

Mutation (State)

$$f(n) = \sum_{i=0}^n i$$

```

letrec f = λn. if n==0 then 0
              else n + (f (n-1)) in } recursive
f 10

```

```

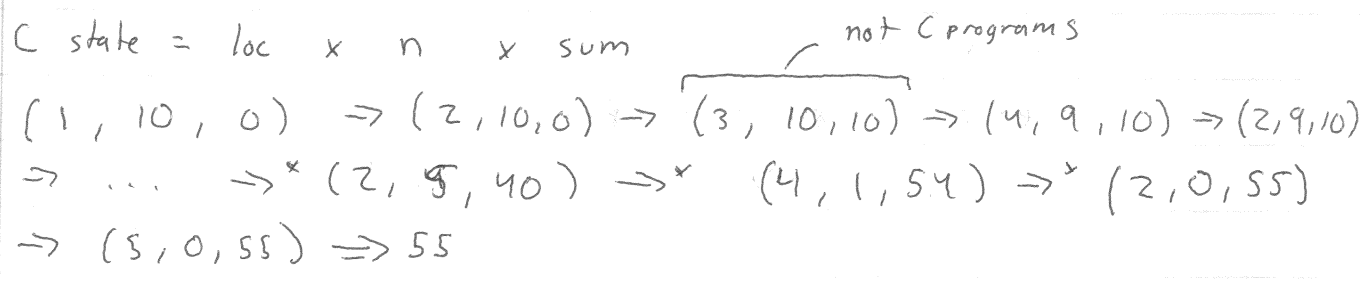
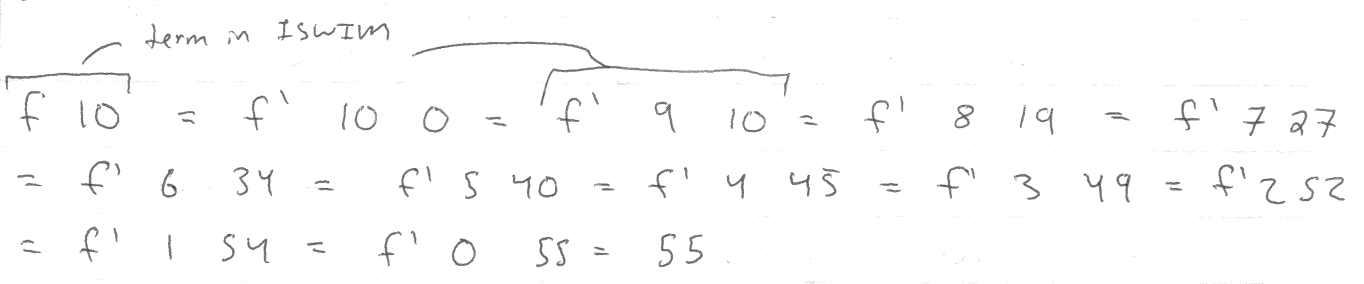
letrec f' = λn. λa. if n==0 then a
                  else f' (n-1, n+a) } tail-recursive
f = λn. f' n 0 in
f 10

```

```

int f (int n) {
  ① int sum = 0;
  ② for while (n > 0) { ③ sum += n; ④ n--; }
  ⑤ return sum;
}

```



q-2/

Compositional Reasoning

ISWIM: $(\dots \boxed{(lambda x) 5} \dots \boxed{m})$

Can you answer Q? What does \uparrow or \curvearrowright do?

Yes, if I know the bindings (FCV(m) = \emptyset , definitely)

$$FV(m) = \{x, y, z\}$$

$\dots ((lambda x. \dots ((lambda y. \dots ((lambda z. \dots \boxed{m}) \dots z) \dots y) \dots x))$

C:

```

int x = 5;
    ...
printf("%d", x);
    ...
return x;

```

Semantics is a homo-morphism

$$f : B \rightarrow C, g : A \times A \rightarrow B$$

$$f(g(x, y)) = g'(f'(x), f'(y))$$

$$f' : A \rightarrow D \quad g' : D \times D \rightarrow C$$

$$g(x, y) = (APP \ x \ y) \quad "(x \ y)" \quad f = eval$$

$$f' = eval \quad g'(x, y) = eval (APP \ x \ y)$$

```
int x = 1;
```

```
return (x++) + (x++);
```

comp. R. $\Rightarrow 4$ actual $\Rightarrow 5$

19-3/

State - ISWIM

(Matthias Felleisen)

