

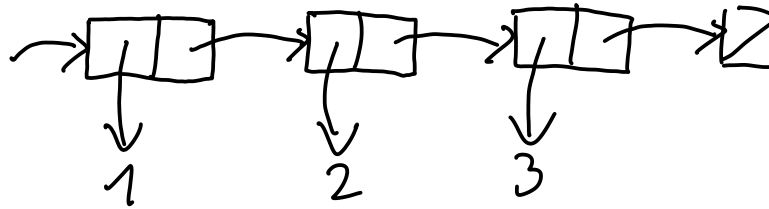
(cons A B)



null



(list 1 2 3)



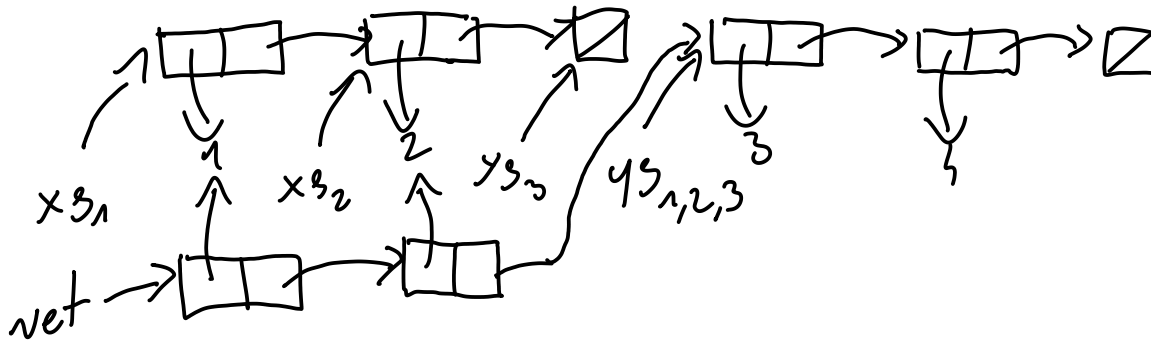
(define (append xs ys)

(if (null? xs)

ys

(cons (car xs) (append (cdr xs) ys))))

(append (list 1 2) (list 3 4))

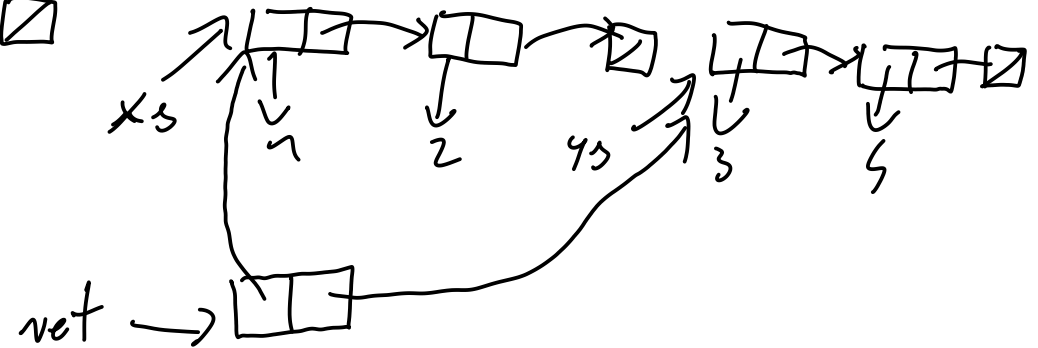


(define (foo xs ys)

(if (null? xs)

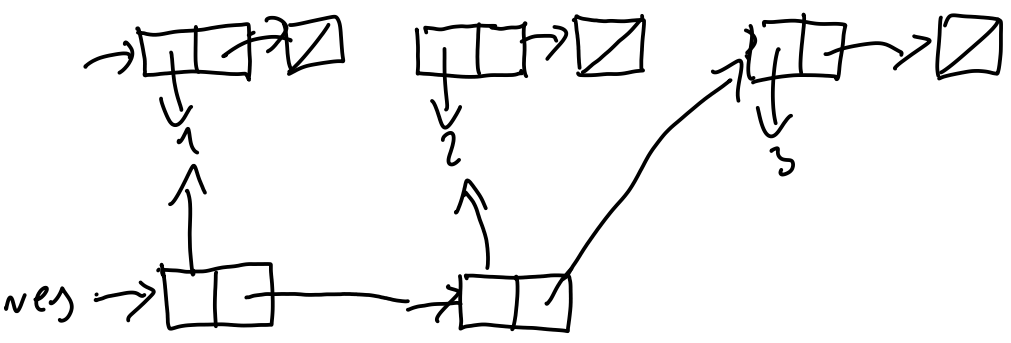
ys

(cons xs ys)))



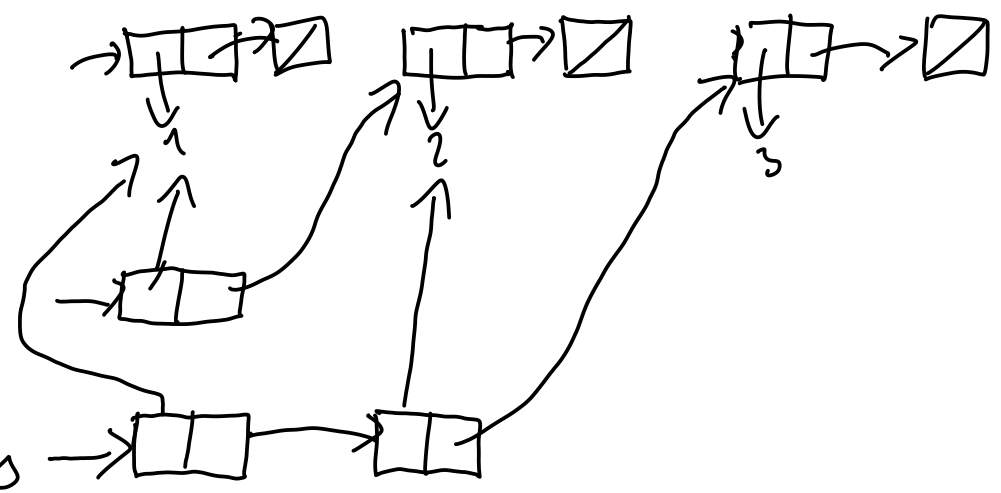
nes1

(append (list 1) (append (list 2) (list 3)))



nes2

(append (append (list 1) (list 2)) (list 3))



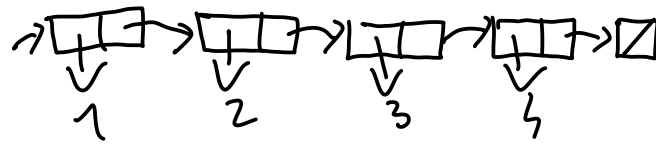
(equal? nes1 nes2) -> ... -> #t  
(eq? nes1 nes2) -> ... -> #f

nes ->

```
(define (append xs ys)
  (if (null? xs)
      ys
      (cons (car xs) (append (cdr xs) ys))))
```

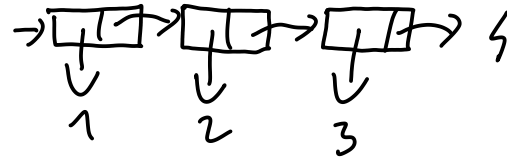
= - równość dla liczb  
equal? - równość strukturalna  
eq? - równość oparta na tożsamości

'(1 2 3 4)



TO JEST LISTA

'(1 2 3 . 4)



A TO NIE JEST LISTA