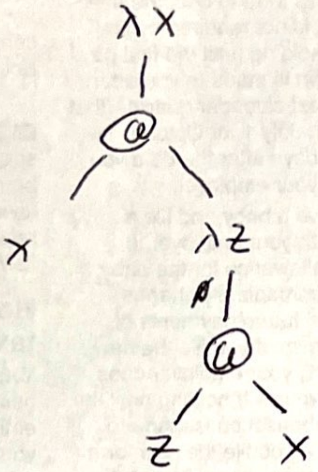
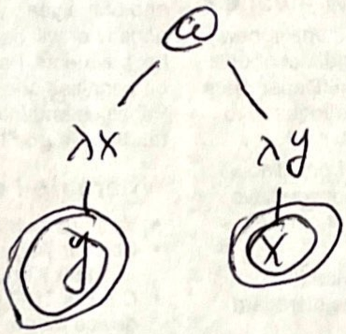
lambda calculus

1. $\lambda x.x \ \lambda z.\cancel{x}\ x$

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

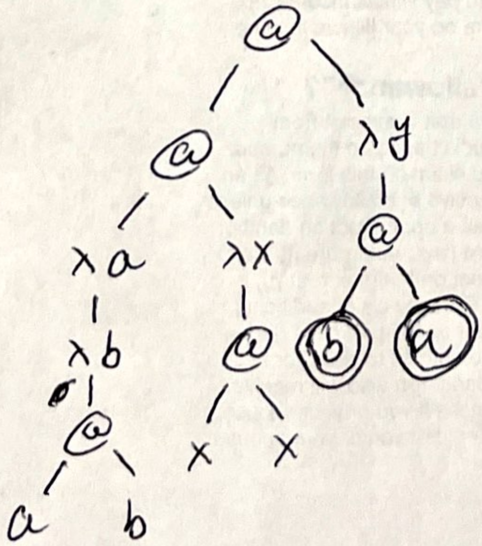


2. $(\lambda x.y) \ (\lambda y.x)$



free variables: $\{x, y\}$

3. $(\lambda a.\lambda b.ab)(\lambda x.x \ x)(\lambda y. \ b \ a)$



free variables: {a and b} under lambda y.

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

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

Since $(\lambda z.(y z))$ can not do β-reduction, we can not evaluate it further, therefore

$\equiv_{\alpha} (((\lambda x.\lambda v.(v x)\ w)(\lambda z.(y z))$

$\equiv_{\beta} (\lambda v.(v\ w))(\lambda z.(y z))$

$\equiv_{\beta}\ (\lambda z.(y\ z))\ w$

$\equiv_{\beta}\ y\ w.$

(a) Since we only have one order, so it is both applicative & normal order, therefore, it can find normal form.

(b) the steps are above.

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

→ Applicative order:

$\equiv_{\beta} (\lambda x.y)(((\lambda y.\ y y y)(\lambda y.\ y y y))(\lambda y.\ y y y))$

$\equiv_{\alpha} (\lambda x.y)(((\lambda w.\ w w w)(\lambda y.\ y y y))(\lambda y.\ y y y))$

We note that $(\lambda w.\ w w w)(\lambda y.\ y y y)$ can generate infinite loops, therefore,

(a): It can not find normal form by applicative order

→ Normal order:

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

$\equiv_{\beta}\ y$

(b): Only normal order can reach normal form.

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

Since $(\lambda w. w\ w)$ can not evaluate further. therefore

$\equiv_\beta (\lambda w. w\ w)\ y$

$\equiv_\beta y\ y$

(a). It can find normal form by applicative order.

(b) steps are above.

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

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

$\equiv_\beta (\lambda y. ((\lambda w. w)\ y\ (\lambda w. w))) (\lambda z. z\ z)$

$\equiv_\beta (\lambda w. w)(\lambda z. z\ z)(\lambda w. w)$

$\equiv_\beta (\lambda z. z\ z)(\lambda w. w)$

$\equiv_\beta (\lambda w. w)(\lambda w. w)$

$\equiv_\beta \lambda w. w$

(a). It can find normal form by applicative order.

(b) steps are above.

8: $DEC \equiv \lambda x.(x\ False)$

we note that $(InC\ n) \equiv (\lambda n.\lambda x.(x\ False)\ n)\ n$

$$\equiv_\alpha (\lambda v.\lambda x.(x\ False)\ v)\ n$$

$$\equiv_\beta (\lambda x.(x\ False)\ n)$$

we can do an assumption of $DEC \equiv \lambda a.a\,b$, which helps us can reduct $(InC\ n)$ further. so

$$DEC\ (InC\ n) \equiv (\lambda a.\ a b)(\lambda x.(x\ False)\ n)$$

$$\equiv_\beta (\lambda x.(x\ False)\ n)\ b$$

$$\equiv_\beta (b\ False)\ n$$

Now, let $b \equiv False$, which is $\lambda x.\lambda y.y$.

$$\equiv ((\lambda x.(\lambda y.y))\ False)\ n$$

$$\equiv_\beta (\lambda y.y)\ n$$

$$\equiv_\beta n.$$

Finally, $DEC \equiv \lambda a.(a\ False) \equiv_\alpha \lambda x.(x\ False)$.

$DEC\ (DEC\ 2) \equiv DEC(DEC\ (INC\ 1)))$, we can do β-reduction on $INC\ 1$.

$$\equiv_\beta DEC(\ DEC\ ((\lambda x.(x\ False)\ 1))$$

$$\equiv DEC((\lambda x.(x\ False))(\lambda x.(x\ False)\ 1))$$

$$\equiv_\beta DEC\ ((\lambda x.(x\ False)\ 1)\ False)$$

$$\equiv_\beta DEC((False\ \ False)\ 1)$$

$$\equiv_\beta DEC((\lambda y.y)\ 1)$$

$$\equiv_\beta (DEC\ 1)$$

Similarly, since $1 \equiv INC\ 0$, $(DEC\ 1) \equiv (DEC\ (INC\ 0))$, we can obtain 0 use the same steps.

9. $IS\_ZERO \equiv \lambda x.(x\ TRUE)$

$(IS\_ZERO\ 0) \equiv (\lambda x.(x\ TRUE))\ 0$

$\equiv_\beta\ 0\ TRUE$

$\equiv\ \lambda x.x\ TRUE$

$\equiv_\beta\ TRUE.$

$(IS\_ZERO\ 1) \equiv_\beta\ 1\ TRUE$

$\equiv_\beta (\lambda x.(x\ False)\ 0)\ TRUE$

$\equiv_\beta (TRUE\ False)\ 0$

$\equiv_\beta (\lambda x.\lambda y.x\ \ False)\ 0$

$\equiv_\beta (\lambda y.\ False)\ 0$

$\equiv\ False$

Similarly. for any number, we can get format as $(TRUE\ False)\ n \equiv False$.