obs: Dacă k dispare → putem avea oricâte benzi.

3. O funcție $f(n)$ s.n. timp construibilă dacă $\exists$ o mT M și un n_0 a.î. $TIME_M(n) - f(n)\ \forall n \geq n_0$.

4. O funcție $f(n)$ s.n timp construibilă complet dacă $\exists$ o mT M a.î. $TIME_M(n) = f(n)\ \forall n \geq n_0$.

TEOREME

1. Eliminarea constantelor

$\lim_{n \to \infty} \frac{f(n)}{n} = \infty \to (N)(D)TIME_K(f(n)) = (N)(D)TIME_K(c \cdot f(n))$

unde c este o constantă pozitivă nenulă dacă $K \geq 2$.

$(D)(N)TIME_K(c \cdot n) \subseteq (D)(N)TIME_K((1+\varepsilon) n)\ \forall K \geq 2, \varepsilon > 0$

2. Reducerea nr. de benzi

$(D)(N)TIME_K(f(n)) \Rightarrow (D)(N)TIME_1(f^2(n))\ \forall K \geq 2$ și $\forall f$

$(D)(N)TIME_K(f(n)) \subseteq (D)(N)TIME_2(f(n)\log(f(n)))\ \forall K \geq 2\ \&\ \forall f$

8. Ierarhia claselor & relațiile între clase de complexit.

$(N)(D)TIME_K(f(n)) \subseteq (N)(D)TIME_2(f(n) \log_2 f(n))$

$(N)(D)TIME_K(f(n)) \subseteq (D)(N)SPACE(f(n))$

→ Dacă $f(n) \geq \log_2 n \rightarrow \forall L \in DSPACE(f(n)) \ \exists C_L$ a.î. $L \in DTIME$

→ $\forall L \in NTIME(f(n)) \ \exists C_L$ a.î. $L \in DTIME(C_L^{f(n)})$

→ Dacă $f(n) \geq \log n$ și $f(n)$ e spațiu construibil complet ⇒ $NSPACE(f(n)) \subseteq DSPACE(f(n))$

Dem Reducerea nr. de benzi: $(D)(N)TIME_K(f(n)) \subseteq (D)(N)TIME_1(f^2(n))$

→ Fie mT M cu $TIME_M(n) = f(n)$;

→ Construim mașina M' astfel:

- M' are o singură bandă auxiliară, iar elementele ei vor fi vectori cu $2K$ piste:
  - pe pista $2*i-1$ se află conținutul benzii i a mT M
  - pista $2*i$ conține 0-uri mai puțin pe o poziție, are 1 unde se află capul de I/O a benzii i a mT M

O pistă poate avea în cel mai rău caz $f(n)$ celule ocupate (deoarece $TIME_M(n) = f(n)$, M nu are timp să ocupe mai mult de $f(n)$ celule pe una din benzile ei)

→ Mașina M':

Citește conținutul benzii de la stânga la dreapta și memorează de pe pistele $2*i-1$ aflate imediat deasupra simbolurilor de pe pistele $2*i$ (maxim $f(n)$ pași)...

Actualizează conținutul benzii de la dreapta la stânga: parcurgerea fiecărei celule: maxim $f(n)$ pași.

- actualizează simbolurile de pe celula i: 1 pas.
- dacă un cap de citire/scriere aflat pe poziția i al mașinii M se mută la dreapta, trebuie actualizate pistele pare de pe celula din dreapta.

— 1 pas ca să ne mutăm la dreapta cu 1 pas.
— un pas ca să ne întoarcem

$\Rightarrow 3 \cdot f(n)$ pași

Deci sunt $f(n) + 3f(n) = 4f(n)$ pași care pot fi executați de maxim $f(n)$ ori $\Rightarrow 4f(n) \cdot f(n)$.

Nu știm dacă putem aplica th. constantelor pt. că nu știm dacă f este supra liniară.

Dar putem construi Mașina M'' cu $L(M'') = L$ care face maxim $\frac{f(n)}{2}$ pași.

Atunci mașina M' poate simula în același mod mașina M'' și va face pași.

→ Eliminarea constantelor

$(N)(D)TIME_K(f(n)) = (N)(D)TIME_K(c \cdot f(n))$

Avem o mașină Turing M_1 cu k benzi și $TIME_{M_1}(n) \leq f(n)$.

<u>PAS 1</u>: Copiez cele r simboluri pe banda 2 ca un simbol $\Rightarrow$ $TIMP = n+1$.

