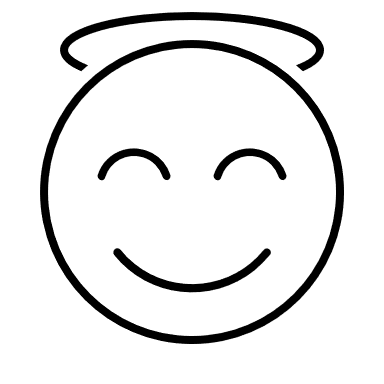
תרגיל בית 1

|  |  |  |
| --- | --- | --- |
| שם | ת.ז | מייל |
| עבידאת מוחמד | 319053823 | obedat@campus.technion.ac.il |
| מוחמד חטיב |  |  |



**שאלה 1)**

התבקשנו לממש פונקציה Cat שמקבלת שתי רשימות ומחזירה שרשורן

נשתמש בשתי פונקציות עזר :

(defun cat (l1 l2)(cat\_help(swap l1 ()) l2))

(defun car\_help (li1 li2)(cond

((Eq li1 nil) li2); if li1 =nil => li2( maybe li2 = nil)

((eq li2 nil) li1); if li2 = nil => li1 (li1 != nil)

((eq (cdr li1) nil) (cons(car li1) li2))

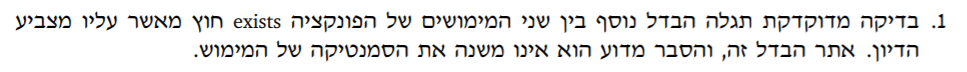
(t(cat\_help(cdr li1) (cons (car li1) li2)))))

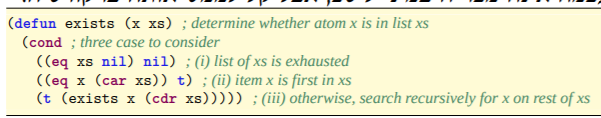
(defun swap (x y)(cond

((Eq x nil)nil)

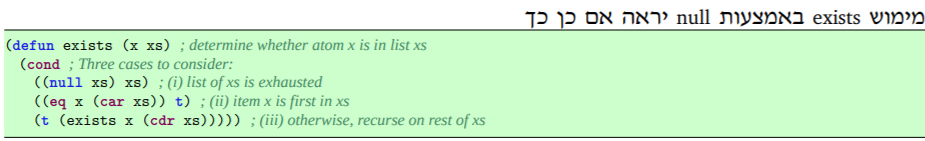
((eq (cdr x) nil) (cons (car x) y))

(t(swap (cdr x) (cons (car x) y)))))

**שאלה 2)**

מימוש ראשון :

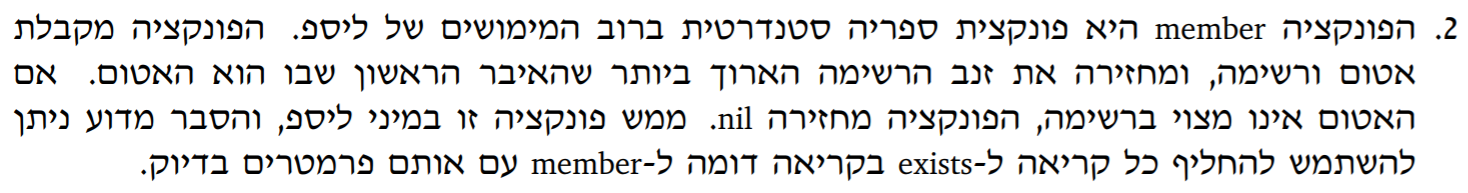
מימוש שני :



ההבדל בין שני המימושים הוא בבדיקה האם הרשימה XS ריקה , במימוש הראשון בודקים את זה ישירות אם הרשימה ריקה אז מחזירים nil אבל במימוש שני עושים זאת באמצעות פונקציית עזר קוראים לה null שבפועל כל מה שהיא עושה בודקת האם XS ריקה , אם היא ריקה כלומר כפי שהגרנו ()=NIL אז הפונקציה NULL תחזיר T .



ואז התנאי הראשון מתקיים ונחזיר כפי שכתוב XS , אבל נשיים לב ש XS=NIL כי בפועל קיבלנו T מהפעלת הפונקציה NULL לכן שני המימושים עושים אותה סמנטיקה .



(defun member (x xs)

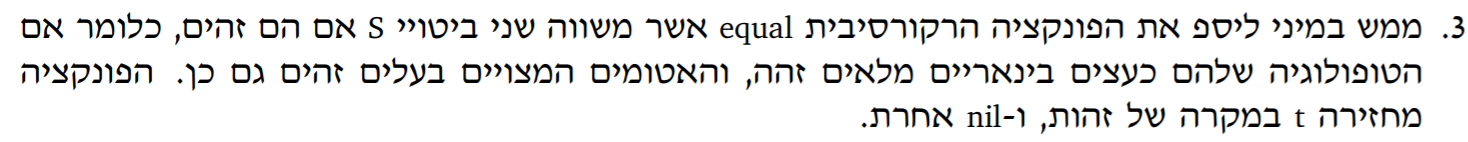
(cond

((null xs) xs) ; xs=nil => nil

((eq x (car xs)) xs) ; return xs\*\*\*\*\*\*\*\*\*

(t ( member x (cdr xs)))))

ניתן להשתמש בפונקציה MEMBER במקום EXIST מכיוון ש Member מחזירה את אורך תת הרשימה הארוך ביותר המתחיל מאטום נתון , כלומר אם לא חזר Nil מהפעלת הפונקציה נוכל להסיק שקיים האטום הזה כלומר , נוכל להשתמש בפונקציה Member במקום Exist כך שכל מה שעלינו לעשות לבדוק ערך ההחזרה . אם הוא NIL אז לא קיים האטום .

**שאלה 3)**

שני ביטוי S בנויים מאטומים , לכן עלינו לבדוק כל אטום בנפרד .

(defun equal (x y)

(cond

((null x) null b); if x=nil return null (b) (if b = nil return t else return nil)

((null b) b); if b=nil => nil (x != nil)

((eq (car a) (car b)) equal (cdr a) (cdr b))

((t) nil)

)

)