$$M := (\text{let } ([X \ m]) \ M) \mid \dots \mid (\text{set! } X \ M)$$

$$E = (\text{let } ([X \ E]) \ M) \mid (\text{let } ([X \ V]) \ E) \mid \dots$$

$$\mid (\text{set! } X \ E)$$

$$E[(\text{let } ([X \ V]) \ E' [X])] \mapsto E[(\text{let } ([X \ V]) \ E' [V])] \\ \text{FV}(E'[X]) \ni X$$

$$E[(\text{let } ([X \ V]) \ E'[(\text{set! } X \ V')])] \\ \mapsto E[(\text{let } ([X \ \underline{V}]) \ E'[V'])]$$

$$E[(\text{let } ([X \ V]) \ V')] \mapsto E[V'] \\ \downarrow (\lambda \psi. X)$$

$$M := \dots \mid (\text{set! } X \ M)$$

$$P := (\text{let } ([X \ V] \dots) \ M)$$

$$(\text{let } (\dots [X \ V] \dots) \ E[(\text{set! } X \ V')]) \\ \mapsto (\text{let } (\dots [X \ V'] \dots) \ E[V'])$$

$$\sigma = \text{some set distinct from } X \quad (\sigma_1, \sigma_2, \sigma_3)$$

$$M = \dots \mid \sigma \mid (\text{set! } \sigma \ M) \quad \cancel{(\text{let } ([X \ V]) \ M)}$$

$$P = (\text{with } ([\sigma \ V] \dots) \ M)$$

$$(\text{with } ([\sigma_1 \ V_1] \dots [\sigma_n \ V_n]) \ E[(\lambda X. M) V])$$

$$\mapsto (\text{with } ([\sigma_1 \ V_1] \dots [\sigma_n \ V_n] [\sigma_{n+1} \ V]) \ E[M[X \leftarrow \sigma_{n+1}]])$$

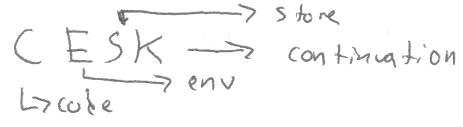
$$(\text{with } ([\sigma_1 \ V_1] \dots [\sigma_x \ V_x] \dots [\sigma_n \ V_n]) \ E[\sigma_x])$$

$$\mapsto (\text{with } (\dots) \ E[V_x])$$

$$E[(\text{set } \sigma_x \ V)]$$

$$\mapsto (\text{with } (\dots [\sigma_x \ V] \dots) \ E[V])$$

19-4



$N, M = X \mid (\lambda X.M) \mid (M N) \mid (\text{set! } X M) \mid b \mid 0^n M \dots$

$E = \bullet \mid [X \mapsto \sigma] E$

$S = \bullet \mid [\sigma \mapsto V] S$

$V = b \mid \text{clo}(\lambda X.M, E)$

$K = \text{ret} \mid \text{fun}(N, E, K) \mid \text{arg}(V, K) \mid \text{set}(\sigma, K) \mid \text{op}(0^n, E, V, M, K)$

$\langle X, E, S, K \rangle \mapsto \langle S(E(X)), \bullet, S, K \rangle$

$\langle \lambda X.M, E, S, K \rangle \mapsto \langle \text{clo}(\lambda X.M, E), \bullet, S, K \rangle$

$\langle M N, E, S, K \rangle \mapsto \langle M, E, S, \text{fun}(N, E, K) \rangle$

$\langle V, E, S, \text{fun}(N, E', K) \rangle \mapsto \langle N, E', S, \text{arg}(V, K) \rangle$

$\langle V, E, S, \text{arg}(\text{clo}(\lambda X.M, E'), K) \rangle$

$\mapsto \langle M, E'[X \mapsto \sigma'], S[\sigma' \mapsto V], K \rangle$

$\langle \text{set! } X M, E, S, K \rangle \mapsto \langle M, E, S, \text{set}(E(X), K) \rangle$

$\langle V, E, S, \text{set}(\sigma, K) \rangle \mapsto \langle V, E, S[\sigma \mapsto V], K \rangle$

$M = \dots \mid \text{snapshot} \mid \text{restore } M \quad K = \dots \mid \text{restore}(K)$

$V = \text{sto}(S) \mid \dots$

$\langle \text{snapshot}, E, S, K \rangle \mapsto \langle \text{sto}(S), E, S, K \rangle$

$\langle \text{sto}(S'), E, S, \text{restore}(K) \rangle \mapsto \langle I, E, S', K \rangle$

$\langle \text{restore } M, E, S, K \rangle \mapsto \langle M, E, S, \text{restore}(K) \rangle$