**Documentatie L1**

Constantinescu Ana-Gabriela, 321

04/12/2020

**Problema 8**

**Subpunctul a)**

Sa se elimine elementul de pe pozitia a n-a a unei liste liniare.

*Modele matematice:*

* l1…ln este lista initiala din care se va sterge elemental
* poz reprezinta pozitia curenta din lista la care se afla functia
* m reprezinta pozitia de pe care se sterge elemental din lista
* cum apelul initial al functiei trebuie sa aiba mereu poz=1, voi face o alta functie elim care sa primeasca ca parametrii doar lista si pozitia de pe care se elimina elementul

elim\_elem(l1…ln, m, poz) =

l1…ln daca poz > m

elim\_elem(l2…ln, m, poz+1) daca m = poz

l1 ⊕ elim\_elem(l2…ln, m, poz+1) altfel

elim(l1…ln, m) =

elim\_elem(l1..ln, m, 1)

*Implementare:*

(defun elim\_elem(l m poz)

(cond

((> poz m) l)

((equal m poz)(elim\_elem (cdr l) m (+ poz 1)))

(t(cons (car l) (elim\_elem (cdr l) m (+ poz 1))))

)

)

(defun elim(l m)

(elim\_elem l m 1))

*Testare functie:*

O imagine care conține text

Descriere generată automat

**Subpunctul b)**

Definiti o functie care determina succesorul unui numar reprezentat cifra cu cifra intr-o lista. De ex: (1 9 3 5 9 9) --> (1 9 3 6 0 0)

*Modele matematice:*

* transf\_nr este o functie care returneaza numarul format din elementele ale unei liste liniare(toti atomii listei sunt considerati cifre)
* transf\_lista este o functie care primeste un numar si returneaza o lista formata din cifrele numarului respectiv (invers). De exemplu: 1234 ⇒ (4 3 2 1)
* inversare\_lista returneaza o lista liniara cu elementele in ordine inversa
* successor rezolva cerinta problemei, folosindu-se de cele 3 functii enuntate anterior

transf\_nr(l1…ln, nr) =

nr daca n=0

transf\_nr(l2…ln, nr\*10+l1)

transf\_lista(nr) =

[ ] daca nr=0

nr%10 ⊕ transf\_lista(nr/10)

inversare\_lista(l1…ln) =

[ ] daca n=0

inversare\_lista(l2…ln) ⊕ l1

succesor(l1…ln) =

inversare\_lista(transf\_lista(transf\_numar(l1…ln, 0) +1))

*Implementare:*

(defun transf\_nr(l nr)

(cond

((null l) nr)

(t(transf\_nr (cdr l) (+ (\* nr 10) (car l))))

)

)

(defun transf\_lista(nr)

(cond

((= nr 0) ())

(t(cons (mod nr 10) (transf\_lista (floor nr 10))))

)

)

(defun inversare\_lista(l)

(cond

((null l) ())

(t(append (inversare\_lista (cdr l)) (list (car l))))

)

)

(defun succesor(l)

(inversare\_lista (transf\_lista (+(transf\_nr l 0) 1)))

)

*Testare functii:*

O imagine care conține text

Descriere generată automat

**Subpunctul c)**

Sa se construiasca multimea atomilor unei liste. Exemplu: (1 (2 (1 3 (2 4) 3) 1) (1 4)) ==> (1 2 3 4)

*Modele matematice:*

* el\_in\_lista returneaza t daca un element este intr-o lista si nil altfel
* mult\_atomi returneaza multimea tuturor atomilor dintr-o lista

el\_in\_lista(l1…ln, el) =

false daca n=0

true daca l1=el

el\_in\_lista(l2…ln, el) altfel

mult\_atomi(l1…ln, c)=

c daca n=0

mult\_atomi(l2…ln, l1⊕c) daca l1 este atom si el\_in\_lista(c, l1)=false

mult\_atomi(l2…ln, c) daca l1 este atom

mult\_atomi(l2…ln, mult\_atomi(l1, c)) altfel

multime(l1…ln) =

mult\_atomi(l1…ln, [ ])

*Implementare:*

(defun el\_in\_lista(l el)

(cond

((null l) nil)

((equal el (car l)) t)

(t(el\_in\_lista(cdr l) el))

)

)

(defun mult\_atomi(l c)

(cond

((null l) c)

((and (atom (car l)) (not(el\_in\_lista c (car l)))) (mult\_atomi (cdr l) (cons (car l) c)))

((atom (car l)) (mult\_atomi (cdr l) c))

(t (mult\_atomi (cdr l) (mult\_atomi (car l) c)))

)

)

(defun multime(l)

(mult\_atomi l ())

)

*Testare functii:*

O imagine care conține text

Descriere generată automat

**Subpunctul d)**

Sa se scrie o functie care testeaza daca o lista liniara este o multime.

*Model matematic:*

* pentru rezolvarea cerintei folosesc functia el\_in\_lista definite la subpunctul c

e\_multime(l1…ln) =

true daca n=0

false daca el\_in lista(l2…ln, l1)

e\_multime(l2…ln) altfel

*Implementare:*

(defun e\_multime(l)

(cond

((null l) t)

((el\_in\_lista (cdr l) (car l)) nil)

(t(e\_multime (cdr l)))

)

)

*Testare functie:*

O imagine care conține text

Descriere generată automat