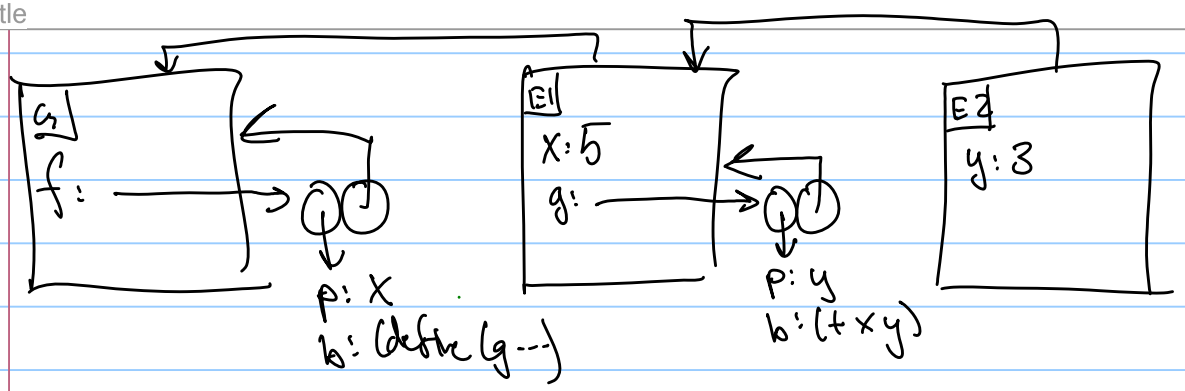
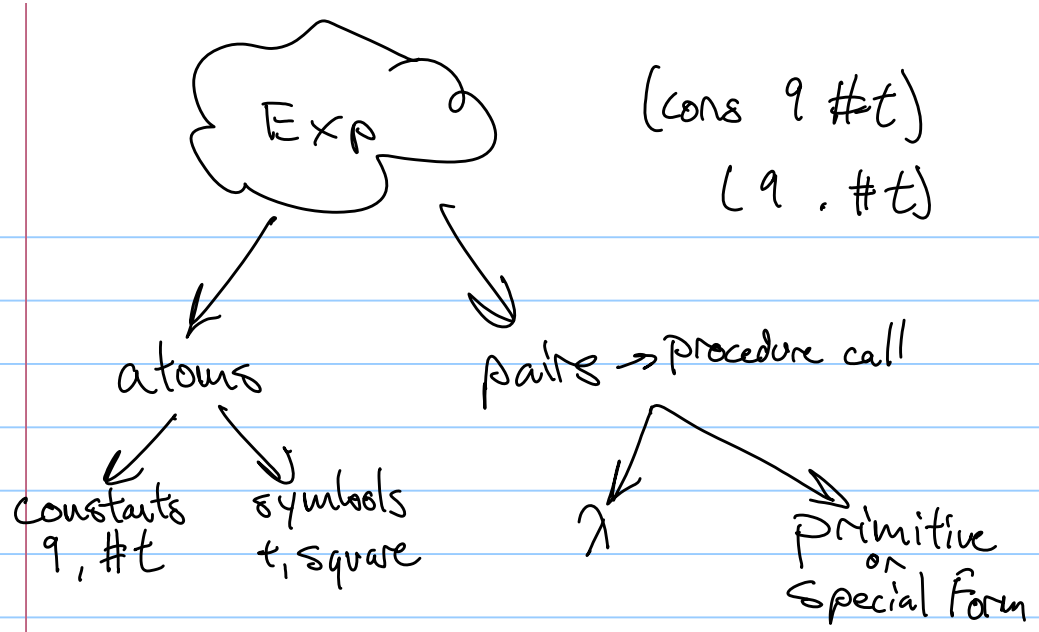
Environments & Local State 2

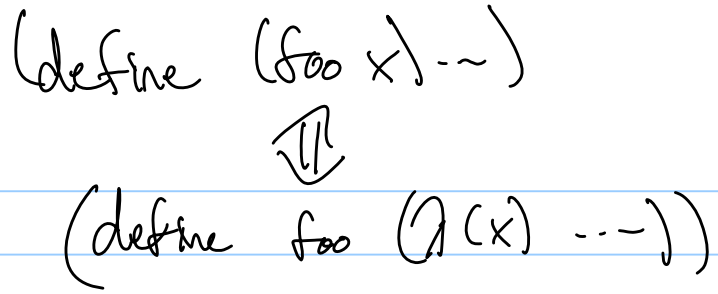
Note Title 7/15/2010




Exp
atoms
pairs → procedure call
constants
9, #t
symbols
t, square
λ
primitive
or
special form

(cons 9 #t)

(9 . #t)

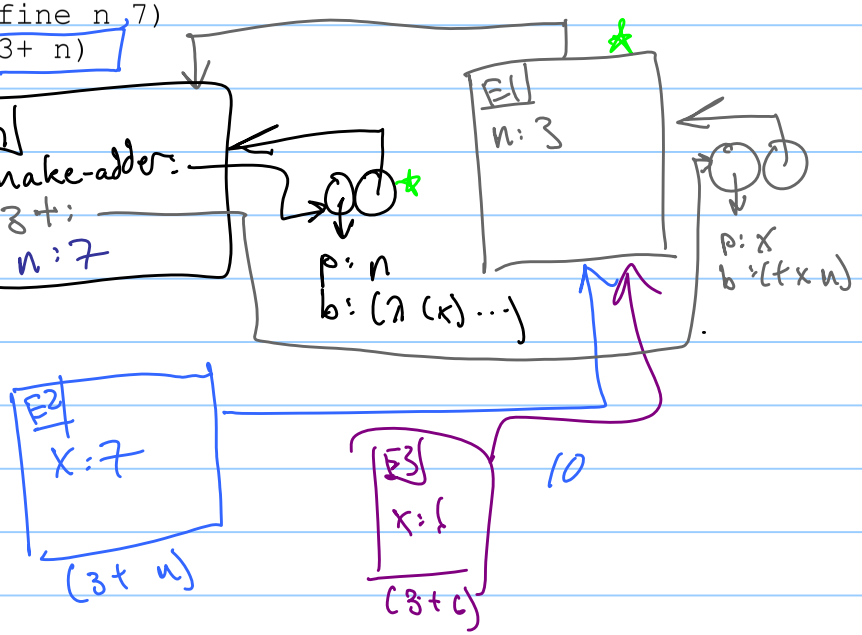



```
(define (make-adder n)
(lambda (x) (+ x n)))
(define 3+ (make-adder 3))
1
(define n 7)
> (3+ n)
```

```
(define (make-adder n)
  (lambda (x) (+ x n)))
(define 3+ (make-adder 3))
(define n 7)
> (3+ n)
```

(define make-adder (λ (n) (λ (x) ...)))

3+ 6



```
;;;;; In file cs61a/lectures/3.2/count4.scm
(define make-count
  (let ((glob 0))
    (lambda ()
      (let ((loc 0))
        (lambda ()
          (set! loc (+ loc 1))
          (set! glob (+ glob 1))
          (list loc glob))))))
```

((λ (glob) ...) 0)

((λ (loc) ...) 0)

(define mc (make-count))

instance

(mc)

(1 1)