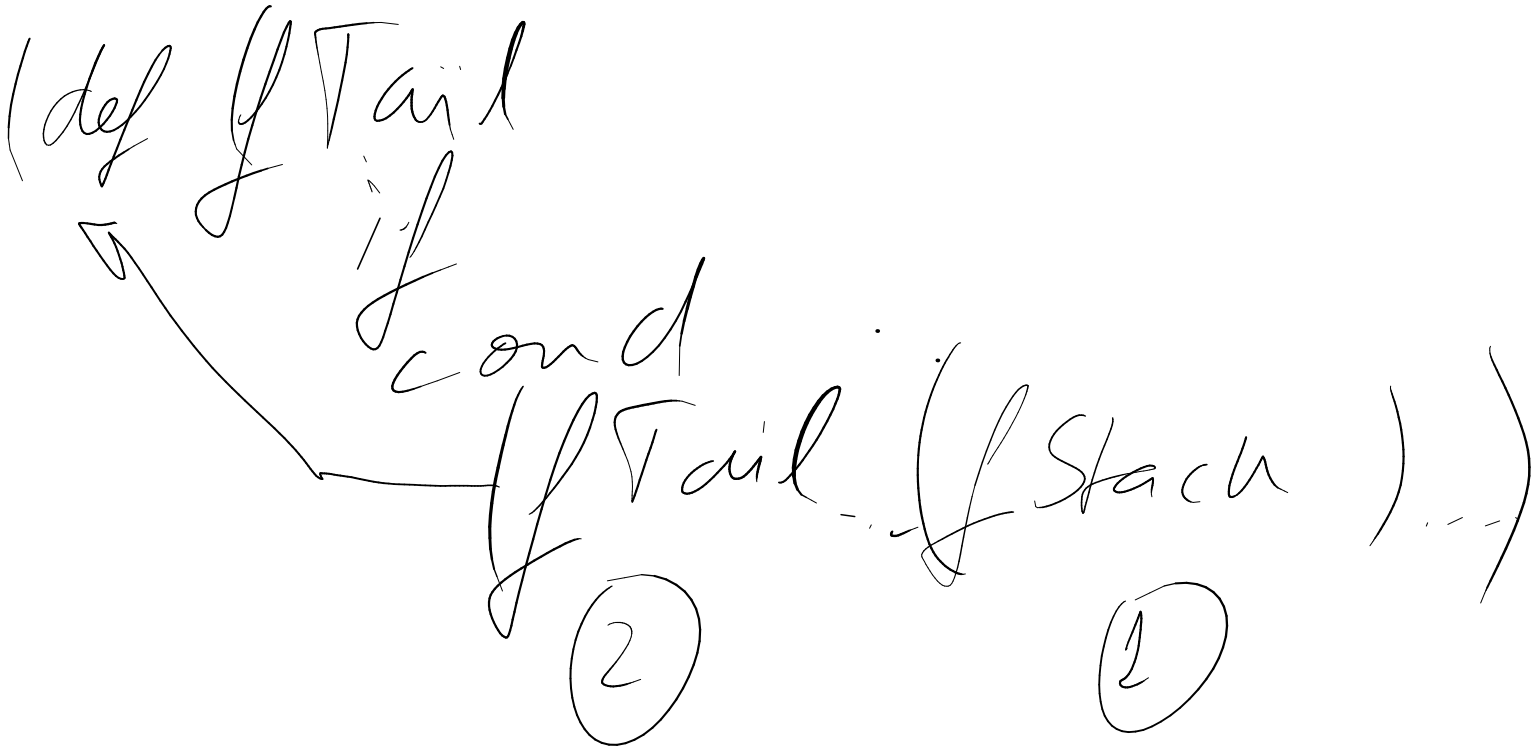
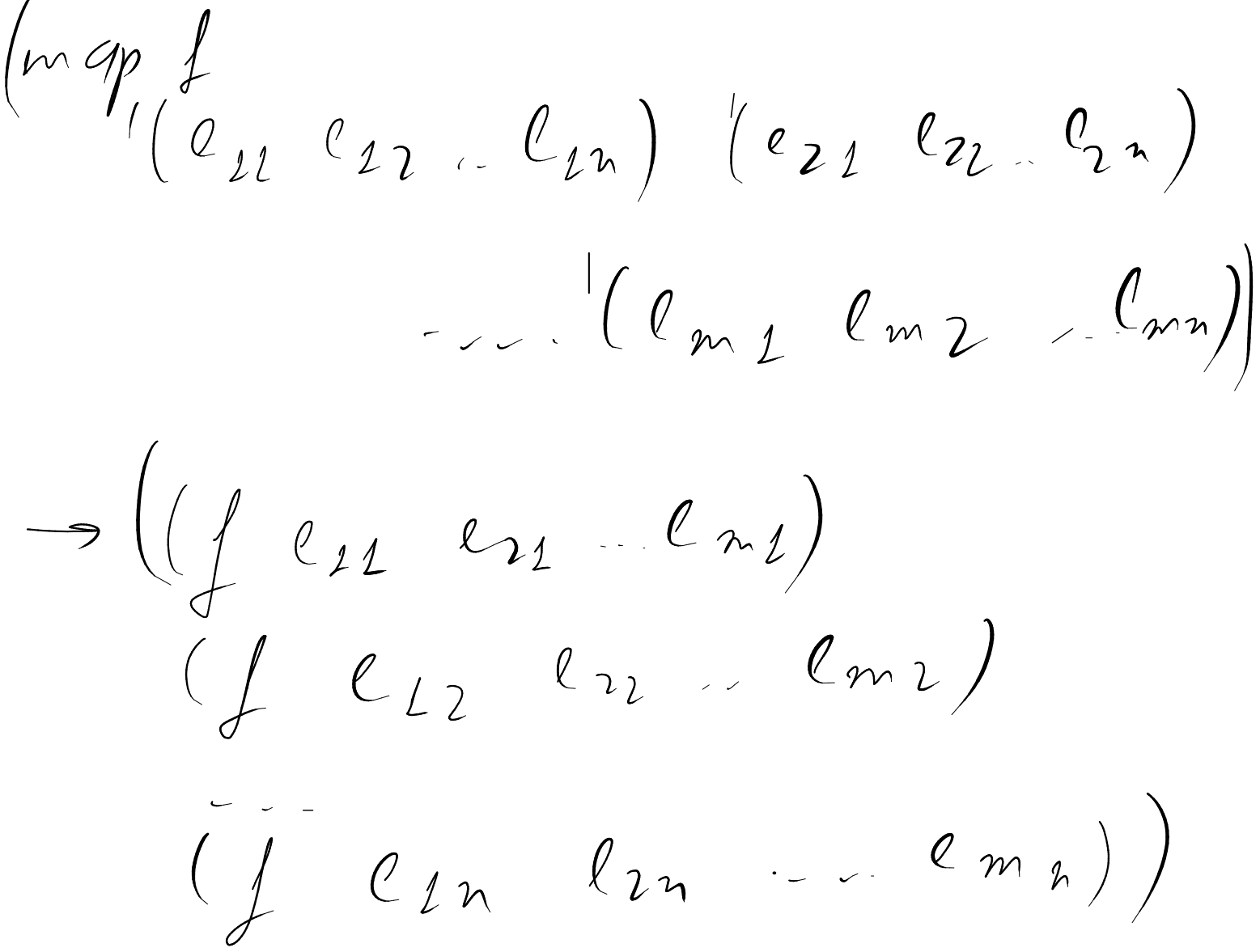


