

I

1. Soluție pentru evitarea apelului dublu:

```
(defun fct (F L)
  (cond
    ((null L) nil)
    (+ (camada x)
       (cond
         (x (cons x (fct F (cdr L)))))
         (+ nil)
         )
       )
    )
  )
)
```

2. Evitarea dublului apel:

R(T, -1).

R(H1T3, S) :- R(T, S1), aux(H, S1, S).

aux(H, S1, H) :- S1 = H, !.

aux(-, S1, S1).

3. Funcția G returnează și există conținutul, de 2 ori, primul element al unei liste date.

(setg Q 'G) \rightarrow Q se va evalua la G

(setg P Q) \rightarrow P se va evalua la G

(funccapp P '(A B C)) \rightarrow apelul funcției G cu parametrul (A B C)

\Rightarrow Rezultatul este: $(A \ A)$ (furnseale și evaluarea param.

și apoi realizarea operei funcției \Rightarrow P se eval. la G, apoi se aplică funcția G cu param (A B C)

4. Predicatul R va da eroare, deoarece, deși -P consideră pe

S parametru de ieșire, încearcă să -P folosească pentru a calcula și valoare. \Rightarrow rezultatul este eroare

Totuși, dacă am fi vrut ca predicatul să fie unu correct, ar trebui să calculeze -P sumă eform. din lista, astfel căm Rii schimbat următoarea linie de cod, în: S := S_i - H.

II Lista aranjamenteelor de la elemente, având un punctus P dat.

inserarea ($e_1 \dots e_m, e$) =

1. $e \oplus e_1 \dots e_m$

2. $P_1 \oplus \text{inserarea } (e_2 \dots e_m, e), m \geq 1$

aranj ($e_1 \dots e_m, R, p$) =

1. (e_1), $R_2 = 1$ și $p = e_1$

2. aranj ($e_2 \dots e_m, R, p$)

3. inserarea (aranj ($e_2 \dots e_m, R - 1, \bar{e}_1$)), $R > 1$,

$$p \circ \circ e_1 = 0$$

maiin (e, R, p) = \cup aranj (e, R, p)

% inserarea ($L \rightarrow$ Listă, E \rightarrow Element, R \rightarrow Lista Rezultat)

% inserarea, pe înțind, un elem. între -ă lista, pe foarte puț. acestora

% model de flux: ($\cdot, +, Q$) \rightarrow mediteranist

insertare \tilde{a} ($L, E, \Sigma E | L_J$).

insertare \tilde{a} ($[H | T_J, E, R$): -

insertare \tilde{a} (T, E, R_1),

$R = [H | R_1]$.

% aranj (L - Lista, K - Nr. întreg, P - Nr. întreg, R - Lista rez.)
% model de REX: (L, i, j, Q) \rightarrow determinist

aranj ($[H | -J, I, H, [H | J]$).

% corpul în care se îndeplinește condițiile de lungime și produs

aranj ($\Sigma | T_J, K, P, R$): -

% se caută mai departe (aft element, afer soluție)

aranj (T, K, P, R).

aranj ($[H | T_J, K, P, R$): -

% se alege un element, dacă acesta îndeplinește

% condiție: mai este REX pt. ce împarte exact R prod. rămase

$K > \underline{1}$,

$P \bmod H = := 0$,

K_1 îs $K - \underline{1}$,

P_1 îs P div H ,

aranj (T, K_1, P_1, R_1),

insertare \tilde{a} (R_1, H).

% main (L - Lista, K - Nr. întreg, P - Nr. întreg, R - Lista de
% liste)

% predicatul returnează o listă cu toate aranjamentele rezultante

% model de REX: ($, i, j, Q$) \rightarrow determinist

$\text{main} (L, \kappa, P, R) :=$

$\text{binop} (R, \text{array} (L, \kappa, P, R), R).$

III Nodurile de pe nivelurile impare sunt impari
ce se văd. de la.

$$\text{imparire} (e, e, \text{min}) = \begin{cases} e, & \text{e este } \text{par} \text{ și } \text{min} \text{ e impar} \\ e, & \text{e este } \text{impar} \text{ și } \text{min} \text{ e par} \\ \bigcup_{i=1}^m \text{imparire} (e_i, e, \text{min} + 1), & \text{altele} \end{cases}$$

$$\text{main} (e, e) = \text{imparire} (e, e, -1)$$

; imparire ($e - \text{listă}$) Atom, $e - \text{Eexprm}$, $\text{min} - \text{Nr. întreg}$

; $\text{min} \rightarrow$ nivelul modulu lui curent

(definim imparire (e și min))

(cond

$$((\text{and} (\text{atom} e) (\text{oddp min})) \Rightarrow e)$$

$$((\text{atom} e) \wedge)$$

; dacă nivelul atomului e fost impar, sun impariz

; dacă nu, sun parfizat valoarea modulu.

; dacă atomul e lista, aplicăm Recurția pe toate

; exprm. acestora, încă la final le reunim cu list

$$(+ (\text{mapcar} \#' (\text{functie} x))$$

$$(\text{imparire } x \wedge (+ \text{min} 1))$$

; Se urmărește funcție functie param, etc.

; corect nu putem impara param.

; e și min direct în mapcar pt.

; "Recurția" imparire

; main (P - Lista, e - Elém)

(defun main (P e))

(measuring P e -1)

; se initializa cu niv. curent ca -1 pt. ca ultimul cînd

; se ajunge la sfîrșit. Pentru că este (mîine pe suprafață,

; mîine pe traseu), acesta să devină 0

)