

locals:

'(x y)

$R'' \rightarrow C_0$

Start:

1. $y = (-42);$
2. $x = y;$
3. $\text{return } (-x);$

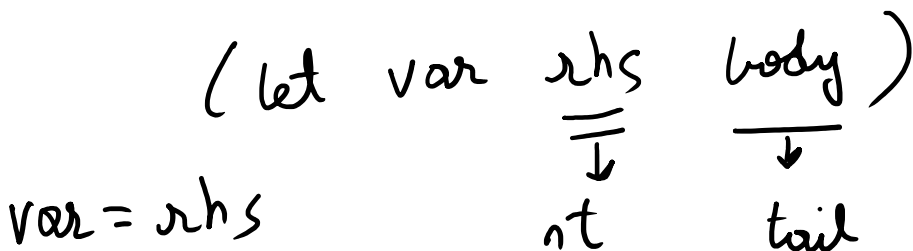
R''
 \downarrow
 tail R_1
 C_0

$\text{atom} ::= \text{var} \mid \text{int}$

$\text{nt} ::= \text{atom} \mid (\text{read}) \mid (-\text{atom}) \mid (+\text{atom } \text{atom}) \mid$
 $(\text{let } [\text{var } \underline{\text{nt}}] \text{nt})$

$\underline{\text{tail}} ::= \text{atom} \mid (\text{read}) \mid (-\text{atom}) \mid (+\text{atom } \text{atom}) \mid$
 $(\text{let } ([\text{var } \underline{\text{nt}}]) \underline{\text{tail}})$

$R'' ::= \text{tail}$



ret (body)

\mathcal{L}_0

atom ::= var | int

exp ::= atom (read) | (- atom) | (+ atom atom)

stmt ::= var = exp; \rightarrow nt

tail ::= return exp; | stmt tail \Downarrow tail

st.
st.
.
ret exp;

$\mathcal{L}_0 ::= \underline{\text{tail}}$

tail \rightarrow stmt tail \rightarrow stmt stmt tail

stmt; stmt; ret exp;

stmt; var₁ = exp₁; ret exp₂;

var₀ = exp₀; var₁ = exp₁; ret exp₂;

expl_tail : exp \rightarrow $\mathcal{L}_{\text{tail}}$ * var list

exp_assign : exp, var, $\mathcal{L}_{\text{tail}}$ \rightarrow $\mathcal{L}_{\text{tail}}$ * var list (Var, RHS)

$x = (+10\ 32)$

\downarrow \downarrow \downarrow

RHS var $\mathcal{L}_{\text{tail}}$

(var = RHS)

$\mathcal{L}_{\text{tail}}$

(1) { } , { }

local: (x y)

start: y = -42;
x = y;
return -x;

C0 → x86 {vars...}

x = (+10 32);

↓

{
 movq \$10, x
 addq \$32, x
}

return e; e'

movq e', rax

jmp conclusion

x = (+10 x);

