$\boxed{5}$ $mult = \lambda m n . \lambda f x . m(nf)x$

It works because if you apply „succ" to a number it will do the „succ" the number of times:

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

succ

If we apply another number next it will change every „f" on itself, making a multiplication.

Why 1 is a unit?

1) $mult\ n\ 1 = \lambda f x . n(1 f)x =$
$= \lambda f x . n((\lambda f x . f x) f) x = \lambda f x . n(\lambda f x . f x)x$
$= \lambda f x . n\ x$

same for n

same as before

2) $mult\ 1\ n = \lambda f x .\ 1\ (n f) x = \lambda f x . (n f) x =$
$= \lambda f x . n\ x$

change f to f

$\boxed{1}$ $\lambda x. MN = S(\lambda x. M)(\lambda x. N)$

def S:

$S(\lambda x. M)(\lambda x. N) = (\lambda f g x. f x (g x))(\lambda x. M)$
$(\lambda x. N) = \lambda x. (\lambda x. M) x ((\lambda x. N) x) = \lambda x. MN$

[2] xor = λ x y . x (not y) y

since A xor B $\stackrel{def}{=}$ if A $\rightarrow$ not B
if not A $\rightarrow$ B

we can write it using λ:

X (not y) (y)

if x if not X

if it's correct, then its commutative
(because xor a b = xor b a)
Let's prove that it's correct:

xor true true = true (false)(true) = false ✓
xor true false = true (true)(false) = true ✓
xor false true = false (false)(true) = true ✓
xor false false = false (true)(false) = false ✓

$\Rightarrow$ its commutative

4) I didn't come up with this solution.

The factorial can be calculated using this helper function:

$F(n_1, n_2) = F(n_1+1, n_1 \cdot n_2)$

Then $F(F(\dots(F(1,1)))) = (n+1, n!)$

What's a (n_1, n_2)? It's $\lambda abg\ (..)\ (..)$

$n_1\ n_2$ projection

Then the factorial will be

$\lambda n.\, n F\, (\lambda g. g\ (1)\ (1))\ (\lambda ab.\, b)$

same trick as in mult

(1, 1) + projection

take the second value $\equiv$ false

(+1)

Then $F = \lambda p. p\, (\lambda ab.\, \lambda g\, (\lambda f x.\, f\,(a f x))\ (\lambda f\, a\, (b f))\ )$

a+b

Now we can put it together:

$$\text{factorial} = \lambda n.\, n\, (\lambda p.p\, (\lambda ab.\lambda g.g\, (\lambda fx.f(afx))\, (\lambda f.a(bf))))$$

$$(1,1)$$

$$(\lambda g.g\, (\lambda fx.fx)(\lambda fx.fx))$$

$$(\lambda ab.b) \leftarrow \text{false}$$

factorial 3 calculations.

For now I will do a bird's eye view of the calculations, since it's a pain to do it completly.

$$\text{fact } 3 = 3\ F\ (1,1)\ \text{false} =$$

$$= F(F(F(1,1)))\ \text{false} = F(F(2,1))\ \text{false} =$$

$$= F(3,2)\ \text{false} = F(4,6)\ \text{false} = 6$$

3 Let's make some helper functions:

$trans\ p = \lambda p.\lambda ab.\lambda g.(p\ false)(succ(p\ false))\}$

What it does? It takes pair and makes pair from the second element and second element +1:

$trans\ (1,1) = (1,2)$

Now we can do this trick:

$trans(trans(trans(0,0))) = tr(tr(0,1)) = tr(1,2) =$
$= tr(2,3)$

Now, if we apply it n times we can get the prev of n:

$prev = \lambda n.\ n(trans)(0,0)\ true$

do trans n times

take first

Now let's do pred. succ $n =$

$= (\lambda n.\ n(\text{trans})(0,0)\ \text{true})(\lambda n f x.\ f(n \cancel{f})) =$

$= (\lambda n.\ (f(n))(\text{trans})(0,0)\ \text{true}) =$

$= (n, f(n))\ \text{true} = n$

$\Rightarrow$ pred. succ $\equiv$ id