

I

1. Funcția H:

- conține o funcție numită ce returnează primul element din listă
- folosește funcția MAP apelată pt. a apela funcția numită de mai sus pentru fiecare element dintr-o listă X, elementele fiind reunite între-și lista rezultat formata cu list
- crează și lista, cu apelul, alin X și rezultatul din mapare

\Rightarrow rezultatul este: $(1 \ 2 \ 3) \ (4 \ 5) \ \underbrace{2 \ 4}_{\text{}} \)$

\hookrightarrow pentru că adaugă
dintre-și alți elemente în
ținând cont că nu adaugă lista
cu totul

2.

w \rightarrow predicat medeterminist, ce va returna, pe rând, toate
nr. de la un nr. dat, la 10 inclusiv

h \rightarrow apelarea predicatului w și afișarea pe ecran rezultatul
pe care îl scrie pe consola \Rightarrow se afișează toate rezultatele
date de predicatul w, cu spațiu între ele, apoi se termină proces
afișare

\Rightarrow Rezultatul evaluării $h(6, 5)$ este:

6 7 8 9 10

falș.

3. Funcția F calculează paralelul perimetrul și dimensiunile există.

(setg Q F) $\rightarrow Q$ se evaluatează ca F

(setg P Q) $\rightarrow P$ se evaluatează ca F

(apply P '($(2 3 4)$)) \rightarrow realizează apelul funcției F cu
 $L = (2 3 4)$ (argumentele funcției).
Sumă transmisă împotriva existenței)

Rezultatul este : 6

4.

P este un predicat medeterminist ce va da eroare în momentul
în care întâia pe a doua parametru \rightarrow vrea să se aplice valoarea
unei N și să se obțină N_1 de comentând N , dar N este param.
de output și nu are valoare.

Totuși, atunci când N are valoare, predicatul va merge prima
în linie, menind condiție de aprire și altădată nr. de că
o părțea că $- \infty$. (pentru întreagerea pr.).

II Lista subiectelor cu n elem. din multimea $[A, B]$
($A < B$, A, B nr. întregi) a.t. suma elem. să fie un nr. par.

candidat (a, e) =

1. $a, a \leq e$

2. candidat ($a+1, e$), $a \leq e$

se aux (a, e, n, eg, sum, ccl) = $\begin{cases} \text{ccl, } eg = n \text{ și sum e par} \\ \text{se aux } (a, e, n, eg+1, \\ \text{sum} + e, e \oplus ccl), \text{ e ccl} \end{cases}$

$lg \leq n$, unde $e = candidat$
(a, e)

$sue(a, e, m) = subaux(a, e, m, L, e, (e))$, unde
 $e = candidat(a, e)$

$submain(a, e, m) = \bigcup sue(a, e, m)$

- % candidat (A - nr întreg, partea imparică a intervalului,
- % B - nr întreg, partea superioară a int., R - nr. selectat
- % dim intervalul $[A, B]$)

% model de flux: (i, i, α) - medeterminist

candidat (A, B, A): - $A = \angle B$.

candidat (A, B, R): -

$A = \angle B$,

A_1 is $A + 1$,

candidat (A_1, B, R).

% subaux (A - nr întreg, B - nr întreg, N - nr. întreg, Lg - nr.

% \rightarrow sum - nr întreg, coe - lista, R - lista)

% sum \rightarrow suma submultimii curente

% coe \rightarrow submultimea curentă

% model de flux: $(i, i, i, i, i, i, \alpha)$ - medeterminist

subaux (-, -, N, N, sum, coe, coe): - sum mod 2 = := 0.

subaux ($A, B, N, Lg, sum, [hit], R$): - $Lg \leq N$,

candidat (A, B, E), $E \leq 14$,

sum1 is sum + E,

$Lg1$ is $Lg + 1$,

subaux ($A, B, N, Lg1, sum1, [E1 HIT3], R$).

% succ(A - Nr. întreg, B - Nr. întreg, N - nr. întreg, R - Lista)

% model de flux: (i, i, i, Q) → medeterminist

succ(A, B, N, R):-

conditie(A, B, E),

succAux(A, B, N, I, E, [E | S], R).

% succMain(A - Nr. întreg, B - Nr. întreg, N - nr. întreg, R - Lista de liste)

% predicat ce determină lista cu toate rezultatele

% model de flux: (i, i, i, Q)

succMain(A, B, N, R):-

functie(R1, succMain(A, B, N, R1), R).

III Se se determină lista modulilor de pe nivelul R_2 .
Nivelul său dinicu este considerat 0.

Model matematic:

$$\text{găsește}(e, \min, R_2) = \begin{cases} (e), & e \in \text{atam} \text{ și } \min = R_2 \\ \emptyset, & e \notin \text{atam} \text{ și } \min \neq R_2 \\ \bigcup_{i=1}^n \text{găsește}(e_i, \min+1, R_2), & \text{altele} \end{cases}$$

$$\text{main}(e, R_2) = \text{găsește}(e, -1, R_2)$$

; găsește(e - Lista / Atam , min - Nr. întreg , R - Nr. întreg)

; min → reprezentarea nivelurii modulului curent

(de lung găsește(e min R))

(cond

; în cursul în care am găsit un mod pe nivelul R, vom
; returna lista de mod (pt că următorul vom utiliza
; mapcam) → dacă e mod, deci nu pe niv. R => niv.
; (and (atcm e) (equal niv R)) (exist e)

(atcm e) niv

; Ca Remetătoare map se va utiliza mapcam, rezultatul
; fiind că există element

(+ (mapcam #) (comoda x))

(găsește x (+ niv 1) R)
)

f

))
; S-a utilizat Remetătoare map, deci rezultatul său
; R poate apărea pt. "găsește" corect direct ca
; mapcam (aceasta având și parametrul niv și R)

)

; main (l - lista inițială, R - nivelul actual; nr. intrare)
Definim main (e R)

(găsește e -1 R)

; S-a realizat -1 pt. nivelul inițial pt. că astăzi cămă
; se întâlnește în ordine, nivelul următorul să fie 0

)