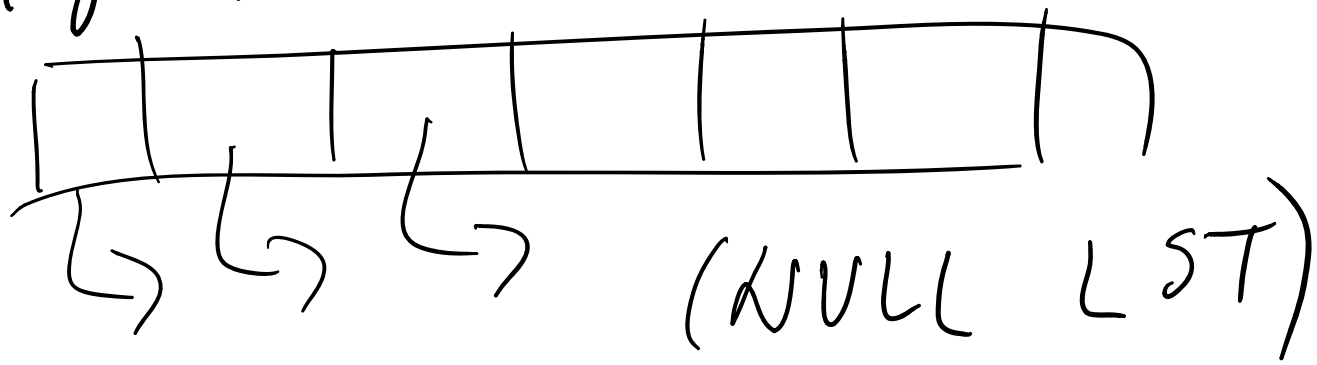
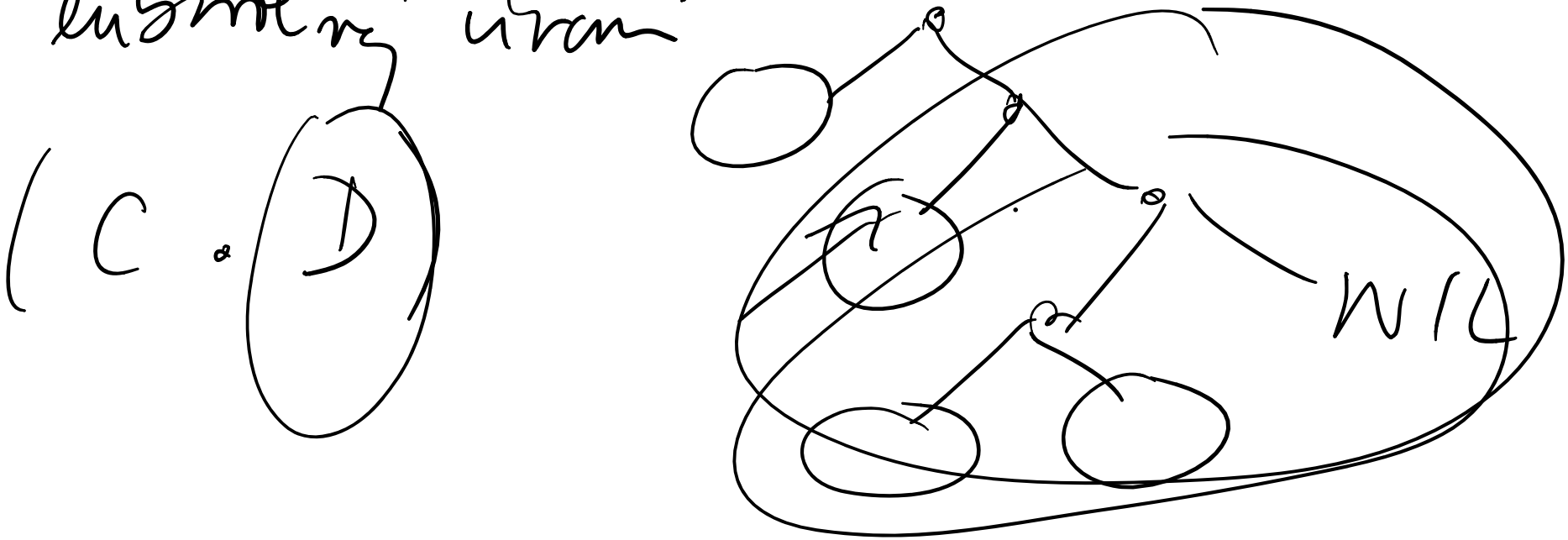
6.3.2018