(def (f T x)
(fTail ...))

map primește o funcție f de 1 arg / de m argumente
primește o listă L / primește m liste de aceeași lungime n
întoarce o listă de lungime n=|L|
aplică f de n ori
pentru (list e1 e2 ... en) întoarce
(list (f e1) (f e2) ... (f en))



filter primește o funcție f de 1 arg (a cărei ieșire este considerată ca boolean)
primește o listă de lungime n
întoarce o listă de lungime <= n
elementele din ieșire sunt elemente din lista de intrare
aplică f de n ori

!! filter nu prelucrează elementele în niciun fel
evaluarea elementelor se face *independent*

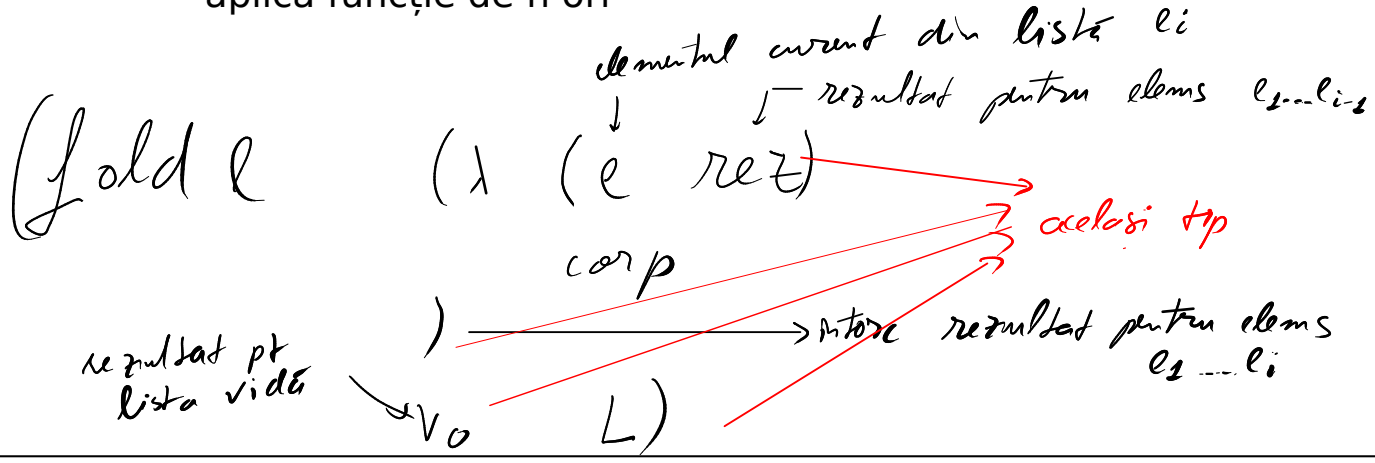
apply !! nu realizează o prelucrare iterativă
primește o funcție
primește o listă
aplică funcția *o singură dată*

L = (list e1 e2 e3 ... en)
(apply f L) -> (f e1 e2 e3 ... en)

- realizează o operație sintactică
pot folosi pentru a apela o funcție fără să știu pe câte argumente o voi apela

foldl/foldr primește o funcție de 2 argumente / m+1 argumente
primește o valoare inițială
primește o listă de lungime n / m liste de lungime n

aplică funcție de n ori



(foldl f v0 (list e1 ... en)) →
(f en ... (f e3 (f e2 (f e1 v0))) ...)

(foldr f v0 (list e1 .. en)) → (f e1 (f e2 ... (f en v0) ...)