

$eval_v : M \rightarrow answers = A$

$$eval_v(M) = \begin{cases} b & \text{if } M =_v b \\ 'fun & \text{if } M =_v \lambda X.N \end{cases}$$

answers = \bigcup (constant, function, or variable)
 $\begin{matrix} b & (\lambda X.M) & x \\ \downarrow & \downarrow & \downarrow \\ 'function & & X \in FV(M) \end{matrix}$

$A = B \cup \{'fun\}$

Is $eval_v$ total or partial?

partial : $\begin{cases} M =_v X \\ M =_v (b \ M) \\ =_v (o^n \ b \ \dots \ (\lambda X.N) \ b' \ \dots) \end{cases}$
 "stuck" (code for "error")

never finish = "diverge" (infinite loop)

$$\Omega = ((\lambda x. (x \ x)) (\lambda y. (y \ y))) \quad M \ B_v \ M$$

$$((\lambda y. (y \ y)) (\lambda y. (y \ y)))$$

Y = the Y combinator (or fixed-point operator)

$$Y_v = (\lambda f. (\lambda x. ((\lambda g. (f (\lambda x. ((g \ g) \ x)))) (\lambda g. (f (\lambda x. ((g \ g) \ x))) \ x))))$$

Theorem: If $K = \lambda Z. \lambda X. L$ then $(K (Y_v K)) =_v (Y_v K)$

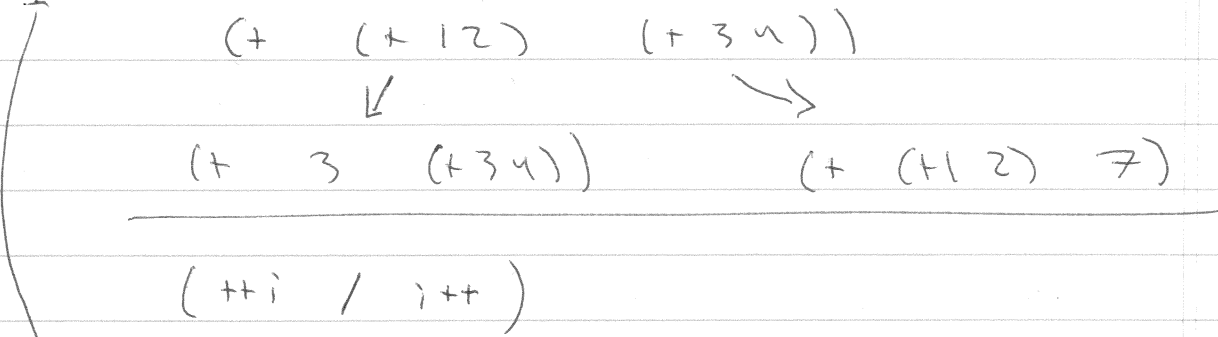
~~$(\lambda Z. \lambda X. L)$~~

$$\begin{aligned} Y_v K &\Rightarrow_v \dots \downarrow [f \leftarrow K] = V \\ Y_v K &\Rightarrow_v V \rightarrow_v \lambda X. (L \ K \ V) \ x \\ &= \lambda X. (([\lambda Z. \lambda X. L] \ V) \ x) \\ &\rightarrow_v \lambda X. L [Z \leftarrow V] [X \leftarrow x] \\ &= \lambda X. L [Z \leftarrow V] \end{aligned}$$

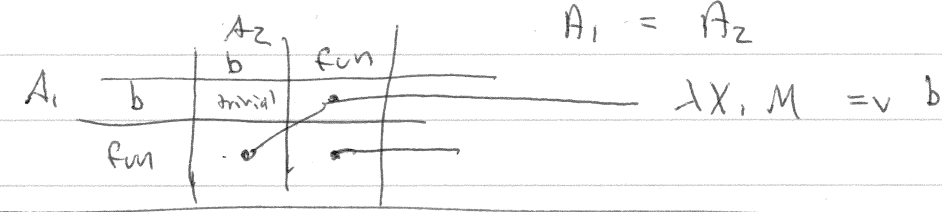
4-4 $\mathbb{R} M =_v V, \forall X, N,$
 $((\lambda X, N) M) =_v N [X \leftarrow M] \quad (\beta)$

Justifies intinaty (pg 51)

