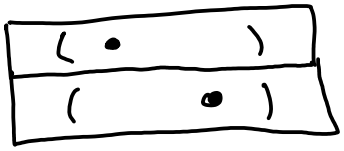
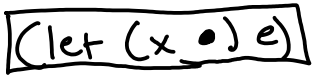
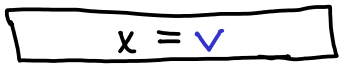


e / v



v

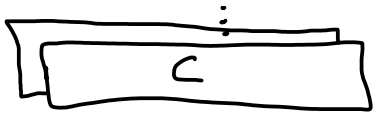
⇒



e

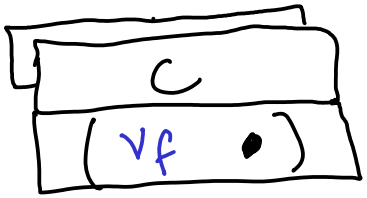
(lam (x) e)

⇒



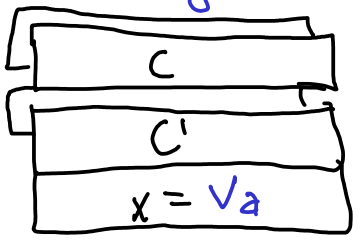
clos(c_B, x, e)

c_B = get-bindings(c)



v_a

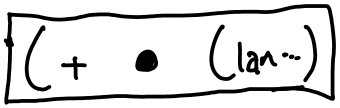
⇒



e

v_f = clos(c', x, e)

(+ 1 ((lam (x) x) 2))

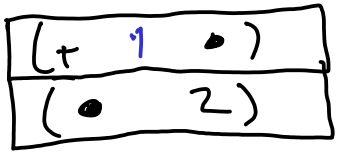


1

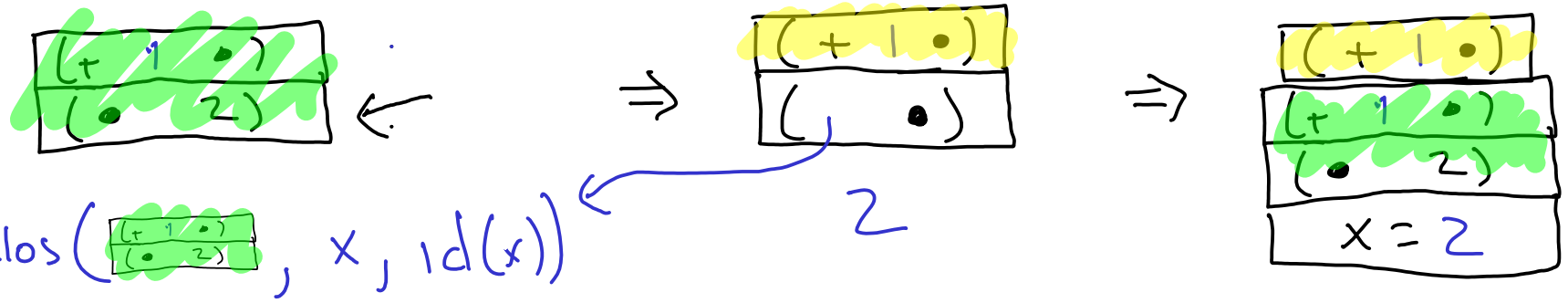
⇒

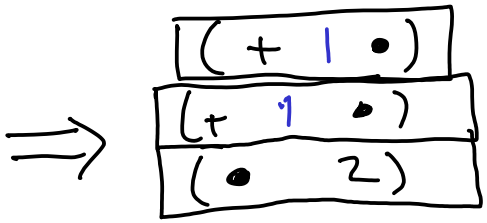
((lam (x) x) 2)

