

I

1. evitarea apelului dublu în lisp

```
(defun F(e)
  (cond
    ((null e) 0)
    (+ ((car e) (F cdr e))
       (F (car e)))))
```

2. evitarea apelului recursiv în paralel

$P(T, 0)$.

$P([H|T], S) :- P(T, S_1), \text{aux}(S_1, H, S)$.

$\text{aux}(S_1, H, S) :-$

$H < S_1, !,$

$S \text{ is } H + S_1.$

$\text{aux}(S_1, -, S) :-$

$S \text{ is } S_1 + 2.$

3. $\gamma \rightarrow$ lista de parametri

Funcția F are următorul efect:

- dacă x este unicul parametru, se returnează tot x
- dacă mai există alți parametri stocați în γ , atunci se returnează lista alcătuită din elem. lui x și primarele elem.

are ceilor altii parametri

$$(F \text{ ' } \underline{(1 \ 2)}) \rightarrow (1 \ 2)$$

$$(F \text{ ' } \underline{(3 \ 4)} \text{ ' } \underline{(5 \ 6)} \text{ ' } \underline{(7 \ 8)}) \rightarrow (3 \ 4 \ 5 \ 7)$$

=> Rezultatul evaluării formulei date este: (1 2 3 4 5 7)

4. Dacă s - ar fi obligat x, y și z într-un write separat în cadrul predicatului s , atunci s - ar fi obligat toate tripletele posibile de (x, y, z) . Însă, sub forma în care este scris write - ul în cerință, cu 3 parametri, nu unul singur, interogarea s . va da eroare

II Să se genereze lista submultimilor cu suma divizibilă cu 3. (trebuie ca fiecare submultime să aibă cel puțin N elemente).

$$\text{subaux}(e_1 \dots e_m, m, \text{sum}, e_g) =$$

$$1. (e_1) > e_g + 1 \geq m \text{ și } (\text{sum} + e_1) \% 3 = 0$$

$$2. \text{subaux}(e_2 \dots e_m, m, \text{sum}, e_g)$$

$$3. e_1 \oplus \text{subaux}(e_2 \dots e_m, m, \text{sum} + e_1, e_g + 1)$$

$$\text{sub}(e, m) = \text{subaux}(e, m, 0, 0)$$

$$\text{main}(e, m) = \bigcup \text{sub}(e, m)$$

% subaux (L - lista de elem, N - Nr. întreg, S - Nr. întreg, L - Nr. întreg,

% R - lista rezultat)

% S -> suma submultimii actuale

% L →lungimea submultimii actuale

% model de flux: (i, i, a) → nedeterminist

subaux([H|_], N, S, L, [H]):-

L₁ is L+1,

S₁ is S+1,

L₁ ≥ N,

S₁ mod 3 == 0.

subaux([_|T], N, S, L, R):-

subaux(T, N, S, L, R).

subaux([H|T], N, S, L, R):-

L₁ is L+1,

S₁ is S+H,

subaux(T, N, S₁, L₁, R₁),

R=[H|R₁].

% sub L → lista, N - Nr întreg, R - listă rezultată

% model de flux: (i, i, a) → nedeterminist

sub(L, N, R):- subaux(L, N, 0, 0, R).

% main (L - lista, N - Nr întreg, R - listă de liste)

% acest predicat va returna lista tuturor subm. cerute

% model de flux: (i, i, a) → determinist

main(L, N, R):-

findall(R₁, sub(L, N, R₁), R).

Pentru III \rightarrow varianta aproape identică în R1 - 2023