<u>PAS 2</u>: Se poziționează la începutul benzii 2 $\Rightarrow \left[\frac{n}{r}\right] + 1 = Timp$.

Corespondența între benzile lui M_2 și M_1: banda$_1(M_1) \rightarrow$ banda$_2(M_2)$
banda$_2(M_1) \rightarrow$ banda$_1(M_2)$
$\forall j = \overline{3,k}$ banda$_j(M_1) \rightarrow$ banda$_j(M_2)$

- Fiecare simbol de pe benzile lui M_2 codifică r simboluri adiacente de pe banda corespunzătoare în M_1.
- necesitatea $K > 2$ ⟶ pt a interschimba primele 2 benzi
- M_2 calculează conținutul blocului curent de dimensiune r de pe fiecare bandă și conținutul blocurilor adiacente $\Rightarrow$ $\Rightarrow$ Timp 4 (în 4 pași am găsit informația necesară).
- M_2 actualizează celulele curente și vecinele acestora când unul dintre capetele de citire / scriere ale lui M_1 încearcă să depășească cele 3 blocuri adiacente de pe una din benzile lui M_1 → Timp 4

$$TIME_{M_2}(n) \leq n + \frac{n}{r} + 8\,\frac{TIME_M(n)}{r} + 3$$

Dacă $n \geq 3 \Rightarrow n+3 \leq 2n$

$(\forall)\, d\ \exists\, n_d$ a.î. $\forall n \geq n_d \Rightarrow \frac{TIME_{M_1}(n)}{n} \geq d \Rightarrow n \leq \frac{TIME_{M_1}(n)}{d}$

$$TIME_{M_2}(n) \leq 2n + \frac{n}{r} + 8\,\frac{TIME_{M_1}(n)}{r} \leq 2\,\frac{TIME_{M_1}(n)}{d} +$$

$$+ \frac{TIME_{M_1}(n)}{dr} + \frac{8\,TIME(n)}{r} = TIME_{M_1}(n)\left(\frac{2}{d} + \frac{1}{dr} + \frac{8}{r}\right)$$

Alegem $d = \frac{r}{4} + \frac{1}{8} = \frac{2r+1}{8}$

$$TIME_{M_2}(n) \leq TIME_{M_1}(n)\left(\frac{16}{2r+1} + \frac{8}{(2r+1)r} + \frac{8}{r}\right) =$$

$$= TIME_{M_1}(n) \cdot \frac{16(2r+1)}{r(2r+1)} \leq c \cdot TIME_{M_1}(n)$$

Dacă $n \geq \max(3, n_d) \Rightarrow TIME_{M_2}(n) \leq c \cdot TIME_{M_1}(n)$

Dacă $n \leq \max(3, n_d) \Rightarrow TIME_{M_2}(n) = n+1.$

SUB6: CLASA DE COMPLEXITATE SPAȚIU

Modelul de Mașină Turing folosit – o bandă pentru input read-only și K benzi auxiliare (de lucru) infinite doar la un capăt. Mașina se oprește pe fiecare intrare

Def.

$SPACE_M(n) \rightarrow$ nr. maxim de celule folosite de mașina M pe o bandă auxiliară până la oprirea sa pe o intrare de dimensiune n.

$(D/N)SPACE_k(f(n)) = \{L \mid$ există o mașină Turing deterministă/nedeterministă cu K benzi și $L(M) = L$ și există n_0 a.î. $\forall n \geq n_0$ $SPACE_M(n) \leq f(n)\}$

Funcția $f(n)$ s.n. spațiu construibilă dacă $\exists$ o M.T. M și un n_0 a.î. $SPACE_M(n) = f(n)$ $\forall n > n_0$.

Funcția $f(n)$ s.n. spațiu-construibilă complet dacă $\exists$ o mașină Turing M cu $SPACE_M(n) = f(n)$ $\forall n$

TEOREME

$\rightarrow$ $((D/N)\ SPACE_k(f(n))) = (D/N)\ SPACE_k(c \cdot f(n))$ unde c este o constantă pozitivă nenulă:

$\rightarrow$ $(D/N)SPACE_k(f(n)) = (D/N)\ SPACE_1(f(n))$, $\forall K > 1$ și $\forall f$.

