Today λ calculus evaluation

β-reduction

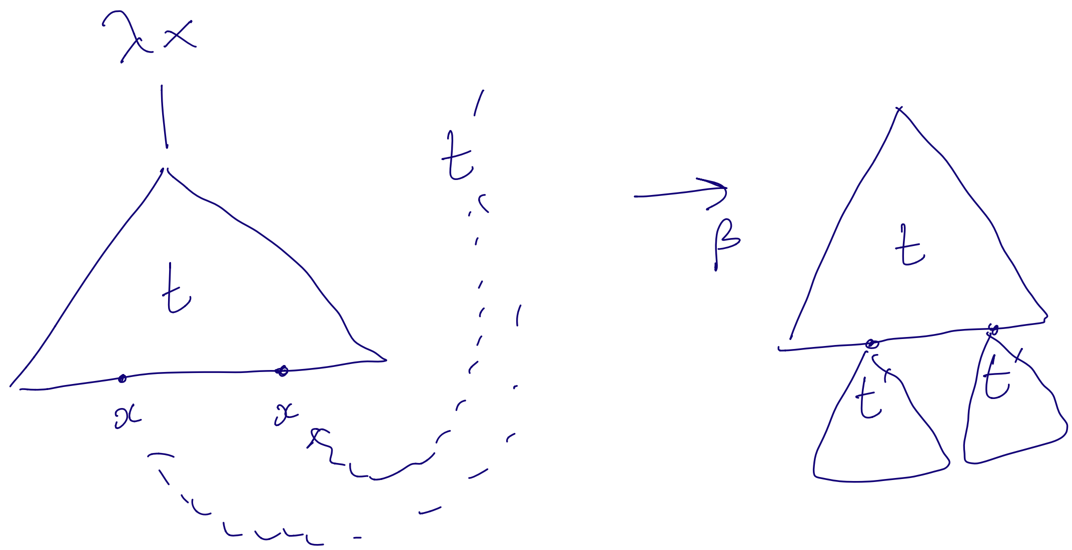
$$\underbrace{(\lambda x . x)}_{\text{identity combinator}} \; y \longrightarrow_{\beta} y$$

$$((\lambda x . x \; (\lambda x . x)) \; (u \; r)$$

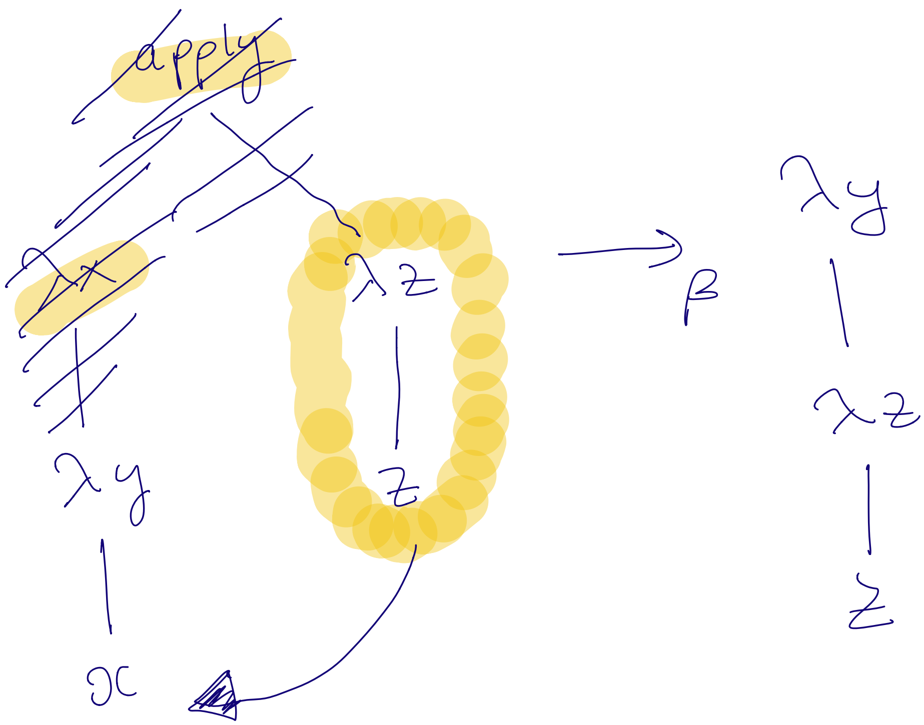
$$\longrightarrow_{\beta}$$

$$(u \; r) \; (\lambda x . x)$$

$$\underbrace{(\lambda x \,.\, t)}\ t'$$



$$(\lambda x.(\lambda y.x))(\lambda z.z) \xrightarrow{\beta} \lambda y.(\lambda z.z)$$



