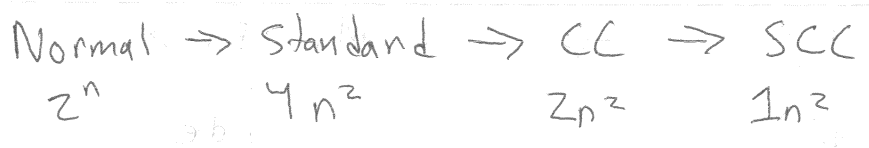


SCC machine $\langle M, E \rangle$

$$\langle (M \ N), E \rangle \xrightarrow{SCC} \langle M, E[(\ \ N)] \rangle$$

$$\langle V, E[(\ \ N)] \rangle \xrightarrow{SCC} \langle N, E[(V \)] \rangle$$

$$\langle V, E[(\lambda x, m) \] \rangle \xrightarrow{SCC} \langle M[\lambda \leftarrow V], E \rangle$$



- $E = \ \$ — new Hole();
- $\ | (E \ M)$ — new OnLeft(Evct E, Term M)
- $\ | (V \ E)$ — new OnRight(~~Val~~ V, Evct E)

\Downarrow

$E' = \text{list of (Frame)}$

- Frame = $(\ \ M)$ — new LeftFrame (Term M)
- $\ | (V \ \)$ — new RightFrame (Val V)

$$E_0 = ((\ \ M) \ N) \ P$$

$$E_0^R = (\ \ M) ; (\ \ N) ; (\ \ P)$$

$$E_0' = (\ \ P) ; (\ \ N) ; (\ \ M)$$

$en : E \rightarrow E'$ $de : E' \rightarrow E$

$$en(\ \) = \perp$$

$$en(E \ M) = (\ \ M) ; en(E)$$

$$en(V \ E) = (V \) ; en(E)$$

$$de(\perp) = \ \$$

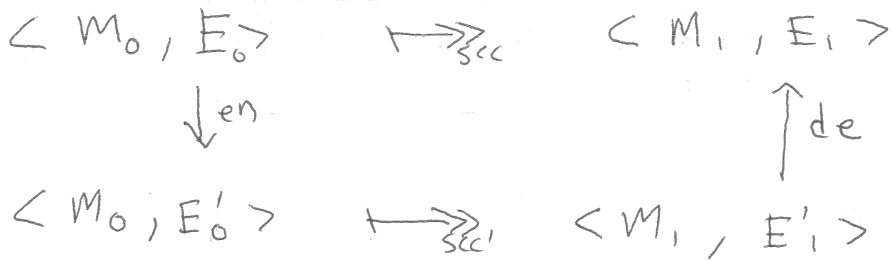
$$de((\ \ M) ; E') = (\ \ M)[de(E')] = (de(E') \ M)$$

$$de((V \) ; E') = (V \ de(E'))$$

8-2)

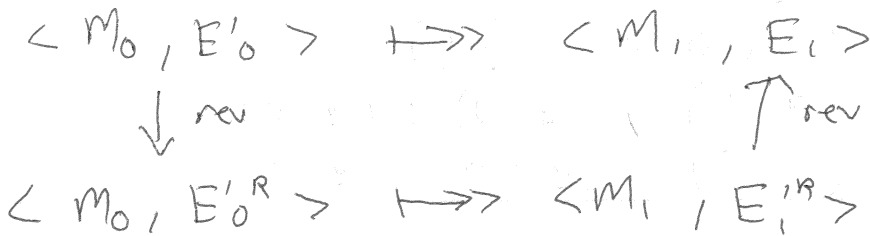
SCC' $\langle M, E \rangle$

- $O(n)$ $\langle (M N), E' \rangle \xrightarrow{SCC'} \langle M, E' ++ (\perp N) \rangle$
- 2. $O(n)$ $\langle V, E' ++ (\perp N) \rangle \xrightarrow{SCC'} \langle N, E' ++ (V \perp) \rangle$
- 2. $O(n)$ $\langle V, E' ++ ((\lambda X.M) \perp) \rangle \xrightarrow{SCC'} \langle M[X \leftarrow V], E' \rangle$