⇒



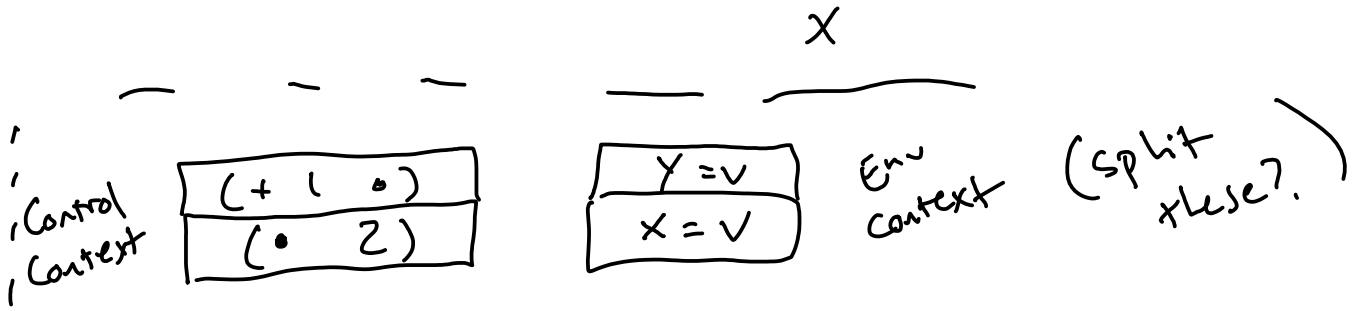
(lam (x) x)





2

err-not-a-fun



get-bindings :: List<Context> → List<Context>

all returned contexts are x = V shaped

```
(let (f (let (x 5)
         (lam (y) x)))
  (f 2))
```

(let (f •) (f 2)) ⇒ (let (f •) (f 2)) / (let (x •) (lam …))

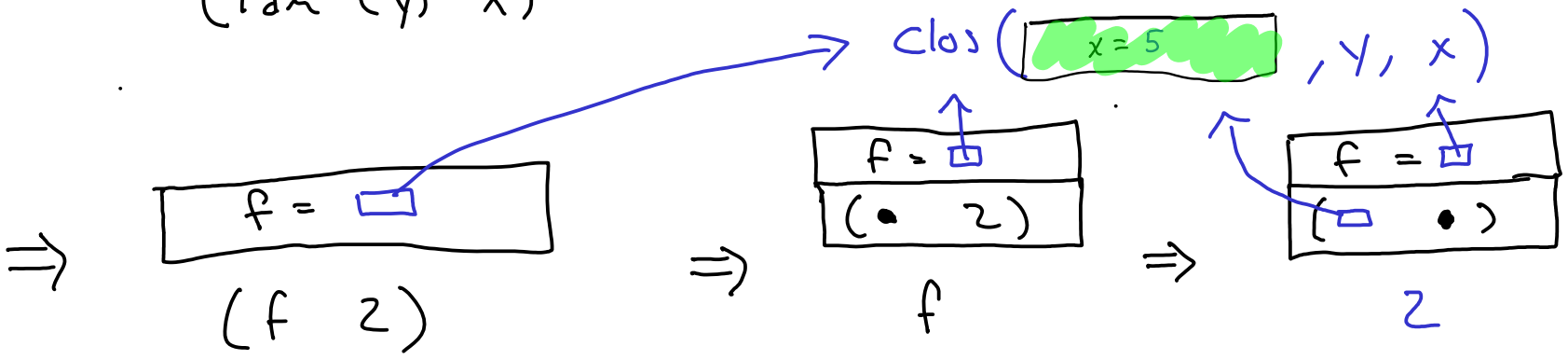
(let (x 5) (lam (y) x))

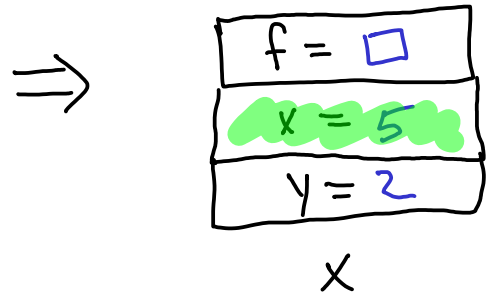
5

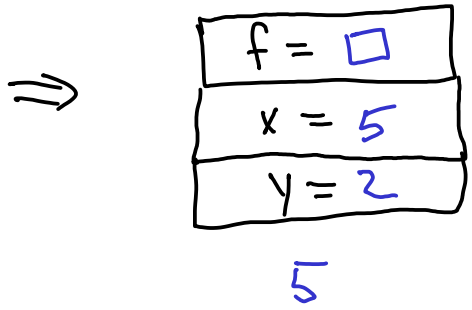
⇒ (let (f •) (f 2)) / x = 5

(lam (y) x)

⇒ (let (f •) (f 2))



