7-1/ Lambda Calculus — Alonzo Church 1930s

$e := v \mid x \mid e\ e$

$v := \lambda x.\, e$

E = hole
$\mid E\ e$
$\mid v\ E$

$E[(\lambda x.e)\ v] \mapsto E[e[x \leftarrow v]]$

J_3 ... if, bools, nums, +/-, multi-arity funs

$J_3 = ((\lambda\ (x\ y)\ (+\ x\ y))\ 2\ 3)$

$(+\ 2\ 3)$

Haskell B Curry

Currying

$(\lambda x.\lambda y.((+\ x)\ y))\ 2\ 3 = (\lambda y.((+\ 2)\ y))\ 3$

72/ what are bools _really_?

if True A B = A

if False A B = B

Object -oriented Programming

True := $\lambda t.\, \lambda f.\, t$

False := $\lambda t.\, \lambda f.\, f$

if := $\lambda c.\, \lambda t.\, \lambda f.\, c\ t\ f = \lambda c.\, c$

not T = F not F = T

not := $\lambda c.\, \lambda t.\, \lambda f.\, c\ f\ t$

7-3/ what is a number?

zero := doesn't do anything

one := does it once

two := do it twice

zero := $\lambda f.\, \lambda x.\; x$

one := $\lambda f.\, \lambda x.\; f\ x$

two := $\lambda f.\, \lambda x.\; f\ (f\ x)$

add1 := $\lambda n.\, \lambda f.\, \lambda x.\; f\ (n\ f\ x)$

add1 one

$= \lambda f.\, \lambda x.\; f\ (\text{one}\ f\ x)$

$= \lambda f.\, \lambda x.\; f\ (f\ x)$

= two

7-4/ add := $\lambda n.\ \lambda m.\ \lambda f.\ \lambda x.$

$n\ f\ (m\ f\ x)$

Zero? := $\lambda n.\ n\ (\lambda x.\ \text{False})\ \text{True}$

mult := $\lambda n.\ \lambda m.\ \lambda f.\ \lambda x.$

$n\ (m\ f)\ x$

$(m\ f)\ \underbrace{(m\ f)\ x}\ \ f^3(f^3\ x) = f^6\ x$

7-5/ pair

fst (pair A B) = A

snd (pair A B) = B

pair := λl. λr. λs. s l r

fst := λp. p True

snd := λp. p False

subl := λn.

fst (n (λp. pair (snd p) (addl (snd p)))

(pair zero zero))

7-6/ make-factorial :=

λfac. λn.

((if (zero? n))

(λx. (add1 zero)))

(λx. ((mult n) (fac (sub1 n)))))

zero

rfac = make-factorial rfac

x = F x

7-7)

$$Z := \lambda f.\ ((\lambda x.\ f\ (\lambda v.\ (x\ x)\ v))\ (\lambda x.\ f\ (\lambda v.\ (x\ x)\ v)))$$

$$Z\ F = ((\lambda x.\ F\ (\lambda v.\ (x\ x)\ v))\ (\lambda x.\ F\ (\lambda v.\ (x\ x)\ v)))$$

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

$$= F\ (\lambda v.\ (Z\ F)\ v)$$

$$= F\ (Z\ F)$$

$$fac := Z\ \text{make-factorial}$$