(let ([y (call/cc (call/cc

(call/cc call/cc)))])

(y list)

(y 4))

**r1: call/cc  
k1: (escaper (lambda (v)**

**(let ([y (call/cc (call/cc v))])**

**(y list)**

**(y 4))**

⇒ (let ([y (call/cc (call/cc (r1 k1)))])

(y list)

(y 4))

⇒ (let ([y (call/cc (call/cc (call/cc k1)))])

(y list)

(y 4))

**r2: k1**

**k2: k1**

⇒ (let ([y (call/cc (call/cc (k1 k1)))])

(y list)

(y 4))

⇒ (k1 k1)

⇒ (let ([y (call/cc (call/cc k1))])

(y list)

(y 4))

**r3: k1**

**k3: (escaper (lambda (v)**

**(let ([y (call/cc v)])**

**(y list)**

**(y 4))**

⇒ (let ([y (call/cc (k1 k3))])

(y list)

(y 4))

⇒ (k1 k3)

⇒ (let ([y (call/cc (call/cc k3))])

(y list)

(y 4))

**r4: k3**

**K4: k3**

⇒ (let ([y (call/cc (k3 k3))])

(y list)

(y 4))

⇒ (k3 k3)

⇒ (let ([y (call/cc k3)])

(y list)

(y 4))

**r5 : k3**

**k5: (escaper (lambda (v)**

**(let ([y v])**

**(y list)**

**(y 4))**

⇒ (let ([y (k3 k5)])

(y list)

(y 4))

⇒ (k3 k5)

⇒ (let ([y (call/cc k5)])

(y list)

(y 4))

**R6: k5**

**K6: k5**

⇒ (let ([y (k5 k5)])

(y list)

(y 4))

⇒ (k5 k5)

⇒ (let ([y k5])

(y list)

(y 4))

⇒ (k5 list)

(let ([y list])

(y list)

(y 4))

⇒ (begin (list list) (list 4))

⇒ (4)