CSCI 5535

$e ::= x$ — Variable

$\mid \lambda x.\, e$ — Abstraction — Syntax

fn x => e

$\mid e\ e$ — Application

Dynamic Semantics

$\boxed{e \longrightarrow e'}$

$$\frac{e \longrightarrow e'}{\lambda x.\, e \longrightarrow \lambda x.\, e'}$$

[Possible Rule]
not popular

Congruence rules

✓
$$\frac{e_1 \longrightarrow e_1'}{e_1\ e_2 \longrightarrow e_1'\ e_2}$$

×
$$\frac{e_2 \longrightarrow e_2'}{e_1\ e_2 \longrightarrow e_1\ e_2'}$$

[possible]

✓
$$\frac{e_2 \longrightarrow e_2' \quad \mathrm{Value}(v_1)}{v_1\ e_2 \longrightarrow v_1\ e_2'}$$

$(\lambda x.\ \lambda y.\ \underline{x}\ \underline{y})$

$(\lambda x.\ \lambda y.\ z\ \underline{x}\ \underline{y})$ free variable

Closed term = a term with no free variables

$\lambda f.\ \lambda x.\ \underline{f}\ \underline{x}$

Variables not values

only values in our UTLC are functions:

value $\underline{v} ::= \lambda x.\ e$

reduction rule

$$\frac{\text{value}(v)}{(\lambda x.\ e)\ v \longrightarrow [v/x]\ e}$$ Beta Reduction

$$\frac{}{(\lambda x.\ e) \longrightarrow \lambda y.\ (y/_{x})\ e}$$ α-Renaming

η — Expansion

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

Church Encoding

$\lambda f. \lambda x.\ x$ zero

$\boxed{\lambda f. \lambda x.\ f\ x}$ 1

$\lambda f. \lambda x.\ f\ (f\ x)$ 2

$\lambda f. \lambda x.\ f\ (f\ (f\ x))$ 3

Succ n = $\lambda f. \lambda x\ f\ (n\ f\ x)$

Succ two =

$\lambda g. \lambda x.\ g\ (g\ (x))$

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

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

$n = \lambda f. \lambda x. f^n x$

$plus = \lambda \underline{m}. \lambda \underline{n}. \quad f^m . f^n . x$

$= f^{m+n}(x)$

$\lambda f. \lambda x. m f \underbrace{(n f x)}$

$= f^m(f^n(x))$

$= f^{m+n}(x)$

Boolean

$True = \lambda a. \lambda b. a$

$False = \lambda a. \lambda b. b$

$Not\ p = \lambda p. \lambda a. \lambda \underline{b}. p\ \underline{b}\ a$

$OR = \lambda p_1. \lambda p_2. \lambda \underline{a}. \lambda b. \ p_1\ a\ (p_2\ a\ b)$

Recursion

$\underbrace{(\lambda x. x\ x)}_{v_1} \ \underbrace{(\lambda x. x\ x)}_{v_1}$ — Ω Combinator

$\rightarrow (\lambda x. x\ x)(\lambda x. x\ x)$

$Y =$

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

$Y \ g = (\lambda x. \ g \ (x \ x)) \ (\lambda x. \ g \ (x \ x))$

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

$= g \ (Y \ g)$

$= g \ (g \ (Y \ g))$

$= g \ (g \ (g \ (Y \ g)))$

Y - Combinator

Z