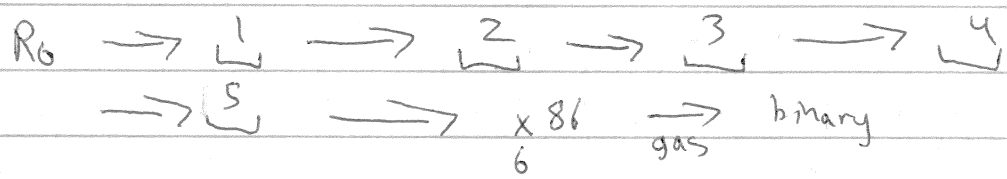


2-1 → (read) (file → value path)
 (+ 10 32)
 ⇒ '(+ 10 32)

Ro	asm
operation & take expressions (+ 1 (+ 2 3)) ↳ vs produce new value	inst take 1 or 2 arguments ↳ must be num, mem, reg addq 1 (addq 2 3) ↳ one arg must be mem/reg ↳ that same arg is the dest
infinite variables	few registers ↳ infinite memory
shadowing [let ([x ...]) [let ([x ...]) ...]]	registers are unique



6 print	x86-tree	x86-string
job: display asm source	(program int inst+)	
	↓ frame-size	

1 unique	Ro	Ro
job: shadowing vs unique		
(let ([x 10]) (+ x (let ([x 20]) x)))		
⇒ ↓	↓	↓
(let ([x ₁ 10]) (+ x ₁ (let ([x ₂ 20]) x ₂)))		

(gensym 'x) = 'x9021 'x1729

2 | Flatten | R0 | Co

job: remove nested expressions

"like"

$(+ 1 (+ 2 3)) \implies$ $x = 2 + 3$
 $(+ (+ 1 2) (+ 3 4)) \implies$ $ret\ 1 + x$
 $x = 1 + 2$
 $y = 3 + 4$
 $ret\ x + y$

$(+ 1 (let ([x 5]) (+ x 3)))$
 lhs1 rhs1 body lhs2 rhs2

- ① lhs1 = 1
- ② x = 5
- ③ lhs2 = ~~x~~
- ④ rhs2 = 3
- ⑤ body = lhs2 + rhs2
- ~~rhs1 = x + 3~~
- ⑥ rhs1 = body
- ⑦ ret lhs1 + rhs1

\implies body = 5 + 3 \implies x = 5
 $ret\ 1 + body$ $body = x + 3$
 $ret\ 1 + body$

Co = (program (var*) stmt+)
 stmt = (!= var expl | (ret arg))
 exp = arg | (read) | (- arg) | (+ arg arg)
 arg = int | var

3 | select-instruction | Co | X* (x86-tree)

job: turn exp of Co into X inst assuming everything is a register

$addq\ \$10, !x$ $addq\ \$10, \%rax$
 $[body = 5 + 3] \implies [movq\ \$3, !body$
 $ret\ 1 + body]$ $addq\ \$5, !body$
 $movq\ !body, \%rax$
 $addq\ \$1, \%rax$
 $retq$

$X^* = X$, but !var is a valid
 arg and
 program = (vs*) inst+

2-3

4 | assign-homes | X^* | X^0

job: map variables to memory (on the stack)

$X^0 = X$ (but may break some x86-rules)

frame setup +

$$\begin{bmatrix} \text{movq} & \$3, & !\text{body} \\ \text{addq} & \$5, & !\text{body} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{movq} & \$3, & -8(\%rbp) \\ \text{addq} & \$5, & -8(\%rbp) \end{bmatrix}$$

+ frame cleanup

$$\begin{bmatrix} \text{movq} & \$3, & !\text{body} \\ \text{movq} & \$8, & !\text{hs} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{movq} & \$3, & -8(\%rbp) \\ \text{movq} & \$8, & -16(\%rbp) \end{bmatrix}$$

(calculate the frame size)

5 | patch | X^0 | X

X^*

$$\begin{bmatrix} \text{movq} & !\text{lhs}, & !\text{rhs} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{movq} & -8(\%rbp), & -16(\%rbp) \end{bmatrix}$$

⇓

$$\begin{bmatrix} \text{movq} & -8, & \%rax \\ \text{movq} & \%rax, & -16 \end{bmatrix}$$

