

```
(define (app x y)
  (if (null? x)
      y
      (cons (car x)
            (app (cdr x) y))))
```

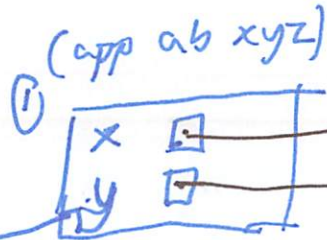
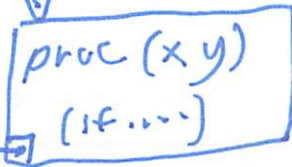
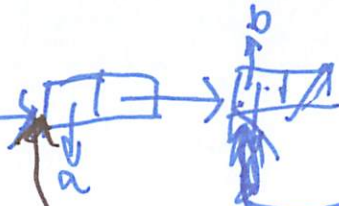
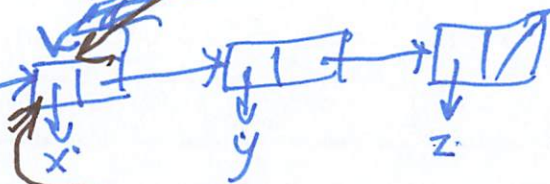
$$\begin{aligned}
 (\text{app } ab \ xyz) &\Rightarrow (\text{app } '(ab) \ '(x\ y\ z)) \\
 &\Rightarrow (\text{cons } a \ (\text{app } '(b) \ '(x\ y\ z))) \\
 &\Rightarrow (\text{cons } a \ (\text{cons } b \ (\text{app } '() \ '(x\ y\ z)))) \\
 &\Rightarrow (\text{cons } a \ (\text{cons } b \ '(x\ y\ z))) \\
 &\Rightarrow (\text{cons } a \ '(b\ x\ y\ z)) \\
 &\Rightarrow '(a\ b\ x\ y\ z)
 \end{aligned}$$

```
(define (app x y)
  (if (null? x)
      y
      (cons (cons (car x)
                    (app (cdr x) y)))))
```

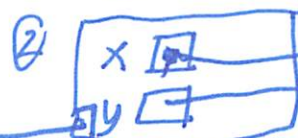
Global environment



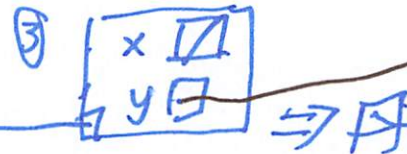
heap



~~(cons a xyz (cdr x))~~
(cons a (app □ y))



✓ (cons b (app '() y))



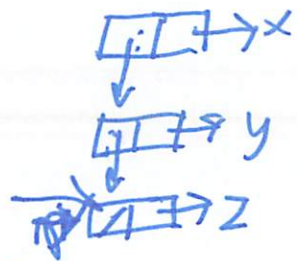
xyz \Rightarrow '(x y z)

(revl '(x y z))

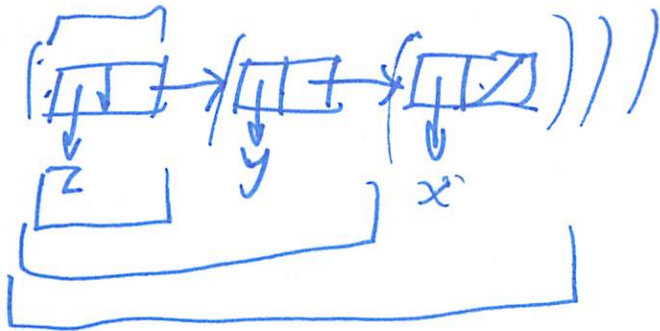
\Rightarrow (cons (revl '(y z)) x)

\Rightarrow (cons (cons (revl '(z)) y) x)

\Rightarrow (cons (cons (cons '() z) y) x)



((('() . z) . y) . z)



(define (revl x)
 (if (null? x)
 '()
 (cons (revl (cdr x))
 (car x))))

~~(cons~~
'(list (car x))
(list 'z))

