

Substitution:

$$M[N/x]$$

$$\text{Subst}: \Sigma_{\text{exp}}^M \rightarrow \Sigma_{\text{exp}}^M \rightarrow \Sigma_{\text{exp}}^x \rightarrow \Sigma_{\text{exp}}$$

Definition 1:

$$x[s/x] = s$$

$$y[s/x] = y \quad \text{if } x \neq y$$

$$(\lambda y. t_1)[s/x] = \lambda y. t_1[s/x]$$

$$(t_1 t_2)[s/x] = t_1[s/x] t_2[s/x]$$

$$(\lambda x. x) \left[\frac{y}{x} \right] \not\rightarrow (\lambda x. x) \quad \text{Counter Example}$$

Boas \Rightarrow Constant.

Definition 2:

$$x[s/x] = s$$

$$y[s/x] = y \quad \text{if } x \neq y$$

$$(\lambda y. t_1)[s/x] = \lambda y. t_1[s/x] \quad \text{if } x \neq y$$

$$(t_1 t_2)[s/x] = t_1[s/x] t_2[s/x]$$

$$\left(\lambda \underbrace{x}_{\text{Constant}}. y \right) \left[\frac{s}{\underbrace{x}_{\text{Constant}} y} \right] \rightarrow \left(\lambda \underbrace{x}_{\text{Constant}}. x \right) \quad \left. \vphantom{\left(\lambda \underbrace{x}_{\text{Constant}}. x \right)} \right\} \text{Capturing Substitution.}$$

"Capture - Avoiding" Substitution:

Definition 3

$$x[s/x] = s$$

$$y \in [s/2] \Rightarrow y \text{ if } x \neq y$$

$$\rightarrow (x, y, t_1) [s/r] = x, y, t_1 [s/h] \quad \text{if } x \neq y \wedge y \notin \text{FV}(s)$$

$$(t_1, t_2) [s/2] = t_1 [s/2] \quad t_2 [s/2]$$

$(\lambda x. xy) \left[\frac{x}{y} \right] = \frac{\lambda x. (\omega/2) xy \left[\frac{x}{y} \right]}{\omega \text{ is free}} \rightarrow \lambda x. \left[\frac{x}{y} \right] (\omega/2) (xy) \rightarrow \lambda x. \omega \lambda x. x$

Definition 4 :

$$\underline{x} [s/x] = \underline{s}$$

$$\underline{y} [s/x] = \underline{y} \quad \text{if } \underline{x} \neq \underline{y}$$

$$\underline{(\lambda y. t_1)} [s/x] = \lambda y. \underline{t_1} [s/x] \quad \text{if } \underline{x} \neq \underline{y} \wedge y \notin \text{FV}(s)$$

$$\underline{(\lambda y. t_1)} [s/x] = \lambda \omega. \underline{t_1} \left[\underline{\left[\frac{\omega}{y} \right]} \left[\frac{s}{x} \right] \right] \quad \text{if } \underline{x} \neq \underline{y} \wedge y \in \text{FV}(s)$$

$$\underline{(t_1 t_2)} [s/x] = \underline{t_1} [s/x] \underline{t_2} [s/x]$$

$\omega \text{ is free}$

$\text{FV} : \underline{\text{Expr}} \rightarrow \text{String set}$

(Var)

(Abs)

(App)

$$\left[\begin{array}{l} \omega \notin \text{FV}(s) \cup \text{FV}(t_1) \\ \omega \in \{x, y\} \end{array} \right]$$

$$\text{FV}(x) = \{x\}$$

$$\text{FV}(y) = \{y\}$$

→ Reduction Rules

$$e ::= \underline{x} \mid (\underline{\lambda x. e}) \mid \underline{e_1} \underline{e_2}$$

1) variable

x

Already in
(Normal form)

2) ^{Lambda} Abstraction

$$\frac{e \rightarrow e'}{(\lambda x. e) \rightarrow (\lambda x. e')} \quad [E\text{-Abs}]$$

3) Application:

- Congruence rules

$$\checkmark \frac{e_1 \longrightarrow e_1'}{e_1 e_2 \longrightarrow e_1' e_2} \quad [E-App-1]$$

$$\checkmark \frac{e_2 \longrightarrow e_2'}{e_1 e_2 \longrightarrow e_1 e_2'} \quad [E-App-2]$$

- Reduction rule

$$\frac{}{(\lambda x. e_1) e_2 \longrightarrow \underbrace{[e_2/x] e_1}_{= e_1 [e_2/x]}} \quad [E-App-3]$$

Full - beta reduction.

