12-1 instruction selection

(set! var (vector-ref vec-~~expr~~ arg n))

=> (movq vec-arg %rax) vec[n] → rbp[8][n]
(movq (deref %rax $(* 8 (n+1))) var)

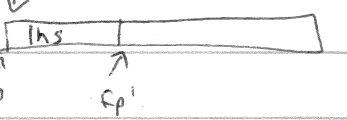
(deref x y)
=> x[y]

(vector-set! v n a)
=> (movq v %rax)
(movq a (deref %rax $(* 8 (n+1))))

(set! lhs e) (void)
=>
(movq (int 1) lhs)

Richard Cobbe
Jawa w/o NULL

~~(allocate~~ (set! lhs (allocate n ty))
=> ① (movq free-ptr lhs)
(addq ~~n~~ $(*8 (n+1)) free-ptr)
(movq lhs %rax)
(movq "ty" %rax[0])



① => movq _free-ptr(%rip), lhs (mode A
② => movq %r15, lhs

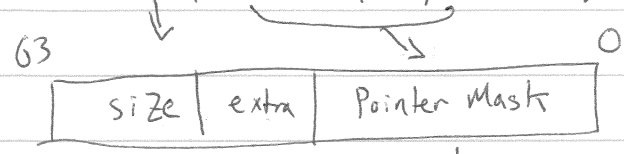
Vec, n fields, ptr, !ptr, ptr, !ptr ...)

10: Vec, 4, 9, 7, 1, 0, funs
0: Bool
1: int
7: Void
9: Vec, 2, 1, 1, funs

.text
.globl _main
_main: ret
movq 8, %rax
.quad 21
tyBool: 0
tyVec18: 4, 4, tybool,

...
f()
...

Vec, 4, 0, 1, 1, 0) >= 64-bits



$n + 2^n = 64$

n → 2^n

$n = 5$ → 5 → 32) → 37 bits

6 ⇐ 64) → 70 bits

OR 6 bits + 58 pointer mask (illegal bit combinations)

(collect bs) constant

```
(movq rootstack %rdi)
(movq bs %rsi)        <-- spill all vector vars
(callq _collect)      <-- unspill
```

i2 → %rcx G V1 → %rbx

⇒ %rsp(-16) B ⇒ _rootstack(-16) VB