

I 1. Evitarea dublării apelurii recursive - LISP

(defun F (e₁ e₂)

((lambda (x)

(append x

(cond

((null e₁) (cdr e₂))

(+ (list x (car e₂)))

)

)

(F (car e₁) e₂)

)

)

d. Accesul la următoarele liste mai sus, dar în PROLOG

R (Σ, 0).

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

aux (S₁, H, S) :-

S₁ ⊂ H, !.

S is H.

aux (S₁, -, S₁).

3.

RECAP: MAPCON aplică funcția pe:

- totași există

- pe adreșe există

- pe cădrul unei liste!

Stocarea rezultatelor într-o listă, pe care o construim, este
NCONC.

Se aplicați Recursia LIST pe lista L, pe cărui se poate scrie, ...,
prin adunare la lista REZULTATUL din ultimul element \Rightarrow
rezultatul va fi o listă ce conține lista L, căreia îl adăugăm,
..., prin adunare la lista REZULTATUL din ultimul element al
listei.

$$(\text{MAPCON } \#)G ((1\ 2)) \rightarrow ((1\ 2)\ (2)\ (2))$$

$$\circ (G ((1\ 2))) \rightarrow ((1\ 2)\ (2))$$

$$\circ (G ((2))) \rightarrow ((2))$$

$$(\text{APPLY } \#) \text{APPEND} (\text{MAPCON } \#)G ((1\ 2))) \rightarrow (1\ 2\ 2\ 2)$$

4. Predicatul definit este medeterminist, având următorul
model matematic:

$$f(e_1 \dots e_m)$$

$$1. \emptyset, m=0$$

$$2. e_1 \oplus R(e_2 \dots e_m), m \geq 1$$

$$3. f(e_2 \dots e_m), e_1 \text{ este }, m \geq 1$$

Lista rezultat este inițializată cu lista vidă.

Apoi există 2 opțiuni:

- se adaugă elementul la listă

- nu se adaugă elem. La listă (având acest drept
locur dacă este nr. par)

\Rightarrow elementul actual este:

- impar \rightarrow se adaugă categoric
- par \rightarrow poate sau se adaugă sau nu

Atunci, rezultatul interacției $R([1, 2, 3], L)$ este:

$$L = [1, 2, 3];$$

$$L = [1, 3];$$

false.

II Rezolvare în 12 jpg

III Eliminarea tuturor atențierilor numerice diverse cu 3.

Model matematic:

$$\text{elimina}(e) = \begin{cases} \emptyset, & \text{daca } e \% 3 = 0 \\ (e), & \text{daca } e \% 3 \neq 0 \text{ sau } e \text{ este} \\ & \text{atenție} \\ & \text{numerical} \\ \bigcup_{i=1}^n \text{elimina}(e_i), & \text{altfel} \end{cases}$$

$$\text{main}(e) = y_1, \text{ unde } y = \text{elimina}(e) \\ (y = y_1 \dots y_m)$$

(definim $\text{elimina}(e)$)

(cond

$$((\text{and}(\text{number}(e), (\text{equal}(\text{mod}(e, 3), 0))) \text{ nil}) \\ ((\text{atm}(e)) (\text{list}(e)))$$

$$)) (+ (\text{list}(\text{mapcar}(\text{#}' \text{elimina}, e))))$$

(definim $\text{main}(e)$)

$$) (\text{car}(\text{elimina}(e)))$$