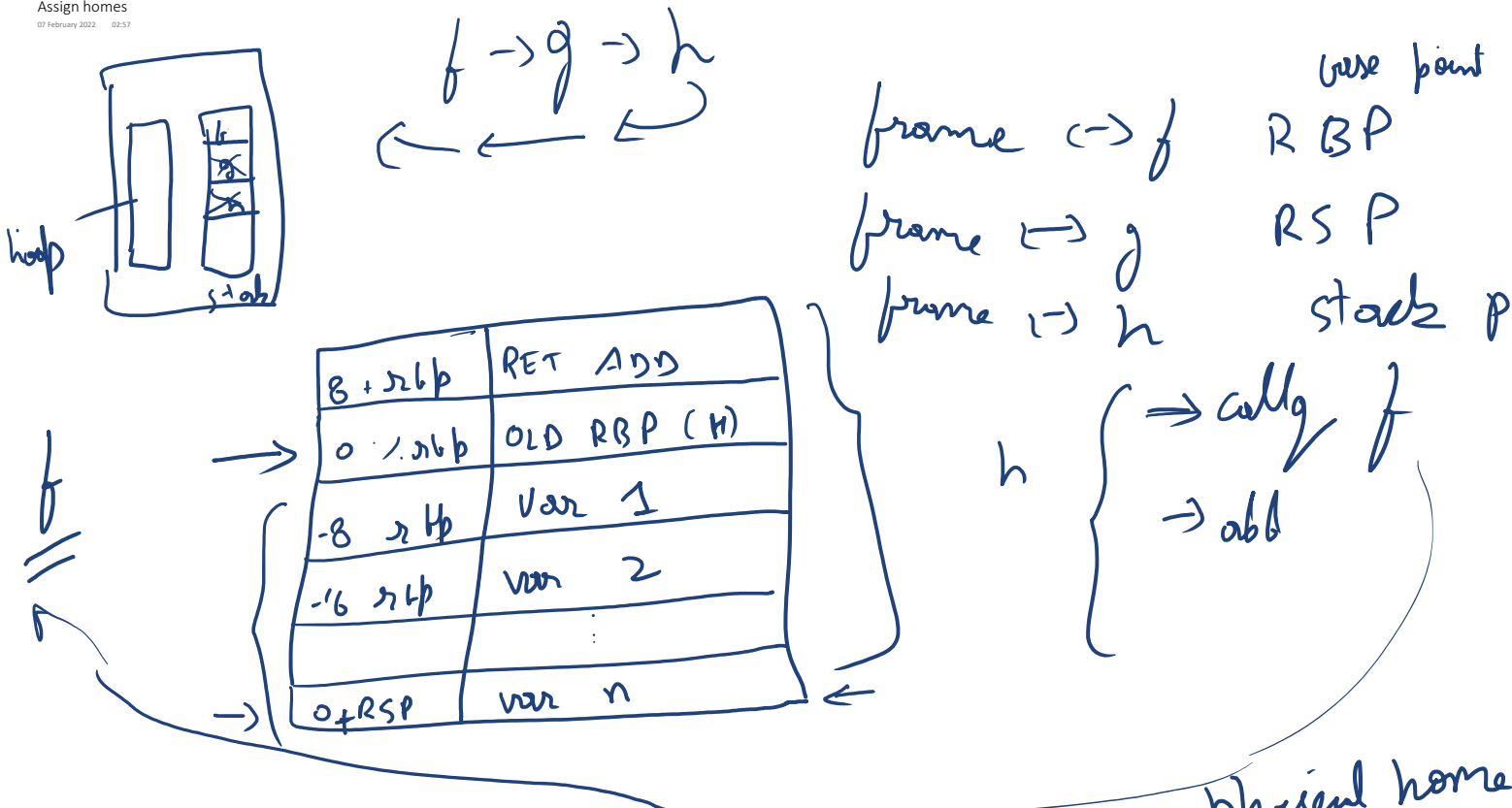
addq \$10 x

x = (read);

↓

callq read-int

movq %rax, x



var 1 $\rightarrow -8 (-\text{rbp})$
 var 2 $\rightarrow -16 (-\text{rbp})$

var \leftrightarrow physical home

x y $-8 (-\text{rbp})$
 $-16 (-\text{rbp})$

locals: (t.1 t.2)
 start:
 t.1 = (-52);
 t.2 = (-10);
 return (+ t.1 t.2)

select int \rightarrow

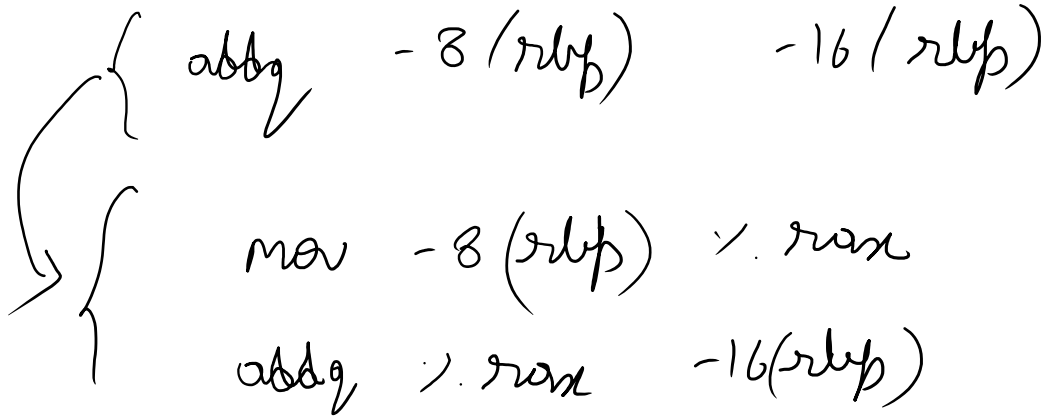
locals: (t.1 t.2)
 movz \$52 t.1
 negz t.1
 mov \$10 t.2
 negz t.2
 addq t.1 t.2
 movq t.2 %rax

t.1 $\rightarrow -8 (-\text{rbp})$ assign home
 t.2 $\rightarrow -16 (-\text{rbp})$ home

$t.1 \rightarrow \dots$
 $t.2 \rightarrow -16 (\text{v. rbf})$
 $-800 (\text{v. rbf})$
 abbg, $-8 (\text{rbf})$ $-16 (\text{rbf})$

x86

one memory access per inst



use ~~rdi~~
reserve reg

3x5x4 → 8x4, 3x20

{ exp ⇒ exp + exp | exp * exp | int | (exp)

3x5x4
 23
 { st : ~~st + st~~ st + exp
 exp: exp * exp | (exp) | int
 st

int * int + int