Reduction strategies

$id = \underbrace{(\lambda x . x)}$

① Full beta reduction

$$\underbrace{id}\ \underbrace{(\underbrace{id}\ \underbrace{(\lambda z . \underbrace{id}\ \underbrace{z})})}$$

$$\xrightarrow{\beta} id\ (\underbrace{id}\ \underbrace{(\lambda z . z)})$$

$$\xrightarrow{\beta} id\ (\lambda z . z)$$

$$\xrightarrow{\beta} \lambda z . z \quad \checkmark$$

normal form

② Normal order reduction

NOR left-most, outermost redex first

$$\underline{id}\ (\underline{id\ (\lambda z.\ id\ z)})$$

$$\xrightarrow[\beta]{}\ \underline{id}\ (\underline{\lambda z.\ id\ z})$$

$$\xrightarrow[\beta]{}\ \lambda z.\ \underline{id}\ \underline{z}$$

$$\xrightarrow[\beta]{}\ \lambda z.\ z$$

<u>Theorem</u>: <u>if t has a normal form t'</u> following NOR, $t \xrightarrow[NOR]{} t'$

③ Call by name (CBN)

no reductions inside abstractions

$\underline{id}\ \underbrace{(\underline{id}\ (\underline{\lambda z.\ id\ z}))}$

$\rightarrow\ \underbrace{id}\ (\lambda z.\ \underbrace{id\ z})$

$\rightarrow\ \boxed{\lambda z.\ id\ z}$

ALGOL 60

Haskell

call by need

lazy evaluation

④ Call by value

only outermost redexes reduced

redex reduced when its RHS has been reduced

$\underline{id}\ (\underbrace{id}\ \underbrace{(\lambda z.\ id\ z)})$

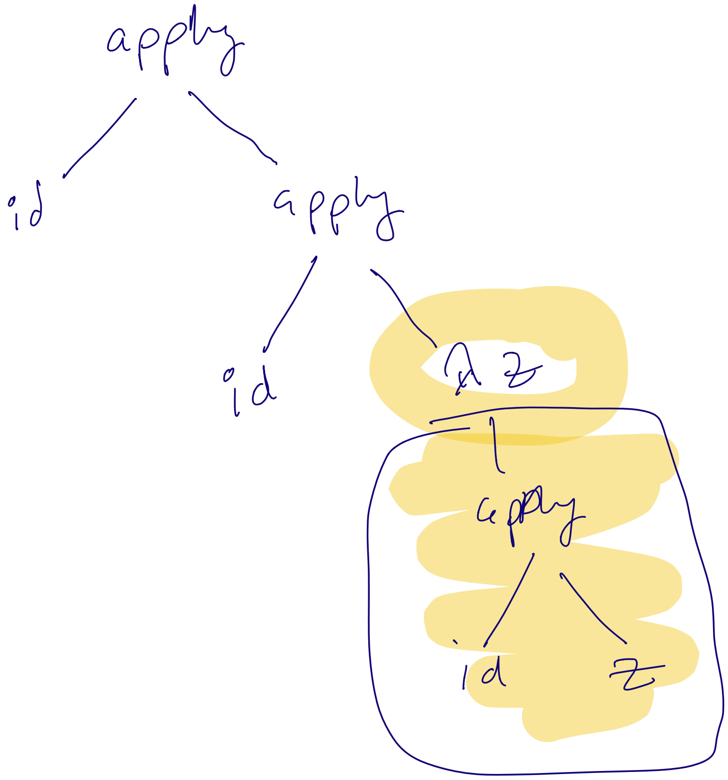
$\rightarrow_\beta\ id\ \underbrace{(\lambda z.\ id\ z)}$