na najnižší úrovni



na libovolné úrovni



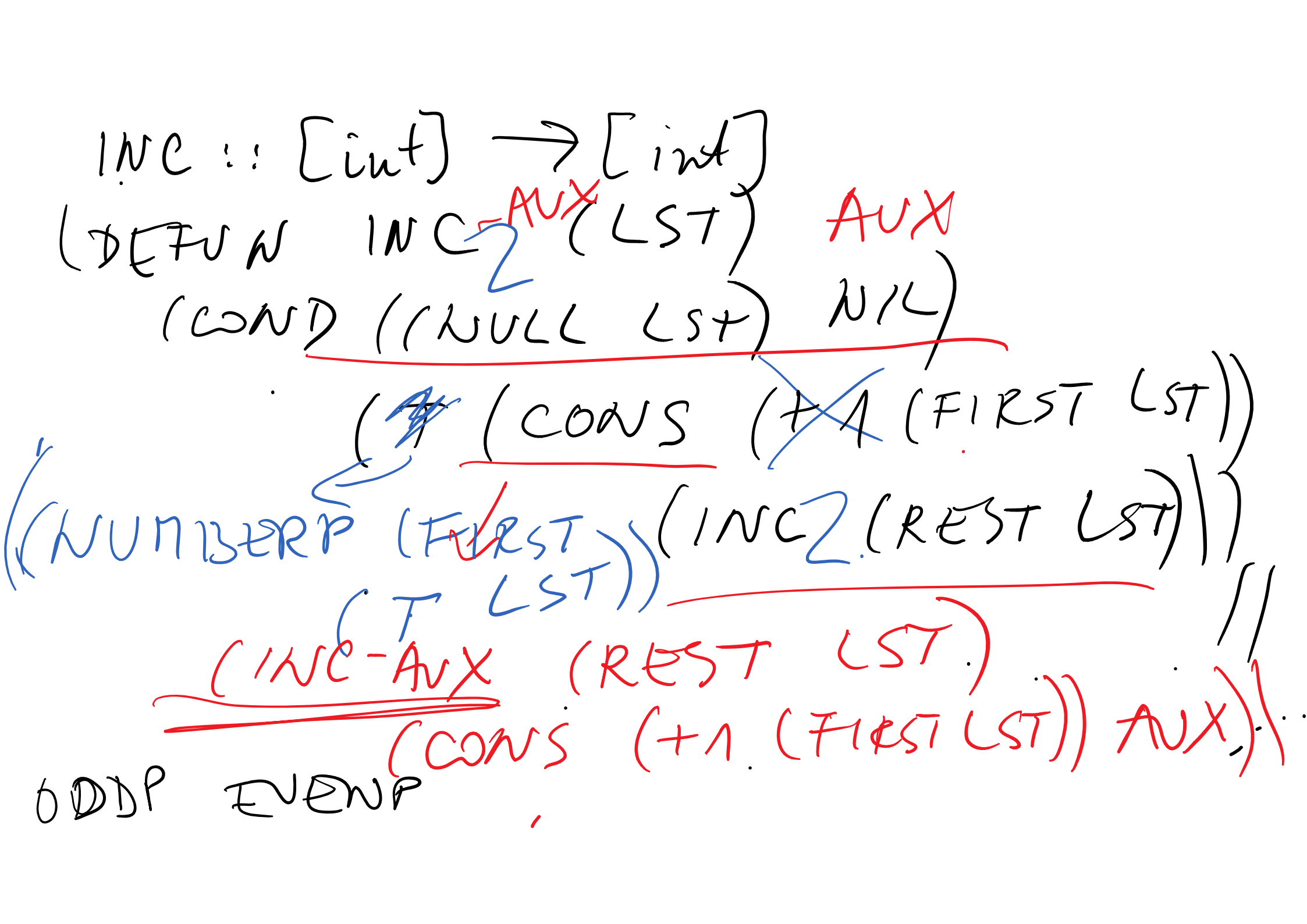
DEFINOVAT NEJAKÚ FUNKCIU

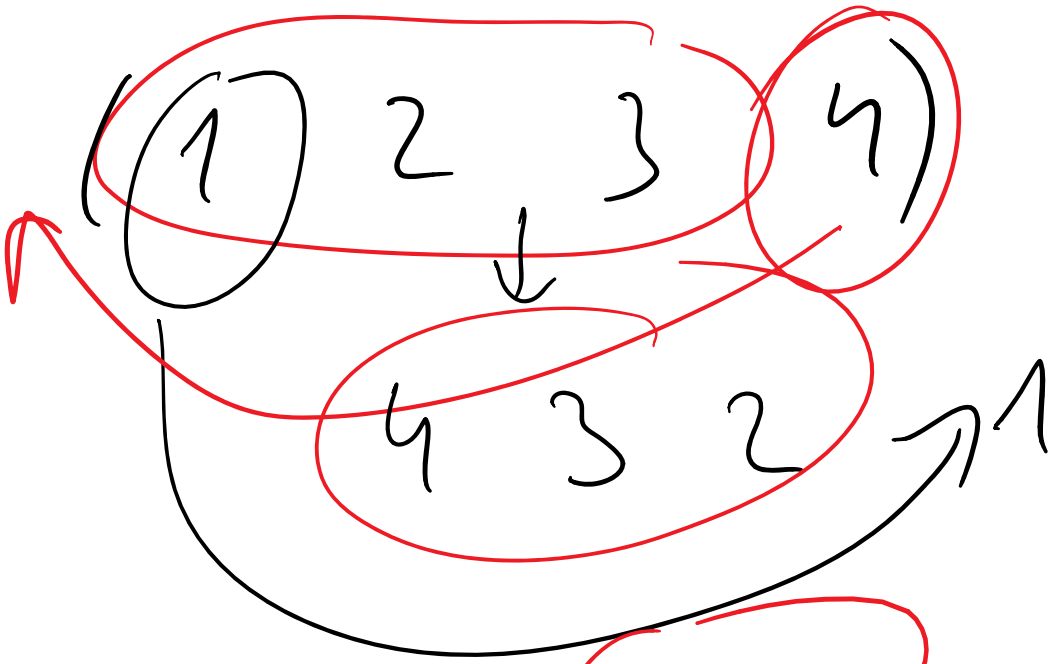
ZOBRAZENIE

Funkcia, ktorá každému prvku

množiny A priradí práve jeden prvok

množiny B.


INC :: [int] → [int]
(DEFUN INC-AUX (LST) AUX)
(COND ((NULL LST) NIL)
(T (CONS (+1 (FIRST LST))
(INC-AUX (REST LST) (CONS (+1 (FIRST LST)) AUX))))
((NUMBERP (FIRST LST)) (INC2 (REST LST)))
ODDP EVENP



(APPEND '(4 3 2) '(1))

(CONS (LAST LST) (REVERSE (BUT-LAST LST)))

