9-1/ $C_0 \to C_1$

$C_1 ::=$ arg ::= .... | true | false

cmp ::= = | < | ≤ | ≥ | >

exp ::= ..... | (not arg) | (cmp arg arg)

tail ::= .... | (goto label) | (gotoif (cmp arg arg) label label)

$L_{(lab) \Rightarrow tail}$

interp$_C$ (goto lab) = interp$_T$ L (lab)

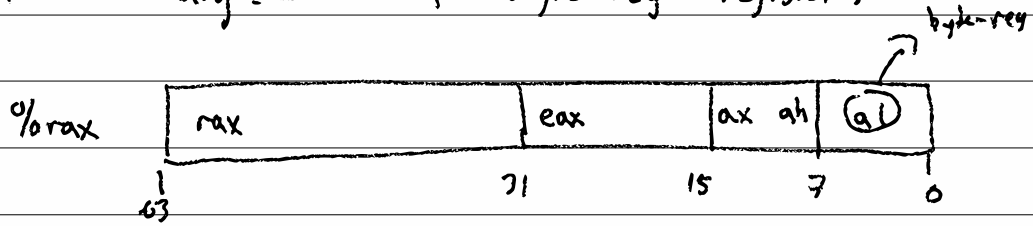
interp$_C$ L (goto-if (cmp a_L a_R) lab_T lab_F) =

interp$_T$ L (lab_K) where lab_K = lab_T if v o.w lab_F

v = interp$_E$ (cmp a_L a_R)

9-2/ $X_0 \rightarrow X_1$

$X_1 ::=$ arg := .... | (byte-reg register)



cc := e | l | le | g | ge

instr := .... | xorq arg, arg | cmpq arg, arg

| set cc, arg

| movzbq arg, arg

| (jmp-if cc label)

5 < 4 → cmpq $4, $5 "jmp" cc ⌣ label

set l , %al

"setl"

cmpq cannot have a constant in the 2nd spot

movzbq also

9-3)

r10-e := ... | (if (cmp a a) e e)

r10-c := ... | (not a) | (cmp a a)
| (if (cmp a a) e e)

a := ... | true | false

9-4)

(+ (if (< (read) 5) 17 (+ 8 (+9 10)))
(+ (read) 21))

=>

let v0 := (read)
let v1 := if (v0 < 5)
then 17
else let v_2 := 9 + 10
let v_3 := 8 + v_2
v_3
let v4 := (read)
let v5 := v4 + 21
let v6 := v_1 + v5
v6

9-5)

let* : List (Pair (var, e)) × e → e

rco_p : prog ⇒ prog

rco_p (program _ e) = (program ∅ (rco_e ∅ true e))

rco_e : (x ⇒ e) × Bool × e → e

rco_e σ tail? e = let* nv a

where (nv, a) = rco_a σ tail? e

rco_a : (x ⇒ $\overset{\downarrow a}{e}$) × Bool × e ⇒ List (Pair var $\overset{\downarrow e}{e}$) × $e^{\downarrow arg}$

rco_a σ tail? (Var x) = (mt, σ(x))

(Num n) = (mt, Num n) (Bool b) = (mt, Bool b)

(Read) = ([(readvar, (Read))], readvar)

(Una e_L) = (nv_L ++ [(unavar, Una a_L)], unavar)

where (nv_L, a_L) = rco_a σ false e_L

(Bin e_L e_R) = (nv_L ++ nv_R ++ [(binvar, Bin a_L a_R)], binvar)

where (nv_L, a_L) = rco_a σ false e_L

(nv_R, a_R) = rco_a σ false e_R

(Let x xe be) = (nv_x ++ nv_b, a_b)

where (nv_x, a_x) = rco_a σ false xe

(nv_b, a_b) = rco_a σ[x ↦ a_x] tail? be

9-b)

rco$_a$ σ tail? (If e_c e_t e_f) =

if tail? then

(nv_c, ; f')

else (nv_c ++ [(ifun, if')], ifun)

where

(nv_c, cmp_c, a_L, a_R) = rco$_c$ σ e_c

if' = If (cmp_c a_L a_R)

(rco$_e$ σ tail? e_t)

(rco$_e$ σ tail? e_f)

9-7/

$rco_c : (X \to e^{\uparrow\wedge}) \; e \Rightarrow (\text{newvars}, cmp, a, a)$

