12-1// resolve-complex / rco

collect → expr / complex

allocate → expr / complex

global → arg

unit → arg

vector-ref → expr / complex

vector-set → expr / complex

let v = $\underset{(unit)}{c}$

(let v = c in e)

(seqn c e)

12-2/ econ

$C_1 \to C_2$

arg := .... | (global str) | (unit) | (var$^{\nearrow ty}$: ty)

exp := .... | (allocate num ty) | (vector-ref arg num)

stmt := .... | (collect num) | (vector-set! arg num arg)

econ (let x = (allocate num ty) in body) =

seqn (set! x (allocate num ty)) (econ body)

econ (let _ = (collect num) in body) =

seqn (collect num) (econ body)

12-3) uncover-locals

old ans: info about names of variables

new ans: variables and their types

12-4/ $X_1 \to X_2$

arg := ... | (global str) | (type ty)

instr := ... | leaq arg, arg

↗ load effective address (like & in C)

emit (global str) = $\overset{\text{"\_"}}{\text{str}}$(%rip)

emit (type ty)

emit (type (Vector S64 Bool S64)) = $\overset{\text{.rodata}}{\text{read-only}}$ region (section)

type27(%rip)

type27:

.quad 3 — Vector

.quad 3 → 3 elements

.quad 1 — S64

.quad 2 — Bool

.quad 1 — S64

0 ↦ Unit

1 ↦ S64

2 ↦ Bool

3 ↦ Vector

12-5/ C ⇒ X (select)

```
selecta (global str) = (global st)    // str(%rip)
selecta (unit)       = $1

selecte^dst (allocate num ty) =
    movq (global "free_ptr"), dst
    addq $(8 x (1+num)), (global "free_ptr")
    movq dst, %rax
    leaq ty, %r11            ~~leaq ty, dst(0)~~
    movq r11, %rax(0)
```

12-6/

select$_e$ dst (vector-ref arg num) =

movq (select$_a$ arg) , %rkk

movq %~~ar~~kk (8 x (1+num)) , dst

select$_s$ (vector-set! va num na) =

movq (select$_a$ va), %rax

movq (select$_a$ na), % r11

movq %r11, %rax (8 x (1+num))

12-7/

Select$_s$ (collect num) =

```
    movq ROOT-STACK-REG, %rdi // 1st arg
    movq $num, %rsi  // (2nd arg)
    callq _collect
```

12-8/ live ×
conflicts ×
assign
patch
main

Live/ leaq is just movq

conflicts/ leaq just like movq

update callq

old:

$\forall v \in LA_k$. $\forall r \in$ CALLER-SAVED.

add (r, v) to I

new. old v

$\forall v \in LA_k$. where $\Gamma(v) =$ Vector ...

$\forall r \in$ CALLEE-SAVED

add (r, v) to I

12-9

assign

old: vars either go to a register or the stack

↙ ≤13

↓ %rbp (8 × (c-13))

new: split arguments into three sets:

- registers ↓ %r8
- vector variables on the stack ↓ ROOT-STACK (8 × which num)
- non-vector vars on the stack ↓ %rbp (8× which num [illegible])

patch leaq is like movq

12-10

main/ BEGIN:

```
old:  push all of the used callee-saved register
      movq  %rsp , %rbp
      subq  SS , %rsp
      jmp   BODY


      callq  _initialize
      movq   _root_stack_start , root-reg
      subq   root-ss, root-reg
```

12-11

main: END:

```
old:  movq  %rax, %rdi
      callq _print_int  or  _print_bool

new:  leaq  ans-type, %rdi
      movq  %rax, %rsi
      callq _print_val
```