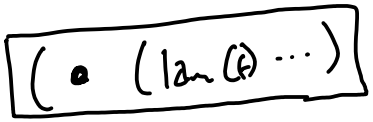




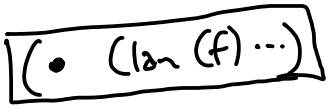
5

empty stack + value means termination

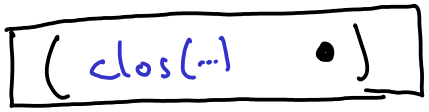
((lam (f) (f f)) (lam (f) (f f)))



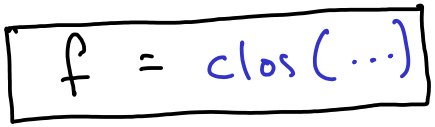
(lam (f) (f f))



clos(∅, f, (f f))

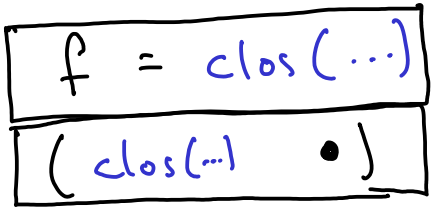


clos(⋯)

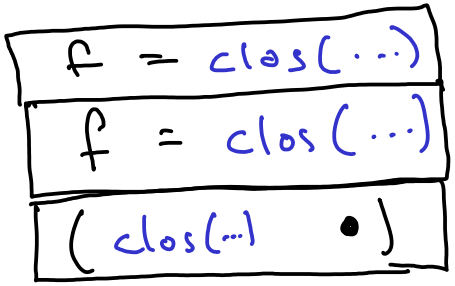


(f f)

⋯



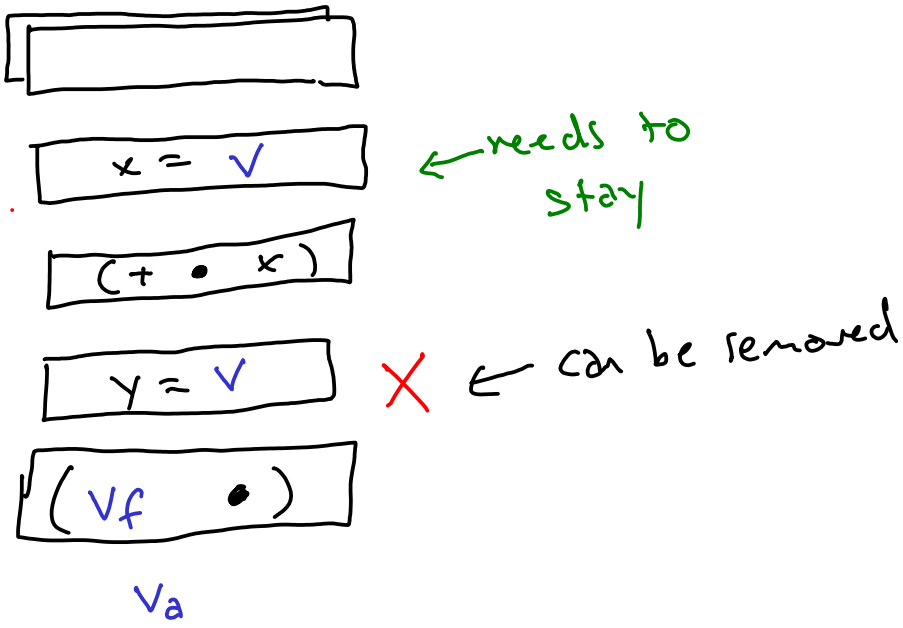
clos(⋯)



clos(⋯)

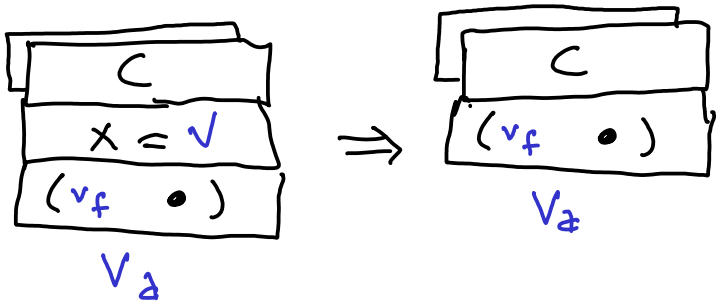
. . . .

STACK OVERFLOW




```
(let (y 5)
  ((lam (z) y) 4)))
```

SPACE OPTIMIZATION



$v_f = clos(C, z, e)$

if e refers to y
y must be in C

TAIL POSITION

TAIL CALL

PROPER TAIL CALLS

TCO