3-1

The lambda calculus — Alonzo Church

$M, N, L ::= X$ — variable reference

$\mid (\lambda X. M)$ — abstraction

$\mid (M\ N)$ — application

$X ::=$ some set

· programs = answers

$((\lambda X. X)\ N) \rightarrow N$

$f(x) = x + 5$
$f(10) = 10 + 5$
$= 15$

$((\lambda X. (X\ X))\ N) \rightarrow (N\ N)$

$(((\lambda X. (\lambda Y. X))\ N)\ M) \rightarrow ((\lambda Y. N)\ M) \rightarrow N$

Haskell B. Curry

curry — represents n-arity funs as n-separete unary functions

α

$\beta \quad ((\lambda X. M)\ N) \mapsto M[X \leftarrow N]$

η

$M[X \leftarrow N]$ means "M where all Xs (reference to X) are replaced with N"

$(\lambda X. X)[X \leftarrow (M\ N)] = \lambda (M\ N). (M\ N)$ X wrong — not a var

$\lambda X. (M\ N)$ X wrong

$((\lambda X. (\lambda X. X))\ (M\ N))$ $\quad \lambda \underline{(M\ N)}. X$ X wrong — not a var

$\lambda X. X$

$(\lambda Y. X)[X \leftarrow Y] = (\lambda Y. Y)$ X

$= (\lambda Z. Y)$ (strange that Y turns to Z?)

~~[illegible]~~

$((\lambda X. (\lambda Y. X))\ Y)$

3-2

```
int f (int x,) { return x + 5*y;}
           int y
int main () {

   f( 5 , x);          ~~5+5×5~~
}
```

```
int f (int x) { return x+y; }
int main () {
  int y = 8;
  f (5);
}
```

$M[X \leftarrow N] \;:\; M \times X \times N \longmapsto N$

$X_1\,[X_1 \leftarrow N] = N$

$X_2\,[X_1 \leftarrow N] = X_2 \quad \text{if } X_1 \neq X_2$

$(M_1\, M_2)\,[X_1 \leftarrow N] = (M_1[X_1 \leftarrow N]\;\; M_2[X_1 \leftarrow N])$

$(\lambda X_1 . M)\,[X_1 \leftarrow N] = (\lambda X_1 . M)$

$(\lambda X_2 . M)\,[X_1 \leftarrow N] = (\lambda X_3 .\, M[X_2 \leftarrow X_3]\,[X_1 \leftarrow N])$

$\quad \text{if } X_1 \neq X_2, \;\; X_3 \notin FV(N) \text{ or } FV(M) - \{X_2\}$

$FV : M \rightarrow P(X)$ — free variables

$FV(X) = \{X\}$ "capture-avoiding substitution"

$FV(M\,N) = FV(M) \cup FV(N)$

$FV(\lambda X . M) = FV(M) - \{X\}$

$\beta \quad ((\lambda X . M)\, N) \rightarrow M[X \leftarrow N]$

$\alpha \quad (\lambda X_1 . M) \longmapsto (\lambda X_2 .\, M[X_1 \leftarrow X_2]) \quad \text{if } X_2 \notin FV(M)$

$\eta \quad (\lambda X . (M\, X)) \rightarrow M \quad \text{if } X \notin FV(M)$

$n = \beta \cup \alpha \cup \eta$

3-3

Booleans

$\text{true} := \lambda X. \lambda Y. X$ $\quad$ $\text{if } C\ T\ F := ((C\ T)\ F)$

$\text{false} := \lambda X. \lambda Y. Y$

$\text{true}\ M\ N \mapsto M$

$(\lambda X. \lambda Y. X)\ M\ N \mapsto (\lambda Y. M)\ N \rightarrow M$

$\text{false}\ M\ N \rightarrow N$

Pairs : fst , snd , pair

$\text{fst}\ (\text{pair}\ M\ N) \rightarrow M$

$\text{snd}\ (\text{pair}\ M\ N) \rightarrow N$

$\text{pair} := \lambda X. \lambda Y. \lambda Z. (Z\ X)\ Y$

$\text{fst} := \lambda P. P\ \text{true}$

$\text{snd} := \lambda P. P\ \text{false}$

Numbers

$0 = \lambda F. \lambda Z. Z$

$1 = \lambda F. \lambda Z. F\ Z$

$2 = \lambda F. \lambda Z. F(F Z)$

$\text{add}1 = \lambda N. \lambda F. \lambda Z. N\ F (F Z)$

$\text{plus} = \lambda N. \lambda M. \lambda F. \lambda Z.$
$\quad N\ F\ (M\ F\ Z)$

$(\lambda X. (X\ X))\ Y$

$\rightarrow (Y\ Y)$

$\Omega = (\lambda X. (X\ X))\ (\lambda X. (X\ X)) \rightarrow ((\lambda X. (X X))\ (\lambda X. (X\ X)))$

$\Omega \rightarrow \Omega$

$(\lambda X. F\ (X\ X))\ (\lambda X. F(X\ X)) = ?$ $\quad$ Y-combinator

$\rightarrow F\ ((\lambda X. F(X X))\ (\lambda X. F(X\ X)))$

$\rightarrow F(F(\ ?\ ))$

$\rightarrow F(F(F\ \ ?))$ $\quad$ $F := \lambda N.$ if $\underline{\quad}$ zero N

$(Y\ F) = F\ (Y\ F)$