5-1

$M, N, L, K ::= X \mid \lambda X. M \mid M\ N \mid b \mid o^n\ M_1 \ldots M_n$

$V, U ::= \lambda X. M \mid b$

$\beta_v : (\lambda X. M)\ V \rightarrow M[X \leftarrow V]$

$\delta : (o^n\ V_1 \ldots V_n) \rightarrow \delta(o^n, V_1, \ldots, V_n)$

e.g. $\delta(+, 5, 10) = 15$

$v = \beta_v \cup \delta$

$\text{eval}(M) = b$ if $M =_v b$

'fun if $M =_v \lambda X. N$

Theorem: eval is a partial function.

$\iff \forall M. \forall b_1, b_2.$
$\text{eval}(M) = b_1$
$\wedge\ \text{eval}(M) = b_2$
$\rightarrow b_1 = b_2$

---

$\rightarrow_v$ is the compatible closure of $v$

"perform $v$ anywhere inside $M$"

"These two programs (M and N) do the same thing."

($\exists b.\ M \twoheadrightarrow_v b$ and $N \twoheadrightarrow_v b$) then we miss functions

$(\lambda X. \Omega)$ and $(\lambda X. (\Omega\ \Omega))$ "do the same thing"

$f$ and $g$, $f = g$ : $\forall i.\ f(i) = g(i)$

$\forall I.\ (M\ I) =_v (N\ I) \iff M$ and $N$ dtst

A (syntactic) context e.g. $= (\blacksquare\ I)$

can be "plugged" (or "filled") $(\blacksquare\ I)[N] = (N\ I)$

$C = \blacksquare \mid (\lambda X. C) \mid (C\ N) \mid (M\ C) \mid (o^n\ M \ldots C\ M \ldots)$

$\rightarrow_v$ is defined as (forall $C$, $C[M] \rightarrow_v C[N]$ iff $M\ v\ N$)

$((5 + 10)\ ((\lambda X. X)\ 6)) = (\blacksquare\ ((\lambda X. X)\ 6))[(5+10)]$ $\qquad 5+10\ v\ 15$

$(15\ ((\lambda X. X)\ 6)) = \rightarrow_v (\blacksquare\ ((\lambda X. X)\ 6))[15]$

Observational Equivalence ("Do the same thing") $M \simeq_v N$

iff $(\forall C.\ \text{eval}_v(C[M]) = \text{eval}_v(C[N]))$

refl, trans, sym

$M \simeq_v N$ then $\forall C.\ C[M] \simeq_v C[N]$

5-2

L is a language

F is a feature

L+F is another language

imagine M

M takes 1 call to F in L+F and translates it to L.

If $\forall p.$ $M(P) \simeq P$ meaning there's no context to tell the translation from the original l.

expressiveness

Optimizer replaces expressions with cheaper versions

int x=3;
int y=5;  $\Rightarrow$ ret 8;  (constant propagation)
ret x+y;

$\mapsto_v 8$

$((\lambda X. (\lambda Y. (X+Y)) 5) 3)$

$\blacksquare \rightarrow 8$

$(\lambda X. \blacksquare) \Rightarrow$ 'fun

$(\blacksquare\ (\lambda X.X)) \rightarrow$ stuck

$((\lambda X.(X+1))\ \blacksquare) \Rightarrow 9$

$(10 + \blacksquare) \rightarrow 18$

Imagine the language P, with the feature "print"

$M ::= \ldots \mid \text{print } M \mid \hat{M}$    P: $(\text{print } M) \rightarrow \hat{M}$

$\hat{M} := {}'X \mid (\text{list } '\lambda\ X\ \hat{M}) \mid (\hat{M}\ \hat{M})$

$b \mid (\text{print } \hat{M}) \mid (\text{list } \vec{o}^n\ \hat{M} \ldots)$

$C ::= \ldots \mid \text{print } C$

$(\forall m. \forall N.\ M \simeq_p N \Rightarrow M = N)$

$(\forall m. \forall N.\ M \neq N \Rightarrow M \not\simeq_p N)$    $C = (\text{print } \blacksquare)$

Imagine the language X, which is like assembly

$M ::= \text{movq } R, A \mid \text{addq } R, A \mid M ; M \mid \text{ret}$

$R ::= \text{rax} \mid \text{rbx}$    $A ::= A \mid R$    $\hat{A}(n_a, n_b, \text{rbx}) = n_b$

$P := (n, n, m)$    $\hat{A}(n_a, n_b, n) = n$

$C ::= \blacksquare \mid C; M \mid M; C$    $\hat{A}(n_a, n_b, \text{rax}) = n_a$

$m_a:\ (n_a, n_b, \text{movq rax } A) \rightarrow (\hat{A}(n_a, n_b, A), nb, \text{ret})$

$a_a:\ (n_a, n_b, \text{addq rax } A) \rightarrow (n_a + \hat{A}(n_a, n_b, A), nb, \text{ret})$

$s:\ (n_a, n_b, M_1 ; M_2) \rightarrow (n_a', n_b', M_2)$
iff $(n_a, n_b, M_1) \rightarrow\!\!\!\rightarrow (n_a', n_b', \text{ret})$

$\text{evalx}(M) = (n_a, n_b)$ iff $(0, 0, m) \rightarrow\!\!\!\rightarrow_v (n_a, n_b, \text{ret})$