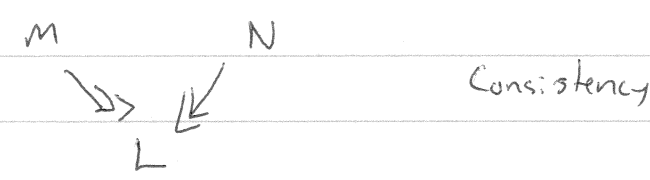
5-1 eval_v is a partial function



$\forall M, A_1, A_2, \text{eval}_v(M) = A_1 \wedge \text{eval}_v(M) = A_2 \rightarrow$

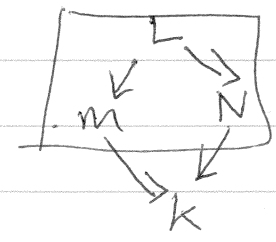


If $M =_v N, \exists L, M \rightarrow_v L$ and $N \rightarrow_v L$.



Church-Rosser

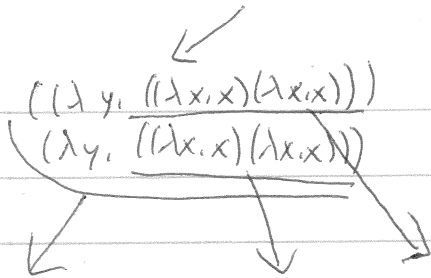
If $L \rightarrow_v M$ and $L \rightarrow_v N, \exists k,$
 $M \rightarrow_v k$ and $N \rightarrow_v k$



B - C.R. was true for \rightarrow_r

S-2

$$((\lambda x. (x x)) (\lambda y. ((\lambda x. x) (\lambda x. x))))$$



$$((\lambda x. (x x)) (\lambda y. (\lambda x. x)))$$

$$((\lambda y. (\lambda x. x)) (\lambda y. (\lambda x. x)))$$

$$M \hookrightarrow_v M$$

$$A : (0^n b_1 \dots b_n) \hookrightarrow_v \delta(0^n, b_1, \dots, b_n)$$

$$B'_v : ((\lambda X. M) u) \hookrightarrow_v M'[x \leftarrow v]$$

if $M \hookrightarrow_v M'$ and $u \hookrightarrow_v v'$

$$(M N) \hookrightarrow_v (M' N') \quad \text{if } M \hookrightarrow_v M' \text{ and } N \hookrightarrow_v N'$$

$$(\lambda X. M) \hookrightarrow_v (\lambda X. M')$$

$$(0^n M_1 \dots M_n) \hookrightarrow_v (0^n M'_1 \dots M'_n)$$

Substitution Lemma:

If $M \hookrightarrow_v M'$ and $N \hookrightarrow_v N'$, then

$$M[x \leftarrow N] \hookrightarrow_v M'[x \leftarrow N']$$

$$M = (+ (+ 3 10) (x \frac{1}{2} 5)) \quad M' = (+ 13 (x \frac{1}{2} 5))$$

$$N = (\lambda y. (+ ~~2~~ (+ 2 4) y))$$

$$N' = (\lambda y. (+ 6 y))$$

$$\left[(+ (+ 3 10) ((\lambda y. (+ (+ 2 4) y)) 5)) \right]$$

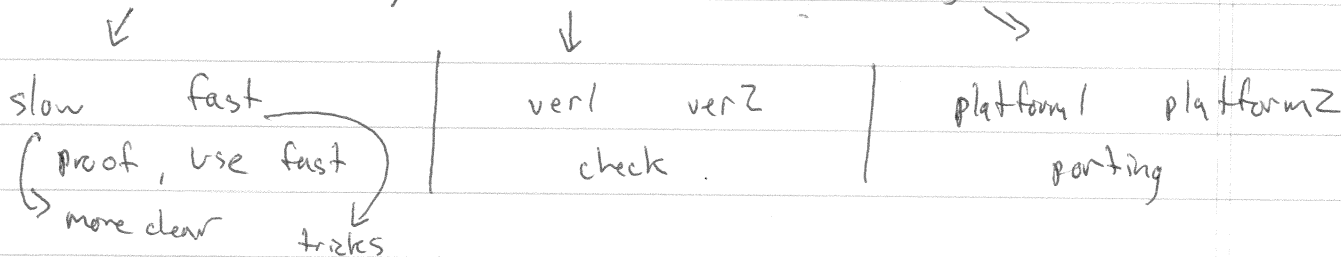
$$\left[(+ 13 ((\lambda y. (+ 6 y)) 5)) \right]$$

S-3 Observational Equivalence

What does \equiv mean?