$\rightarrow$ $\forall f(n)$ recursivă, $\exists$ un limbaj recursiv L a.î. $L \notin DTIME(f(n))$
Se aplică și pt. DSPACE, NTIME, NSPACE

$\rightarrow$ Dacă $S_1(n)$ și $S_2(n)$ sunt spații complet construibile și $\frac{S_1(n)}{S_2(n)}$ tinde la 0 când n tinde la infinit, atunci

$DSPACE(S_2(n)) / DSPACE(S_1(n)) \neq \emptyset$

IERARHIE

$$DTIME(f(n)) \subseteq DSPACE(f(n))$$

$$DSPACE(f(n)) \subseteq DTIME(c^{f(n)}), \quad f(n) \geq \log(n)$$

$$NTIME(f(n)) \subseteq NSPACE(f(n))$$

$$NTIME(f(n)) \subseteq DTIME(c^{f(n)})$$

$$NSPACE(f(n)) \subseteq DSPACE(f(n)^2), \text{ pentru } f(n) \geq \log(n)$$

și f spațiu construibilă complet

$DSPACE(\log n) \subseteq P \subseteq NP \subseteq NSPACE = PSPACE$ și

$DSPACE(\log n) \subset PSPACE$

DEMONSTRAȚII

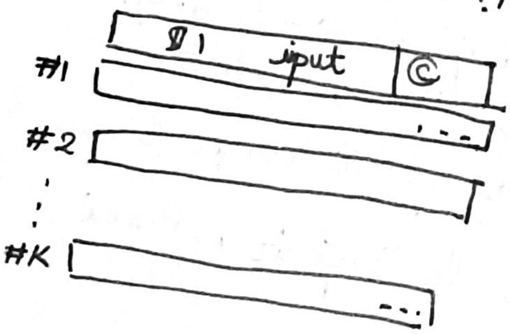
→ $(D/N)SPACE_k(f(n)) = (D/N)SPACE_k(c \cdot f(n))$, unde c constantă pozitivă nenulă.

→ Presupunem fără a restrânge generalitatea că $c > 1$. Dacă $c = 1$, este evident, iar dacă $c < 1$, atunci considerăm $1/c$.

→ Dem. prin dublă incluziune.

→ Este evident că $(D/N)SPACE_k(f(n))$ este inclus în $(D/N)SPACE_k(c \cdot f(n))$, pur și simplu considerăm o mașină Turing care ocupă $c * f(n)$ celule pe bandă.

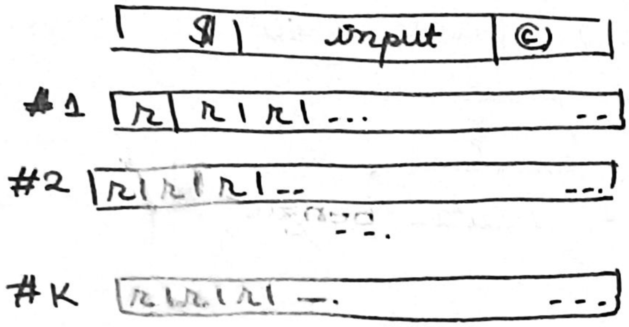
→ Considerăm o M.T.



Alegem un nr. natural r, cu prop. că $r > 2c$.

Mai întâi, împărțim benzile de la #1, la #K în segmente ~~date~~ de câte r celule.

Construim mașina M': fiecare celulă a benzilor de lucru corespunde câte unui grup de r simboluri de pe benzile de lucru ale mașinii M (celulele devin vectori de lg. r)..



Mașina M' va simula mișcările mașinii M.

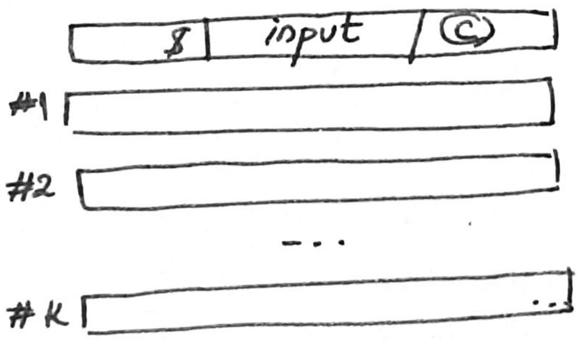
$SPACE_{M'}(n) \leq$ parte întreagă superioară $(SPACE_M(n)/r] \leq$

$\leq$ parte întreagă superioară $(c * f(n)/r) \leq f(n)$

$(D/N)SPACE_K(f(n)) \doteq (D/N)SPACE_1(f(n))$, pentru orice $K>1$ & $\forall f$.

→ Demonstrația pornește de la faptul că o mașină cu K benzi este echivalentă cu o mașină cu o singură bandă.

