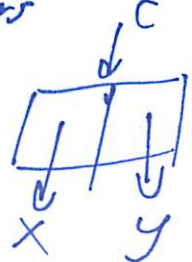


# Lists

Bare cases '() - empty

'a 'abc symbol, atoms

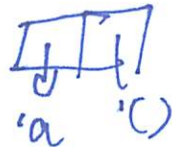
1 17 3.14 numbers

Constructor (cons x y)  $\Rightarrow$   cons cell

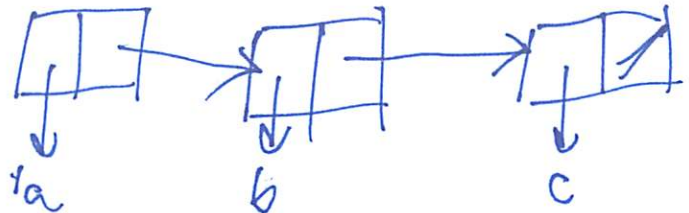
accessors (car c)  $\Rightarrow$  x

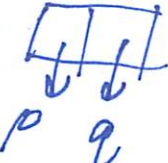
(cdr c)  $\Rightarrow$  y

List (cons 'a '())  $\Rightarrow$



(cons 'a (cons 'b (cons 'c '())))  $\Rightarrow$

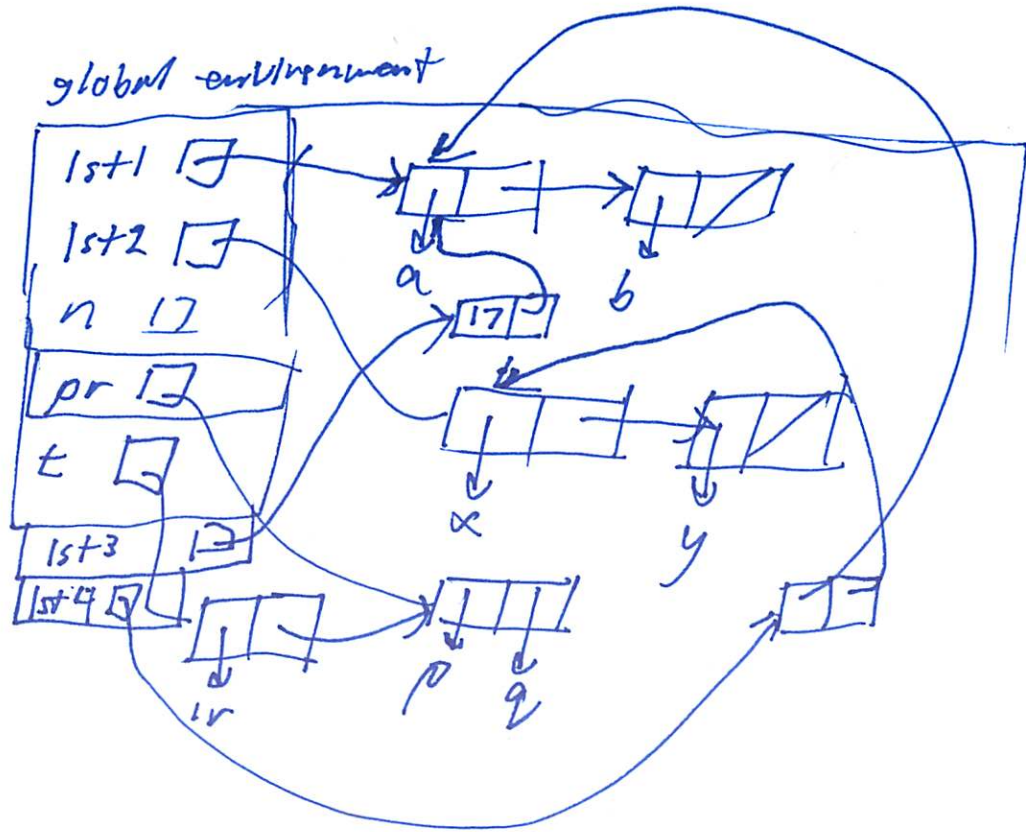


(cons 'p 'q)  $\Rightarrow$  

```

(define lst1 '(a b))
(define lst2 '(x y))
(define n 17)
(define pr '(cons 'p 'q))
(define t '(cons 'r pr))
lst1 => '(a b)
pr => '(p q)
t => '(r p q)
(define lst3 '(cons 17 lst1))
lst3 => '(17 a b)
(define lst4 (cons lst1 lst2))
lst4 => '((a b) x y)

```



(define lst1 '(a b))

(define lst2 '(c d))

(app lst1 lst2)

⇒ (cons (car lst1)  
 (app (cdr lst1) lst2))

⇒ (cons (car lst1)  
 (car (cdr lst1)) (app (cdr (cdr lst1)) lst2))

⇒

⇒ '(a b c d))