SCC^R $\langle M, E^R \rangle$

- $O(1)$ $\langle (M N), E^R \rangle \xrightarrow{SCC^R} \langle M, (\perp N); E^R \rangle$
- 2. $O(1)$ $\langle V, (\perp N); E^R \rangle \xrightarrow{SCC^R} \langle N, (V \perp); E^R \rangle$
- $O(1) + O(n)$ $\langle V, ((\lambda X.M) \perp); E^R \rangle \xrightarrow{SCC^R} \langle M[X \leftarrow V], E^R \rangle$



$en^R : E \rightarrow E^R$

$en^R = en \circ rev$

$en^R(\perp) = \perp$

$en^R((E N)) = \cancel{(\perp N)} en^R(E) ++ (\perp N)$

$en^R((V E)) = \cancel{(V \perp)} en^R(E) ++ (V \perp)$

$de^R : E^R \rightarrow E$

$de^R(\perp) = \perp$

$de^R((\perp N); E^R) = de(E^R) [(\perp N)]$

$de^R((V \perp); E^R) = de(E^R) [(V \perp)]$

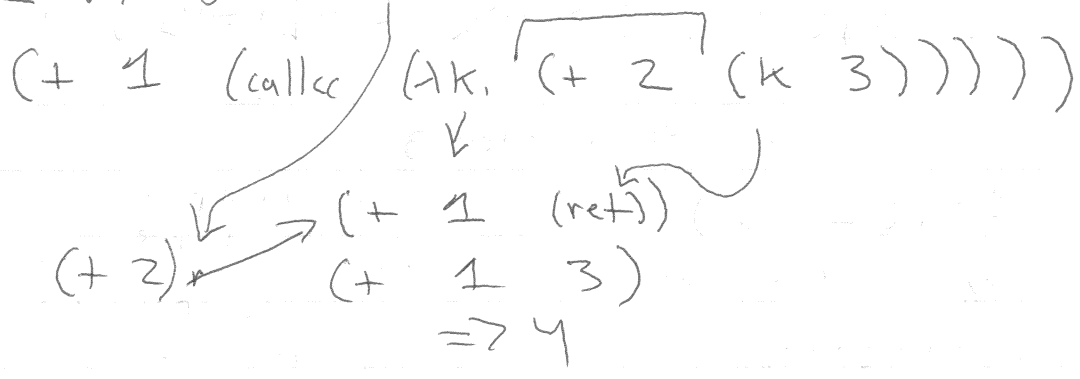
8-4/

$M = \dots$
| (called M)

$\langle \text{(called } M), k \rangle$
 $\mapsto \langle M, \text{fun}(\underline{k}, k) \rangle$

$\langle k, \text{arg}(\lambda X.M, k') \rangle \mapsto \langle M[X \leftarrow k], k' \rangle$

$\langle V, \text{arg}(K, k') \rangle \mapsto \langle V, k \rangle$



every language has continuation
some langs can implement them with stacks

the langs that can't, typically have the feature
"first-class continuations"

$K = \dots \mid \text{prim}(o^n, v \dots, m \dots)$

$\langle (o^n N M \dots), k \rangle$
 $\mapsto \langle N, \text{prim}(o^n, \perp, m \dots, k) \rangle$

$\langle V, \text{prim}(o^n, v' \dots, N M \dots, k) \rangle$
 $\mapsto \langle N, \text{prim}(o^n, v; v' \dots, m \dots, k) \rangle$

$\langle V, \text{prim}(o^n, v' \dots, \perp, k) \rangle$
 $\mapsto \langle \delta(o^n, (v; v' \dots)^R), k \rangle$

CK-machine

$\langle M, k \rangle$

↓
C = control string

↘
k = ... Stack*

k = ret

| fun (~~M, k~~ N, k)

"Continuation"

| arg (V, k)
L

Continuation
R

1. $\langle (M\ N), k \rangle \xrightarrow{ck} \langle M, \text{fun}(N, k) \rangle$
2. $\langle V, \text{fun}(N, k) \rangle \xrightarrow{ck} \langle N, \text{arg}(V, k) \rangle$
3. $\langle V, \text{arg}(X, M, k) \rangle \xrightarrow{ck} \langle M[X \leftarrow V], k \rangle$

ASM

arr[17] (x+21)
↓
requires 4 register
arr * x + y * z + g

;

compute f
push r3
use r0...r3
remember f in r2
pop r3

;

compute arg
push r2
...

arg is in r1
pop r2

ret = r, f, da, dx, rx
fun fx
arg ax