2/17

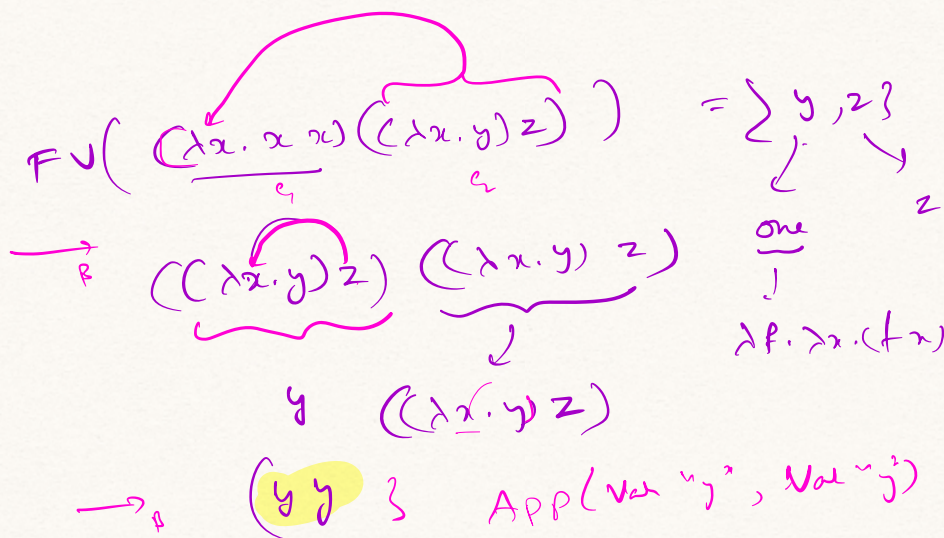
CSCI 7000 - 11

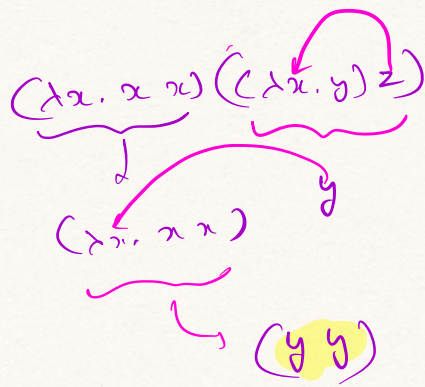
Today's lecture:

- * Recap of dynamic semantics
- * Full-beta reduction (\rightarrow_β)
- * Non-deterministic evaluation
- * Example: $(\lambda x. x x) ((\lambda x. y) z)$
- * β -normal form (no β redexes)
- * Multi-step reduction
 - * Reflexive transitive closure of (\rightarrow_β)
- * Church-Rosser Theorem
- * Lack of normal form: Ω combinator
 - \hookrightarrow contradiction?
- * Reduction strategy: Normal order vs
 - call-by-name/need vs
 - call-by-value

Example: $(\lambda x. x) ((\lambda x. x) (\lambda x. (\lambda x. x) z))$

CBN Example: $(\lambda x. (x x) (x z)) ((\lambda x. x) a)$

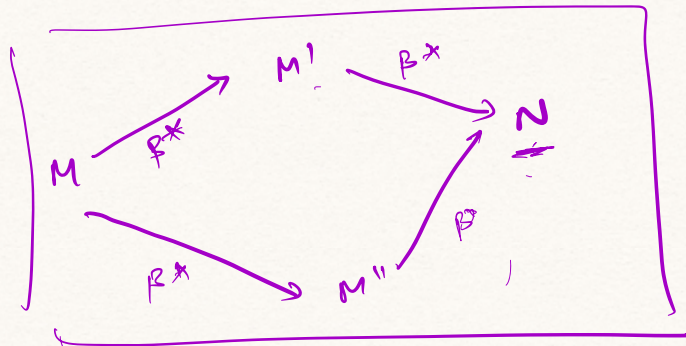




$$(\lambda x. e_1) e_2$$

$$c[e_1/x]$$

Church Rosser:



Ω
Combinator

Normal order reduction

"Left-most outer-most reduce first"

$$(\lambda x. x) ((\lambda x. x) (\lambda z. (\lambda x. x) z))$$

$$\rightarrow_{\beta N} \underline{Id} (\underline{Id} (\lambda z. \underline{Id} z))$$

$$= \underline{Id} (\lambda z. \underline{Id} z)$$

$(\lambda z. \text{Id } z)$

$(\lambda z. z)$

Id

Call-by-Value

→ Evaluate arguments first

→ Do not evaluate inside lambda Abstraction.

