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

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

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

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

$= (\lambda y.(\lambda a.a\ (y\ y))\ b)$

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

can be done in another sequence of β-reduction

$= (\lambda y.(\lambda a.a\ (y\ y))\ b)$

$= (\lambda a.a\ (b\ b)) = (b\ b)$

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

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

$= ((\lambda y.(y\ x)\ x_1)\ z)$ (Avoiding same names) ?

$= ((x_1\ x)\ z)$

page 2

1 (d). $(\lambda g.(g\ \lambda x.\lambda y\ x)\ ((\underbrace{\lambda a.\lambda b.\lambda h.((h\ a)\ b)}\ \underbrace{z_1})\ z_2))$

$\therefore ((\lambda a.\lambda b.\lambda h.((h\ a)\ b)\ z_1)\ z_2)$

$= (\lambda b.\lambda h.((h\ z_1)\ b)\ z_2)$

$= \lambda h.((h\ z_1)\ z_2)$

Proceeding further

$(\underbrace{\lambda g.(g\ \lambda x.\lambda y.x)}\ \underbrace{\lambda h.((h\ z_1)\ z_2)})$

$= (\underbrace{\lambda h.((h\ z_1)\ z_2)}\ \underbrace{\lambda x.\lambda y.x})$

$= ((\underbrace{\lambda x.\lambda y.x}\ \underbrace{z_1})\ z_2)$

$= (\underbrace{\lambda y.z_1}\ \underbrace{z_2}) = z_1$

1 (e). $((\underbrace{\lambda t.\lambda y.(t\ y)}\ \underbrace{\lambda n.\lambda f.\lambda x.(f\ ((n\ f)\ x))}_{\text{Let's call this } P})\ \lambda g.\lambda z.(g\ (g\ z)))$

$= (\underbrace{\lambda y.(P\ y)}\ \underbrace{\lambda g.\lambda z.(g\ (g\ z))})$

$= (P\ \underbrace{\lambda g.\lambda z.(g\ (g\ z))}_{Q.})$

$= (P\ \ Q)$

Therefore, $(\underbrace{\lambda n.\lambda f.\lambda x.(f\ ((n\ f)\ x))}\ \underbrace{Q})$

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

We proceed with computation of $((Q\ f)\ x)$

i.e.,

$$((\underbrace{\lambda g.\lambda z.(g\ (g\ z))}\ f)\ x)$$

$$= (\underbrace{\lambda z.(f\ (f\ z))}\ x)$$

$$= (f\ (f\ x))$$

Therefore

$$(P\ Q) = \lambda f.\lambda x.(f\ (f\ (f\ x)))$$

2(a). Apply on true:

$$(\lambda x.((x\ \text{false})\ \text{true})\ \ \text{true})$$

$$= ((\text{true}\ \text{false})\ \text{true})$$

$$= ((\lambda x.\lambda y.x\ \ \text{false})\ \text{true}) = \text{false}$$

Apply on false you will get true.

Function : $\neg$. boolean operation

2(b) Apply on zero

$$(\lambda n.((n\ \lambda p.((p\ \text{false})\ \text{true}))\ \text{false})\ \ \text{zero})$$

$$= ((\text{zero}\ \lambda p.((p\ \text{false})\ \text{true}))\ \text{false})$$

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

Therefore, $((\underbrace{\lambda f.\lambda x.x}\ \underbrace{\lambda p.((p\ \text{false})\ \text{true}))}_{f}\ \underbrace{\text{false}}_{x})$

$$= \text{false}$$

page4

Apply on one, it would be

$$((\lambda f.\lambda x.(f\ x)\ \lambda p.((p\ false)\ true))\ false)$$

$$\equiv\ (\lambda p.((p\ false)\ true)\ false)$$

$$\equiv\ ((false\ false)\ true)$$

$$\equiv\ ((\lambda x.\lambda y.y\ false)\ true) = true.$$

Apply on two. That would ~~im~~ lead to applying $\lambda p\ ((p\ false)\ true)$ twice on false. We know from above, applying it one-time leads to true.

Applying $(\lambda p\ ((p\ false)\ true)\ true)$

$$=\ ((true\ false)\ true)$$

$$=\ ((\lambda y.\lambda x.y\ false)\ true)$$

$$=\ false$$

In other words, the result toggles from true to false as we increase the # of application of $\lambda p\ ((p\ false)\ true)$

zero-application → False

one application → True

two application → False

Function : isODD, condition for naturals

2(c) (m (mul n)) where m is a natural number implies m-applications of (mul n).

$\therefore$ $\underbrace{(\underbrace{m}\ \underbrace{(\text{mul } n)})}\ \underbrace{(\text{succ zero})})$ implies

m applications of (mul n) to one. $\boxed{\text{Pg } 5}$

In other words,

$$\underbrace{((\text{mul } n)((\text{mul } n) \ldots (\text{mul } n)}_{m\text{- times}} \text{ one}))\underbrace{\ldots)}_{m}$$

$$= \underbrace{(n \times (n \times \cdots\cdots (n}_{m\text{ - times}} \times 1))\ldots)$$

$$= n^m$$

2(d) Apply on some signed number $((\text{Pair } z_1)\ z_2)$

$$((\text{Pair } (\text{sec } ((\text{Pair } z_1)\ z_2)))\ (\text{fst } ((\text{Pair } z_1)\ z_2)))$$

$$= ((\text{Pair } z_2)\ z_1)$$

$\boxed{\text{page 6}}$

The ~~orig~~ input pair represents $z_1 - z_2$

output pair " $z_2 - z_1$

Function: negates a signed number

2(e) Outline: input $z_1 - z_2$ and $y_1 - y_2$

output $(z_1 + y_2) - (z_2 + y_1)$

Function: subtraction of signed numbers.