

Cusiac Andrei

10-01-2022

gr. 222

~~AD~~

Colocuiu
PLF

(A.) Pentru a evita apelul dublu
recursiv folosim o funcție anonimă
locală, apelată cu parametrul dublat

```
( DEFUN F(L)
  +CO(( LAMBDA (X)
    (COND
      ((ATOM L) -1)
      ((> X 0) (+ (CAR L) X
        (F (CDR L))))
      (T (F (CDR L)))
    )
  )
  (F (CAR L))
)
```


~~B.~~

Cusiac Andrei

~~C.~~

gr. 222

✓

(C) Folosim o variabilă auxiliară niv pentru a contoriza nivelul actual din listă. Dacă întâlnim un atom, pe un nivel par și nenumeric, îl ignorăm, altfel îl returnăm, altfel dacă întâlnim o listă apelăm recursiv funcția.

$$\text{elim}(l, \text{niv}) = \begin{cases} \emptyset, & \text{dacă } l = \text{atom nenumeric} \\ & \text{și } \text{niv} \% 2 \neq 0 \\ (l), & \text{dacă } l = \text{atom} \\ \bigcup_{i=1}^n \text{elim}(l_i, \text{niv}+1), & \text{altfel} \end{cases}$$

\uparrow lista dată \uparrow nivelul actual în listă

$$\text{wrapper}(l) = \text{elim}(l, 0)$$

```
(defun elim (l niv)
```

```
  (cond
```

```
    ((and (atom l) (not (numberp l))  
          (= (mod niv 2) 0)) nil)  
    ((atom l) (l))
```

```
    (t (mapcan #'(lambda (x)  
                    (elim x (+ 1 niv))) l))
```

```
  )
```

```
)
```



```
( defun wrapper (L)
  (elim L a)
)
```

Cus. ac Andrei
gr. 222
✓

(B.) candidat (a, b, b).

candidat (a, b, t) :-

b₁ is b - 1,

b₁ > a,

candidat (a, b₁, t).

~~sk~~ subm(a, b, ~~t~~, c) :-

candidat (a, b, t),

subm-aux(a, b, [t], c)

subm-aux(a, b, Col, Col) :-

~~pare(Col)~~

mod(pare(Col), 2) =: 0,

mod(inpare(Col), 2) =: 1.

subm-aux(a, b, ~~t~~ [H, t], Col) :-

candidat(a, b, e),

e < H,

subm-aux(a, b, [e, H/T], Col).

pare(L, 0).

pare([H/T], X₁) :-

~~sk~~ mod(H, 2) =: 0, X₁ is X + 1,

pare(T, X).

p. ③

Cusiac Andrei

gr. 221

~~1/2~~

$\text{pare}([-, T], x) :-$

$\text{pare}(\$T\$, x).$

$\text{inpare}([], 0).$

$\text{inpare}([H, T], X_1) :-$

$\text{mod}(H, 2) = 1,$

$X_1 \text{ is } X + 1,$

$\text{inpare}(T, X).$

$\text{inpare}([-, T], x) :-$

$\text{inpare}(T, x).$