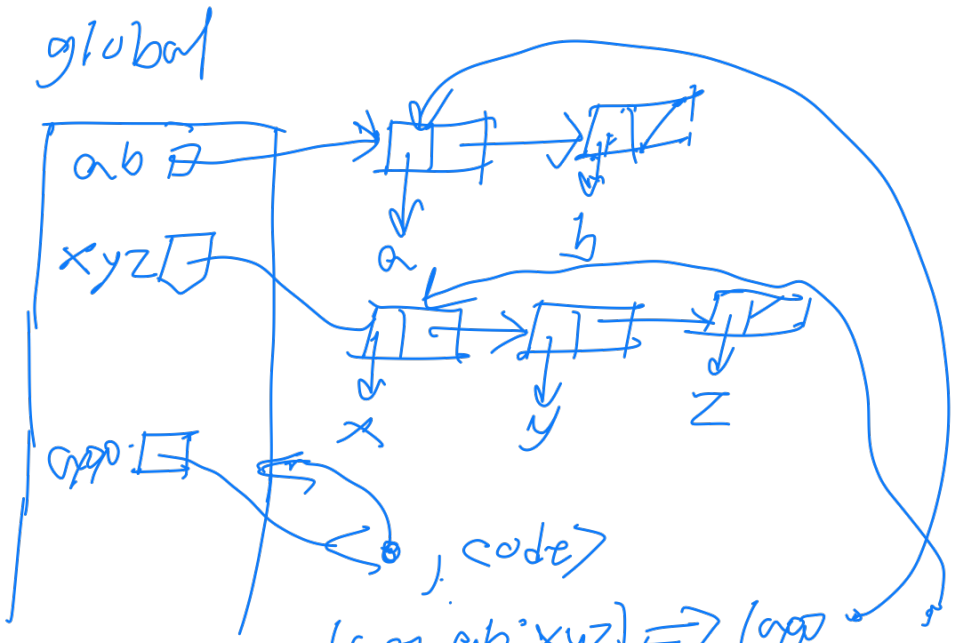


(oppo ab 'xyz) ⇒ (oppo ...)

⇒ (cons a (oppo (... '(b) xyz)))

⇒ (cons a (cons b (oppo '() xyz))))

⇒ (cons a (cons b xyz))

```
(define (rev3 lst)
  (if (null? lst)
      '()
      (append (rev3 (cdr lst)) (list (car lst)))))
```

(rev3 a b) ⇒ (append (rev3 '(b)) '(a))

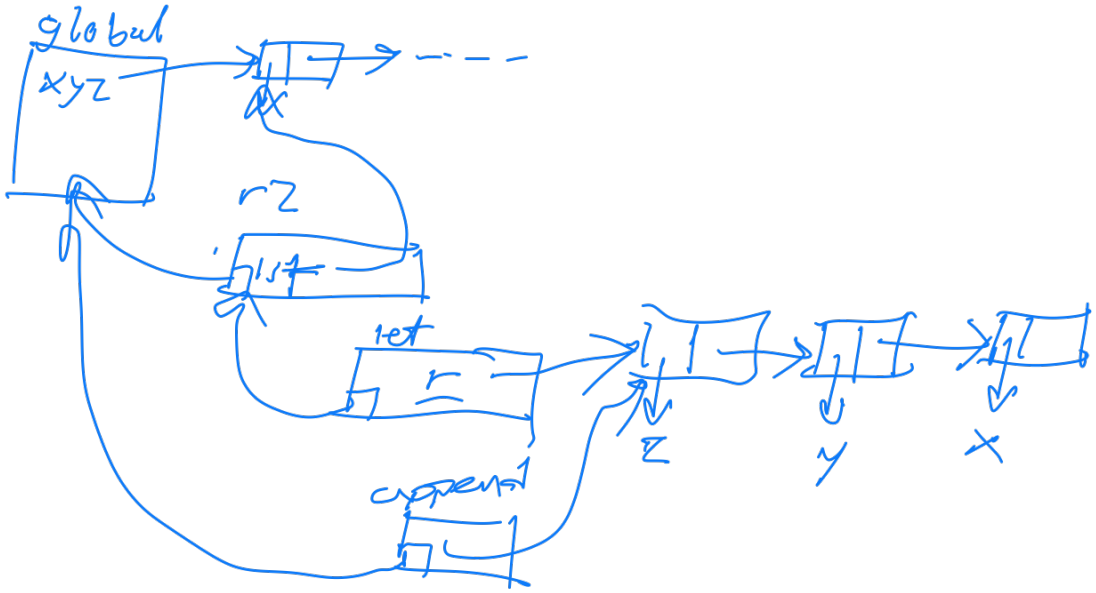
⇒ (append (append '() '(b)) '(a))

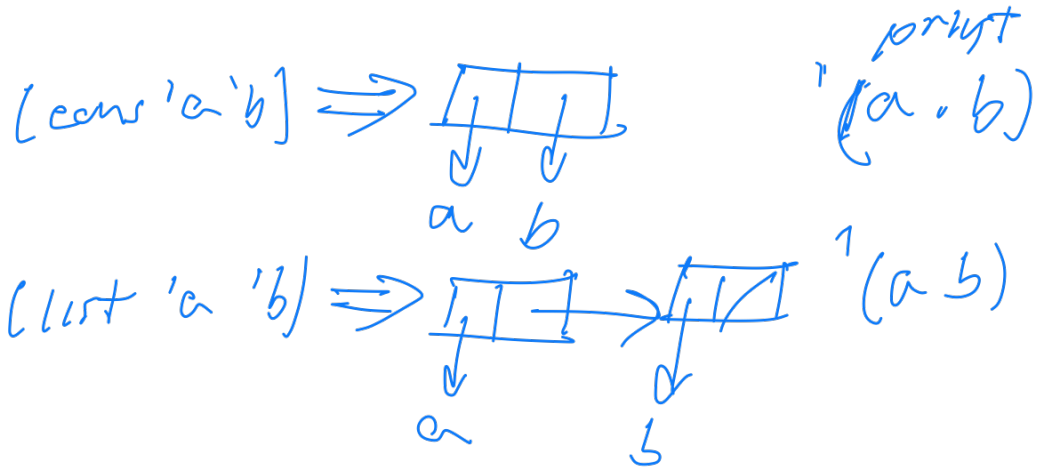
⇒ (append '(b) '(a))

⇒ '(b a)

(r2 lst)

(let [[r (rev3 lst)]]
  (append r r))




(cons 'a 'b) ⇒
'(a . b)
print
a
b
(list 'a 'b) ⇒
'(a b)
a
b