→ Avem mașina M, cu K benzi:



Și acum, construim mașina M' astfel:

→ M' are o sg. bandă auxiliară, iar elementele ei vor fi vectori cu 2K piste:

- pe pista $2*i-1$ se află conținutul benzii i a mașinii M
- pista $2*i$ conține 0-uri mai puțin pe o poziție, și are 1 unde se află capul de citire-scriere al benzii $\underline{\underline{i}}$ a mașinii M.

a_1	a_2	a_3	a_4	a_5	a_6

a_1	a_2	a_3	a_4	a_5	a_6
0	0	0	0	1	0

→ Mașine M' citește conținutul benzii auxiliare de la stânga la dreapta și memorează simbolurile ce ar trebui citite.

→ Apoi, când ajunge la finalul benzii auxiliare, simulează mișcările pe care le-ar fi făcut mașina M.

→ Parcurge din nou banda auxiliară, dar de la dreapta la stânga și actualizează conținutul benzii.

05.10.2018

C.C.: Curs 1

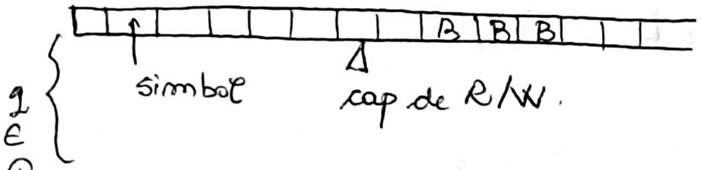
Calculabilitate: – Mașina Turing, – Programe Standard, – Funcții recursive.

Complexitatea: – calculului – timp, – spațiu

NP-completitudine

Constructivitate

Mașina Turing

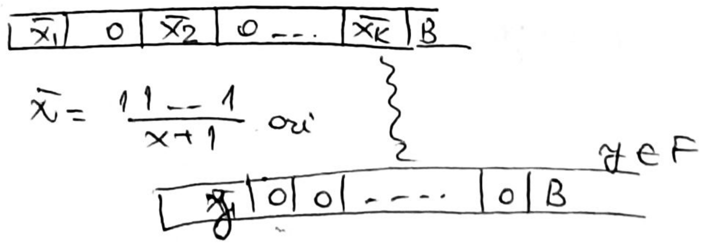


→ Calcul de funcții (I)

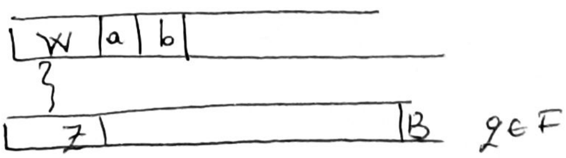
→ Dispozitiv de acceptare stringuri (II)

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

$f(x_1, \dots, x_k)$



(II)



(Def) M.T. este o str. $MT = (Q, \overset{\Sigma}{V}, \overset{\Gamma}{U}, \delta, q_0, F, B)$

Q = mult. finită de stări

q_0 = starea inițială

F = mult. de stări finale

$\Sigma = V$ = alf. de intrare

$\Gamma = U$ = alfabetul benzii; $V \subset U$

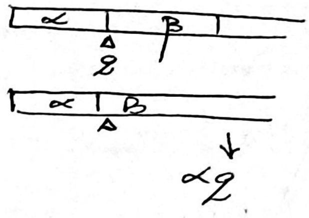
$B \in U \setminus V$ simbolul blank.

$\delta: (Q \setminus F) \times \Gamma \to 2^{Q \times U \times \{L, R\}}$

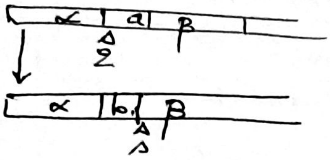
$(a, b, L) \in \delta(q, a)$

cu caract dacă $(a, B, \cancel{X}) \in \delta(q, a) \Rightarrow a = B$.

Configurație / Descriere instantanee (ID) : $\alpha q \beta$ ↔



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



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

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

$\vdash^* \quad \alpha X q \vdash \alpha s X \beta$ dacă $(s, b, L) \in \delta(q, B)$

$J_1 \vdash^* J_2$ — $J_1 = J_2$; $J_1 \vdash J_1' \vdash J_2' \vdash \dots J_k' = J_2$

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

$F(M) = \{ y \mid q_0 \bar{x}_1 0 \bar{x}_2 0 \dots \bar{x}_k \vdash^* q \bar{y} 0^t, t > 0, q \in F \}$.

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

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