Assignment 5 - Functional Programming
Lambda Calculus

$\boxed{\text{Exercise 1}}$

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

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

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

$a\,a$

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

Alpha equivalence: $(\lambda z.z)(\lambda a.a\,a)(\lambda b.b\,y)$

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

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

$y\,y$

c) $(\lambda x.\lambda y.x\,y\,y)(\lambda a.a)b$

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

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

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

$b\,b$

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

Alpha equivalence ~~$(\lambda x.\lambda y.x\,yy)(\lambda a.a)b$~~

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

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

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

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

$y\,y$

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

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

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

$x\,x\,z$

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

Alpha equivalance: $(\lambda x.(\lambda a.(xa))y)z$

$(\lambda a.za)y$

zy

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

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

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

$\lambda y.y$

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

Alpha equivalance: $(((\lambda x.\lambda a.(xa))(\lambda y.y))w)$

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

$(\lambda y.y)w$

w

(i) $(\lambda x.y)((\lambda y.y\,yy)(\lambda x.xxx))$

~~$(\lambda x.y)((\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx))$~~

$(\lambda x.y)\underbrace{((\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx))}_{LHS}$

It will pass the whole LHS to the first function. It doesn't matter what I pass to it, it will always return y. So the result is y. I can keep beta reducing the LHS but it won't take me anywhere.

$\boxed{\text{Exercise 2}}$

(a) or false true = true

$or \equiv \lambda xy.xTy \equiv \lambda x.(\lambda y.(xTy))$

$false \equiv \lambda xy.y \equiv \lambda x.(\lambda y.y)$

$true \equiv \lambda xy.x \equiv \lambda x.(\lambda y.x)$

$(\lambda x.(\lambda y.(xTy)))(\lambda x.(\lambda y.y))(\lambda x(\lambda y.x)) \equiv$

$\equiv$ Alpha equivalence: $(\lambda x(\lambda a.(xTa))(\lambda x.(\lambda y.y))(\lambda x.(\lambda y.x)) \equiv$

$\equiv \lambda a((\lambda x.(\lambda y.y))Ta)(\lambda x.(\lambda y.x)) \equiv$

$\equiv (\lambda x.\lambda y.y))T(\lambda x.(\lambda y.x))) \equiv$

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

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

(b) $+ \; 2 \; 2 = 4$

Successor Function $\equiv \lambda wyx.y(wyx) \equiv \lambda w.(\lambda y.(\lambda x.y(wyx)))$
S

We have to apply the successor function to 2, 2 times to perform the addition.

$2 \equiv \lambda sz.s(s(z))$ → from the pdf.
$\equiv \lambda s.(\lambda z.(s(s(z))))$

So we have:

~~$\lambda(wyx).y(wyx)$~~

~~2S2Z~~

Applying S to 2 one time:

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

$\equiv \lambda y.(\lambda x.y((\lambda s.(\lambda z.(s(s(z)))) xy)))) \equiv$

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

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

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

$\equiv 3$.

Apply S to 3 now:

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

$\equiv \lambda y.(\lambda x.y((\lambda s.\lambda z.s(s(s(z))))yx)) \equiv$

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

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

$\equiv 4$

So we proved the expression.

(c) succ $2 = 3$

succ $\equiv \lambda w(\lambda y.(\lambda x.y(wyx)))$ *

$2 \equiv \lambda s.(\lambda z(s(s(z))))$

$succ\,2 \equiv (\lambda w(\lambda y.(\lambda x.y(wyx)))) (\lambda s.\lambda z(s(s(z)))) \equiv$

$\equiv \lambda y.(\lambda x.y((\lambda s.\lambda z(s(s(z))))yx) \equiv$

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

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

$\equiv 3$

The successor function being applied 2 times to 0.