\hookrightarrow sym (refl + trans (compatible ($v = \Delta \cup \beta v$)))

Want: do these programs do the same thing?



$$(x \text{ } 3 \text{ } y) \equiv z$$

FR ($\lambda x. \dots$ factorial recursive)
 \equiv IF ($\lambda x. \dots$ iterative factorial...)
 X no!

$$\Omega = ((\lambda x. (x x)) (\lambda x. (x x)))$$

$$(\lambda x. \Omega) \text{ diverges} \not\equiv (\lambda x. (\Omega \Omega)) \text{ diverges}$$

\Rightarrow turn the program M into a math function

\hookrightarrow denotational semantics (Scott Domain Theory)

$$\textcircled{1} \exists x. (FR x) \neq (IF x) \quad \text{isort vs qsort}$$

$$\textcircled{2} (a \rightarrow \text{bool}) (\lambda a) \rightarrow (\lambda a)$$

$$\exists \text{comp}, l. \text{isort}(\text{comp}, l) = \text{qsort}(\text{comp}, l)$$

F and G

$$\textcircled{3} X \rightarrow (Y \rightarrow Z) \quad Fx = Gx$$

$$\exists x, \exists y. ((Fx) y) \neq ((Gx) y)$$

Context is a program with a HOLE

$$\textcircled{1} C = ([] x) \quad ([] z)$$

$$\textcircled{2} C = (([] x) y) \quad (([] 10) 1)$$

5-4

$$\begin{aligned}
 C &= [] \\
 | &(\lambda x. C) \\
 | &(C M) \\
 | &(\lambda x. C) \\
 | &(0^n M \dots C M \dots)
 \end{aligned}$$

(There's nothing for X or b)

$C[N]$ to mean "fill the hole in C with N"

$$[] [N] = N$$

$$(\lambda x. C) [N] = (\lambda x. C[N]) \quad // N \text{ can mention } x$$

$$(C M) [N] = (C[N] M)$$

$$(M C) [N] = (M C[N])$$

$$(0^n M \dots C M \dots) [N] = (0^n M \dots C[N] M \dots)$$

$$C = \lambda x. []$$

$$N = x$$

$$C[N] = \lambda x. x$$

$$\neq \lambda x. y \quad \neq \lambda y. x$$

$$M \rightarrow_{\omega} N \text{ iff}$$

$$\exists C. M = C[M'] \text{ and } N = C[N'] \text{ and } M' \nu N'$$

$$\rightarrow_{\omega} = \rightarrow_{\nu}$$

$$M = (x (+34)) \quad N = (x 7)$$

$$C = (x []) \quad M' = \text{~~(+34)~~} \quad N' = 7$$

$M \simeq_{\nu} N$ (M is observationally equivalent to N)

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

wants: $\forall M. M \simeq_{\nu} M$

if $L \simeq_{\nu} M$ and $M \simeq_{\nu} N$, $L \simeq_{\nu} N$ (equivalence)

$$L \simeq_{\nu} M \rightarrow M \simeq_{\nu} L$$

if $M \simeq_{\nu} N$, $\forall C. C[M] \simeq_{\nu} C[N]$

5-5 Soundness + Incompleteness

If $M \equiv_v N$, then $M \preceq_v N$, (soundness)

But, $M \preceq_v N \not\Rightarrow \text{that } M \equiv_v N$ (incompleteness)

↓

$$(\lambda x. \Omega) \preceq_v (\lambda x. (\lambda r. r)) \quad \equiv_v \not\equiv \preceq_v$$

but

$$(\lambda x. \Omega) \not\equiv_v (\lambda x. (\lambda r. r))$$

$$(\lambda r. r)$$

↓

$$(\text{loop}()) (\lambda r. r) \preceq_v \text{loop}()$$

$$\eta \text{ (eta)} : (\lambda x. Mx) \equiv M$$

$$(\lambda x. \Omega x) \quad \Omega$$

$$C = []$$

↓
fun

↓
diverges