$rco_c \; \sigma \; (cmp \; e_L \; e_R) = (nv_L +\!\!+ nv_R, \; cmp, \; a_L, a_R)$

where $(nv_L, a_L) = rco_a \; \sigma \; \text{false} \; e_L$

$(nv_R, a_R) = rco_a \; \sigma \; \text{false} \; e_R$

$rco_c \; \sigma \; (\text{Let} \; x \; xe \; be) = (nv_x +\!\!+ nv_b, \; op_b, a_L, a_R)$

where $(nv_x, a_x) = rco_a \; \sigma \; \text{false} \; xe$

$(nv_b, op_b, a_L, a_R) = rco_c \; \sigma[X \mapsto a_x] \; be$

$rco_c \; \sigma \; other = (nv, =, true, a)$

where $(nv, a) = rco_a \; \sigma \; \text{false} \; other$

9-8) econ : R → C

$$\text{econ}_p\ (\text{program}\ -\ b) = (\text{program}\ \emptyset\ L')$$

where $L' = L\,[\text{BODY} \mapsto t_b]$

$(L, t_b) = \text{econe}\ (\lambda\ (fa)\ (\text{Return}\ fa))\ b$

econe : (C arg → C tail) × R expr → Label → Tail × Tail

1) econe k (Var x) / (Num n) / (Bool b) = (∅, k (econa boring))

econe k (Let x xe (notan if) be) =

(L, (seqn (set! x (econe xe)) bt))

where (L, bt) = econe k be

2) econe k (If (cmp aL aR) et ef)) =

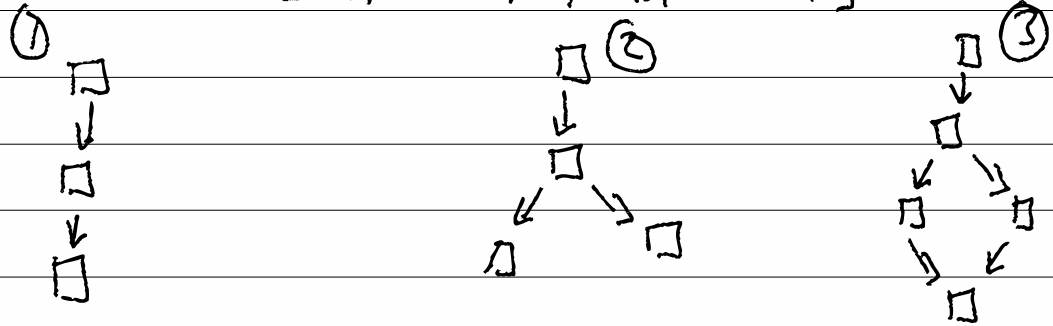
(Lt ∪ Lf ∪ New, goto_if (cmp a'L a'R) labt labf)

where (Lt, tt) = econe k et

(Lf, tf) = econe k ef

$a_L' = \text{econa}\ a_L \qquad a_R' = \text{econa}\ a_R$

New = [labt ↦ tt, labf ↦ tf]



9-9/

econe k (Let x (If (cmp a_L a_R) et ef) e_b) =

(Lt ∪ Lf ∪ L_b ∪ New,

goto-if (cmp a_L a_R) labt labf)

where

(L_t, t_t) = econe nk et

(L_f, t_f) = econe nk ef

(L_b, t_b) = econe k e_b

New = [$lab_t \mapsto t_t$, $lab_f \mapsto t_f$, $lab_b \mapsto t_b$]

nk = (λ (fa) (seq (set! x fa)

(goto lab_b)))

9-10/

```
let v0 := (read)
let v1 := if (v0 < 5)
  then 17
  else let v2 := 9 + 10
       let v3 := 8 + v2
       v3
let v4 := (read)
let v5 := v4 + 21
let v6 := v1 + v5
v6
```

```
[body ↦ seq (set! v0 (read))
            gotoif (v0 < 5)
                   lab1 lab2 ,
lab1 ↦ (seq (set! v1 17)
            (goto lab3) ,
lab2 ↦ (seq (set! v2 (9+10))
            seq (set! v3 (8 + v2))
            seq (set! v1 v3)
            (goto lab3) ,
lab3 ↦ seq (set! v4 (read))
       seq (set! v5 (v4 + 21))
       seq (set! v6 (v1 + v5))
           v6 ]
```