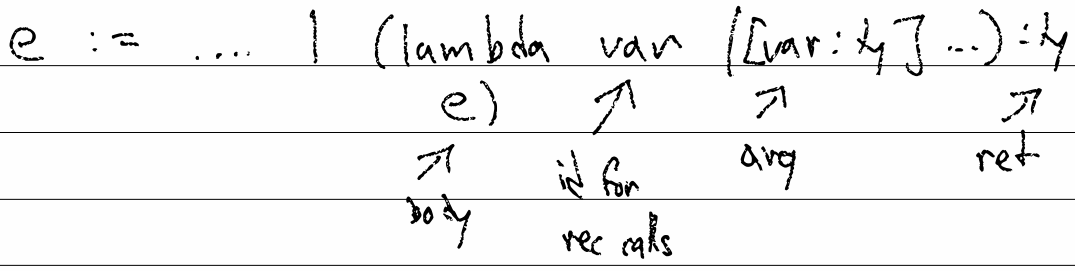


16-1 / $f = g$

$$\forall x. f x = g x$$

$R_4 \rightarrow R_5$



$((\lambda \text{ fac } ([n: \text{sg4}]) : \text{sg4}$
 (if (= 0 n) 1
 (* n (fac (- n 1))))))

5)

for (int i=0; i < 40; i++) { ←

 for (int j=0; j < 60; j++) {

 if ...

 continue —————

 }

}

- 16-y) type of closure
- binding rec
- existing top-levels

rec ($\lambda r (a \dots) e$) \Rightarrow

(define (nf [r] [a] ...) e)

\nearrow
the name used for 'rec'

H] (define (sub1 [x:504]) : 504 (- x 1))
(sub1 4) \Rightarrow

~~(app (vector-ref sub1 0)
sub1 4)~~

(fun-ref f) \xrightarrow{clo} (vector (fun-ref f))

(define (even?) ... odd?)

(define (odd?) ... even?)

clo] ($\lambda r ([x:504]) : 504 (+ x z)$)

\Rightarrow (vector nf 42 z) : ~~vector~~ V

nf 42 : F

F \neq V x 504 \rightarrow 504 = (vector trouble 504)

V = vector F 504 x 504 \rightarrow 504

16-5 cloconv : $e \rightarrow e \quad x \quad (\text{list of defs})$
 \Rightarrow

are all the
lambda bodies

$$\begin{aligned} & \text{cloconv } (+ \ e_L \ e_R) \\ &= (+ \ e_L' \ e_R') \\ & \times (l_L \ ++ \ l_R) \end{aligned}$$

where $(e_L', l_L) = \text{cloconv } e_L$
 $(e_R', l_R) = \text{cloconv } e_R$

more instructions — more dep on arch
 instruction scheduling turns "A" code
 like

$$\begin{aligned} z &= a * 2 \\ \Rightarrow \text{shl } a, \mathbb{I} \\ &\text{mov } q, z \end{aligned}$$

optimization $\Rightarrow \quad i = [0, 80] \quad ; ++$
 $\Rightarrow i = [0, 80] \quad ++ = 8$

$$\begin{aligned} & (+ \ \underbrace{(- \ x \ 1)}_{(+ \ x \ (-1))} \ 1) \Rightarrow x \\ & \quad (+ \ (-1) \ x) \quad \nearrow \\ & (+ \ (+1 \ -1) \ x) \quad (+ \ 0 \ x) \end{aligned}$$

$$\underline{16-6)} \quad (\text{map } f \ (\text{map } g \ L)) \Rightarrow \\ (\text{map } (\lambda (x) \ (f \ (g \ x))) \ L)$$