$\rightarrow_\beta\ \lambda z.\ id\ z$

if CBV converges,

then CBN converges

opposite not true


apply
id
apply
id
λz
apply
id
z

$$(\lambda x.\, x\, x)\,(\lambda x.\, x\, x)$$

$$\to_\beta (\lambda x.\, x\, x)\,(\lambda x.\, x\, x)$$

$$\to_\beta \ldots$$

$$(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)$$

$$\to_\beta ((\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x))\,(\lambda x.\, x\, x\, x)$$

$$\to_\beta (\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)$$

$$W = (\lambda x . x \; x \; x)$$

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

$$I = \lambda x . x$$

$$F \; (W \; W) \; I$$

$$\rightarrow F \; (W W W) \; I$$

$$\rightarrow F \; (W W W W) \; I$$

$$\rightarrow \ldots$$

doesn't terminate

Better strategy

$$(F \; (W \; W)) \; I$$

$$\rightarrow (\lambda y . y) \; I$$

$$\rightarrow \lambda x . x \quad \text{done}$$

terminates

CBN

β-reduction name clashes

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

$$(\lambda x.\ t)\ t' \longrightarrow [x \mapsto t']\ t$$

α-renaming

$$(\lambda x.\ f\ x\ ((\lambda d.\ y\ \ell))\ v)\ z$$

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

$$\longrightarrow (\lambda y.w)\,z$$

$$\longrightarrow w$$

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

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

y got captured !

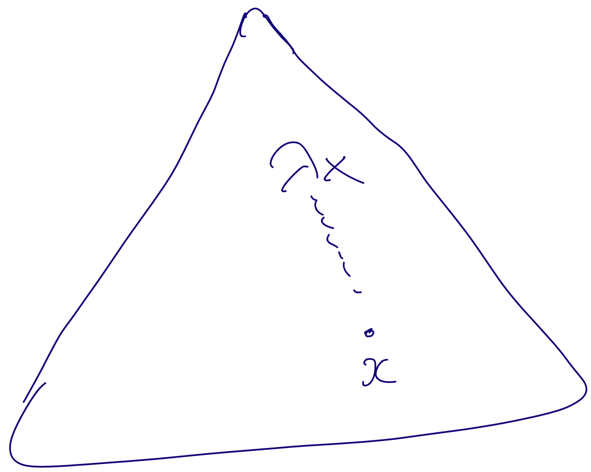
scoping

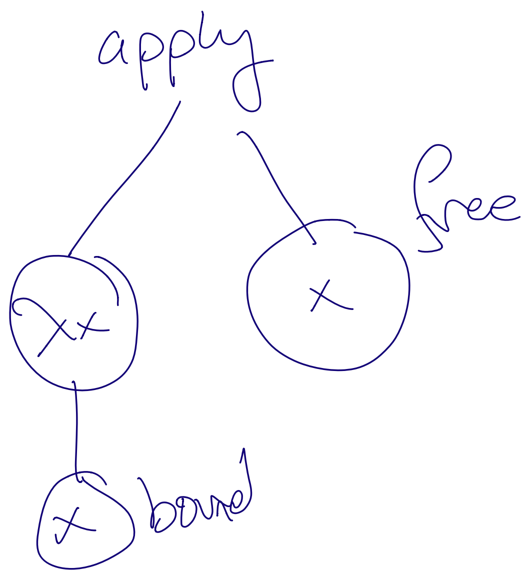
Bound variables

Free variables

$$\lambda x . \underbrace{\qquad\qquad\qquad}_{\text{scope of } x}$$

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





Structural definition

① In x (ie a variable)

x is free

there are no bound variables

② $\lambda x . M$ ← arbitrary term

- Every x in M is bound
- for all $y \neq x$, if y is free in M, it is free in $\lambda x . M$

③ $M \; N$, similar to case $M \; N$

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

$$FV(\underline{\lambda x.\, M}) = FV(M) - \{x\}$$

$$FV(\underline{M \;\; N}) = FV(M) \cup FV(N)$$

$$t = x \mid \lambda x.\, t \mid t\ t$$