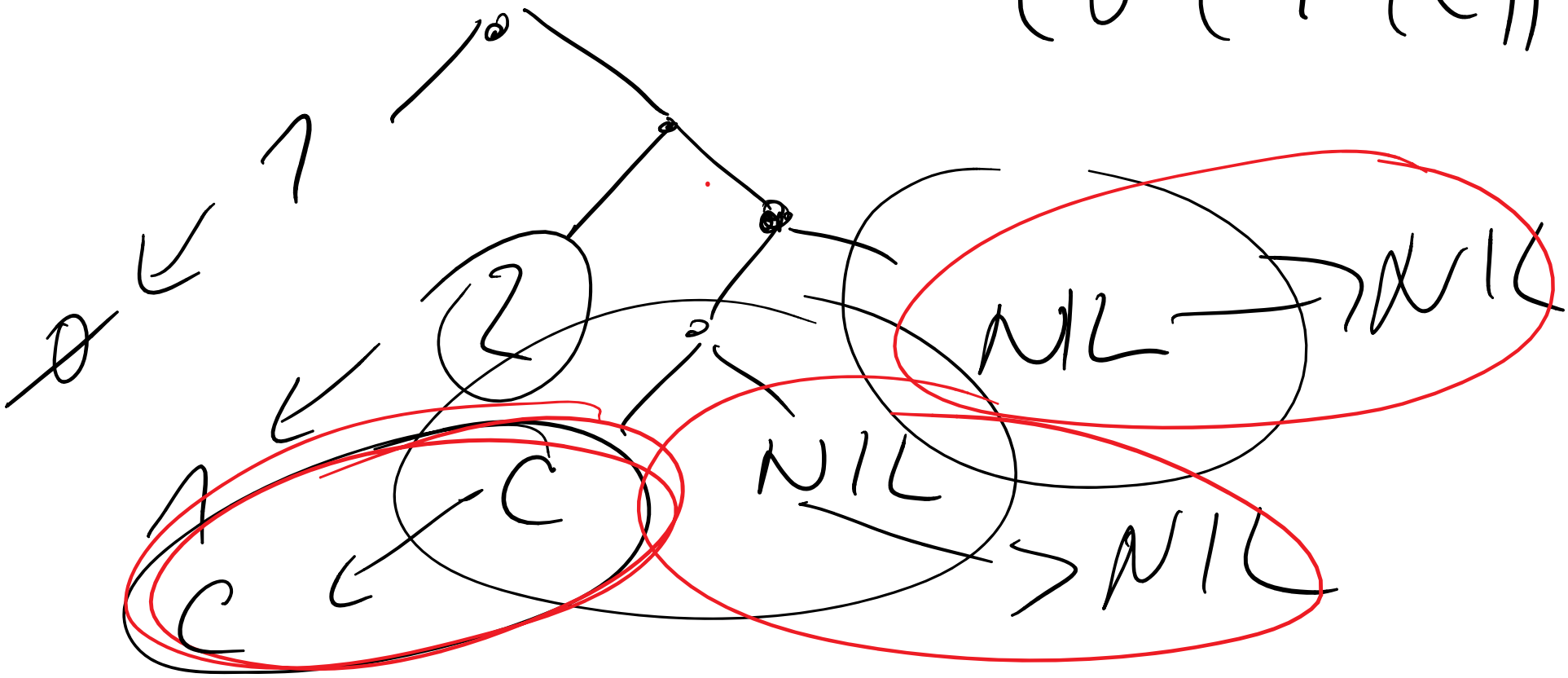
(APPEND (REVERSE (REST LST)) (LIST (FIRST LST)))

```
(DEFUN REV-AUX (LST AUX)
  (COND ((NULL LST) AUX)
        (T (REV-AUX
             (REST LST)
             (CONS (FIRST LST)
                   AUX)))))
```

```
(DEFUN <MAP-SOME) (LST)
  (COND ((NULL LST) NIL)
        ((<FUN> (FIRST LST))
         (CONS (TRANT) (FIRST LST))
               (MAP-SOME (REST LST))))
        (T (CONS (FIRST LST)
                 (MAP ... ))))
```

MAP su liste ricorsive

> (SUBST-D '(1 (2(C))) → (0 (1 (C))



FUNKCIONÁL FUNKCIA

→ ARG FN

– ktorá ako hodnotu vracia FN

(LAMBDA (arg...) forma)

nová F

MAPOVANIE PRVKOV S NEJAKOU VC (LAMBDA ...)

(MAP-MY LST FN)

> (MAP-MY '(1 2 3) #'INC)

(FUNCTION SYMBOL)

(QUOTE SE)

(DEFUN INC (N) (+ 1 N))

MAP-MY :: [int] x (int → int) → [int]

```
(DEFUN MAP-MY (LST FN)
  (COND ((NULL LST) NIL)
        (T (CONS (FN (FIRST LST))
                 (MAP-MY (REST LST) FN)))))
```

FUNCALL

(MAP-MY (FN arg1 ... argn))

(FUNCALL funcia arg1 ... argn)

(MAPCAR '(1 2 3) #'1+)

#'(LAMBDA (x) (+ 1 x))