Seminar 6 – MAP Functions (MAPCAR)

(defun triple(n) (\* n 3))

(MAPCAR #’triple ‘(1 2 3 4 5)) => (list (triple 1) (triple 2) (triple 3) (triple 4) (triple 5) ) =>(3 6 9 12 15)

(MAPCAR #’triple ‘(1 A 2 B 3 C) ? Error

(defun triple(n)

(cond

((numberp n) (\* 3 n))

(t n)

)

)

(MAPCAR #’triple ‘(1 A 2 B 3 C)) => (3 A 6 B 9 C)

(MAPCAR #’triple ‘(1 A (2 B) 3 C)) ? (3 A (2 B) 9 C)

(defun triple(n)

(cond

((numberp n) (\* 3 n))

((atom n) n)

(t (mapcar #’triple n)

)

)

(MAPCAR #’triple ‘(1 A (2 B) 3 C)) => (3 A (6 B) 9 C)

(triple ‘(1 A (2 B) 3 C)) => (3 A (6 B) 9 C)

Triple(n) =

* 3\*n, if n is a number
* n, if n is an atom
* U i=1 n Triple(n\_i), otherwise (n is a list)

(MAPCAR #’CONS ‘(A B C) ‘(1 2 3)) => (list (cons ‘A 1) (cons ‘B 2) (cons ‘C 3)) => ((A . 1) (B . 2) (C . 3))

(MAPCAR #’CONS ‘(A B C) ‘(1 2)) => ((A . 1) (B . 2))

Assume we have a numerical linear list.

(triple ‘(1 2 3 4)) => (3 6 9 12)

Assume that I want the sum of the elements from the result => 30

(apply #’+ ‘(3 6 9 12)) <=> (+ 3 6 9 12)

(+ ‘(3 6 9 12)) => Error

(apply #’+ (triple ‘(1 2 3 4))) => 30

1. Compute the product of the numerical atoms from a list. Ex: (prod ‘(1 a (2 c 3) 6 (a (b 5) 2) 5)) => 1800

Product(n)={n,n is a number

1,n is a non-numerical atom

Π i=1, n Product(n\_I),otherwise,if n is a list

I

(defun prod (n)

(cond

((numberp n) n)

((atom n) 1)

(t (apply #’\* (mapcar #’prod n)))

)

)

(prod ‘(1 a (2 c 3) 6 (a (b 5) 2) 5))

(prod 1) => 1

(prod ‘a) => 1

(prod ‘(2 c 3)) => 6

(prod 2) => 2

(prod ‘c) => 1

(prod 3) => 3 => (2 1 3) => apply \* => 6

(prod 6) => 6

(prod ‘(a (b 5) 2))

(prod ‘a) => 1

(prod ‘(b 5)) => 5

(prod ‘b) => 1

(prod 5) => 5 => (1 5) => apply \* => 5

(prod 2) => 2 => (1 5 2) => apply \* => 10

(prod 5) => 5 =>( 1 1 6 6 10 5) => apply \* => 1800

1. Compute the number of nodes from the even levels of an n-ary tree, represented as (root (subtree\_1) (subtree\_2) …(subtree\_n)). The level of the root is 1. Ex:

(A

(B

(C)

(D

(E)

)

(F)

)

(G

(H

(I

(J)

(K)

)

)

)

(L

(M)

(N

(O

(P)

)

(Q)

)

)

) => 7

CountEvenLvl(Node, Level)

(Level+1)%2, if Node is atom

 (countEvenLvl(Node\_I, Level+1) with i from 1 to n,Otherwise

(Level+1)%2, if Node is atom

=>0, Node is an atom and level is odd

1, Node is an atom and level is even

2

(defun count\_even(l level)

(cond

((atom l) (mod (1+ level) 2))

;((listp l) (apply #‘+ (mapcar #‘count\_even l (1+ level)) ))

((listp l) (apply #’+ (mapcar #’(lambda (a) (count\_even a (1+ level))) l)))

)

)

(+ 1 2)) => 3

(1+ level) <=> (+ level 1)

(mapcar #’count\_even ‘(A (B) (C)) ‘(2))

=> (count\_even ‘A 2)

(count\_even ‘(B)) ?????

(count\_even ‘(A (B)(C)) 3)

=> (lambda A) => (count\_even A 4)

=>(lambda (B)) => (count\_even (B) 4)

(lambda (C)) => (count\_even (C) 4)

Count\_Even\_Main(l1...ln) = count\_even(l1...ln, 0)

1. You are given a nonlinear list. Compute the number of sublists (including the initial list) where the first numerical atom (on any level) is the number 5. Ex: (A 5 (B C D) 2 1 (G (5 H) 7 D) 11 14) the lists that should be counted are: the initial list, (5 H), (G (5 H) 7 D) => 3.

We will have two functions:

* + One to check if the first numerical atom is 5 in a list. => not going to use MAPCAR
  + One to count how many sublists respect the condition. => MAPCAR

We will assume that we have implemented the function verify:

Verify(l1...ln) => 1, if the first numerical atom on any level is 5

-> 0, otherwise

CountLists(L) =

0, L is an atom

Verify(L) + CountLists(L\_I), otherwise

(defun CountLists(L)

(cond

((atom L) 0)

(t (+ (verify L) (apply #’+ (mapcar #’CountLists L))))

)

)

How to implement verify?

Verify(l1...ln)

* + List is empty => false
  + L1 = 5 => true
  + L1 is a number => false
  + L1 is an atom => verify(l2...ln)
  + L1 is a list => verify(l1) OR verify(l2...ln)

((A B C) 5 D)

((A B 2 C) 5 D)

Solutions:

* + Transform the list in a linear one. (optionally, keep only the numbers)
  + Return three different values:
    - 1 if first number is 5
    - 2 if first number is not five
    - 0 if there are no numbers in the list
  + Have a function to check if a list contains a number