$[\text{fun } () \rightarrow \text{raise } (\text{EnvId Arg} \dots)]$

Eager evaluation

Strict evaluation

$(\lambda z. x) ((\lambda x. x) (\lambda z. (\lambda x. x) z))$

$\text{Id } (\text{Id } (\lambda z. \text{Id } z))$

→ BV

Expensive

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

factor 1000000

$\text{Id } (\lambda z. \text{Id } z)$

↓

$(\lambda z. \text{Id } z)$

2

→ Call-by-Name Semantics

Lazy

non-strict

→ Evaluate top-level β-reducers first

→ Do not evaluate inside lambda Abstraction.

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

$$\underbrace{\lambda}_{c_1} \left(\underbrace{\lambda z. \lambda z}_{c_2} z \right)$$

→ BNA

$$\lambda z. \lambda z z$$

$$(\lambda z. \lambda z z)$$

$$(\lambda x. y) (\underbrace{\text{factorial } 10^3}) \quad \text{Avoid this case!}$$

$$(\lambda x. \underbrace{x+x}) (\underbrace{\text{factorial } 10^3}_{10^{99}})$$

call-by-need)

$$: (\underbrace{(\lambda x. (x y) (x z))}_f) (\underbrace{(\lambda x. x) a}_\text{arg})$$

$$((\lambda x. x) a \ y) \quad ((\lambda x. x) a \ z)$$

* α -renaming

$$e_1 =_\alpha e_2$$

* β -reduction

$$e_1 \rightarrow_\beta e_2$$

$$e_1 =_\beta e_2$$

* η -reduction
 η -eliminator

$$e_1 =_\eta e_2$$

$(\text{fun } x \rightarrow \text{foo } x)$

$(\lambda x. \text{foo } x)$
 $=_n \text{foo}$

$[1; 2; 3]$ 3ℓ

↓

$["1"; "2"; "3"]$

List.map $(\text{fun } x \rightarrow \text{string_of_int } x)$ ℓ

List.map string_of_int ℓ

Church encoding

* numbers ✓

* Booleans:

: true

: false

$(\lambda x. \lambda y. x)$ ^{true}

$(\lambda x. \lambda y. y)$ ^{false}

ite = $(\lambda b. \lambda x. \lambda y. b\ x\ y)$ ^{false}

(ifc) $\text{true } e_1 \ e_2 \longrightarrow e_1$

ifc $\text{false } e_1 \ e_2 \longrightarrow e_2$

$\rightarrow FN \neq \lambda N$

$YF = F(YF)$

$\rightarrow Y = \lambda f. (\lambda x. f(x x)) (\lambda x. f(x x))$

$YF = (\lambda x. F(x x)) (\lambda x. F(x x))$

$F(\lambda x. F(x x)) (\lambda x. F(x x))$

$\rightarrow \text{fact} : \lambda f. \lambda n. \text{if } n=0 \text{ then } 1 \text{ else } n * f(n-1)$

$(Y \text{ fact})$ must compute fact at $\underbrace{(Y \text{ fact})}_{n=n=n}$

$\Rightarrow \text{fact} = n!$

$FN \neq \lambda N$

$F(YF) = (YF)$

CBV: $fp : Z = \lambda f. (\lambda x. f(\lambda y. x x y)) (\lambda x. f(\lambda y. x x y))$

$\rightarrow \text{List} : \text{nil} = \lambda c. \lambda n. n$

$\text{cons} : \lambda h. \lambda t. \lambda c. \lambda n. c \ h \ t$

$\text{cons } ch \ t = \lambda c. \lambda n. c \ h \ t$

$\text{cons } (h \ \text{nil}) = \lambda c. \lambda n. c \ h \ (\lambda x. \lambda y. n)$

$= \lambda c. \lambda n. c \ h \ (h \ (\lambda x. \lambda y. n))$

plus m₂ m₁

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

$\text{cur} = \lambda b \ \lambda c. \ \underline{\text{itc}} \ \text{true} \ e_1 \ e_2 \rightarrow \text{true} \ e_1 \ e_2$
 \downarrow
 e_1

$(\lambda x. \lambda y. \text{cur}) \ \underline{\text{cur} \ x \ y} \ \text{fr}$

$(\text{plus} \ m_2 \ m_1) \ (\text{cur} \ m_1 \ m_2)$

$b \ \text{true} \ c$

$b \ \text{false} \ \text{true}$