

15-1 /  $C_2 \Rightarrow C_3$

$exp ::= \dots \mid (fun-ref\ label) \mid (call\ arg\ arg\ \dots)$

$tail ::= \dots \mid (tail/call\ arg\ arg\ \dots)$

$def ::= \dots \mid (define\ (label\ [var = ty]\ \dots) = ty$   
 $\quad info\ [label\ tail]\ \dots)$

$P ::= (program\ info\ (def\ \dots)\ label)$

15-2

$\text{econ } (\text{rp info ds } (\text{app } (\text{framef main}))) =$   
 $(\text{cp info } (\text{map } \text{econ ds}) \text{ main})$

$\text{econ } (\text{define } (f \text{ [arg}_0\text{:t}_1] \dots \text{ [arg}_n\text{:t}_n])) : \text{rty } e) =$   
 $(\text{define } (f \text{ [arg}_0\text{:t}_1] \dots \text{ [arg}_n\text{:t}_n])) : \text{rty } L')$

where  $L, t = \text{econ } e$

$L' = L[\text{body} \mapsto t]$

$\text{econ } (\text{let } x = (\text{app } \text{rator } \text{rands}) \text{ in } \text{body})$   
 $(\text{segn } (\text{set! } x (\text{call } \text{rator } \text{rands})) (\text{econ } \text{body}))$

$\text{econ } (\text{app } \text{rator } \text{rands}) = (\text{tail call } \text{rator } \text{rands})$

15-3/ uncover-locals  $\rightarrow$  operate funs  
vs progs

$X_2 \rightarrow X_3$

arg := .... | fun-ref label

instr := .... | (indirect-callg arg)  
| (tailjmp label)

emit (indirect-callg arg)  $\rightarrow$  callg \*arg

e.g. callg = %rbx

(indirect-callg (fun-ref label))  $\Rightarrow$  callg label

emit (fun-ref label)  $\rightarrow$  label (%rip) (might need  
leag)

15-4/

emit (tail jmp label)

BEGIN: save callee regs

movq %rsp, %rbp

subq \$ss, %rsp

jmp BODY

BODY: ..... callq g

jmp END

END: addq \$ss, %rsp

restore callee regs

retq

jmp label

postlude

15-5/

selectp (cprog ; ds lab) = (xprog ; (map selectd ds) lab)

arg a1: reg5 = < r1i r1s r1x r1c r1e r1f r1g >

selectd (define i (f [a0:ty0] ... [an:ty\_n])) : rty

[lab  $\Rightarrow$  tail] begin-label)

= (define i (f) lab  $\Rightarrow$  blk' start-label)

where lab  $\Rightarrow$  blk = map select+ lab  $\Rightarrow$  tail

lab  $\Rightarrow$  blk' = lab  $\Rightarrow$  blk [START  $\Rightarrow$

(block  $\emptyset$  (movq r0 a0)

...

(movq r5 a5)

(jmp begin-label))

15-6)

selecte dst (call rator rands) =

movq a0 r0  
...  
movq a5 r5

} ①

ret callq - (selecta rator)

movq %rax, dst

selectq (tail call rator rands) =

① ; tail jmp (selecta rator)

15-7/

conflicts

tail-jmp → writes callee-saves regs  
indirectly → reads the array for the return

assign-homes per fun

needs to do some of the job that main

runtime.c

ty\_fun

print\_value

" it's a fun "

" # c function → tree "