

The lambda calculus — Alonzo Church

- $M, N, L ::= X$ — variable reference
- $| (\lambda X. M)$ — abstraction
- $| (M N)$ — application
- $X ::= \text{some set}$

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 fns as n-separate unary functions

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

η

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

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

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

$$\lambda X. X$$

$$(\lambda Y. X)[X \leftarrow Y] = (\lambda Y. Y) \quad \text{X}$$

$$= (\lambda Z. Y) \quad \text{(strange that Y turns to Z?)}$$

~~(((\lambda X. (\lambda Y. X)) Y) Y)~~

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

3-2/

```

int f (int x) { return x + 5; }
int main() {

```

```

    f ( 5 , x );

```

}

```

int f (int x) { return x + y; }

```

```

int main() {

```

```

    int y = 8;

```

```

    f ( 5 );

```

}

$$M [X \leftarrow N] : M \times X \cdot N \mapsto N$$

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

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

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

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

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

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

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

$$FV(X) = \{X\}$$

"capture-avoiding"

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

"substitution"

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

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

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

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

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

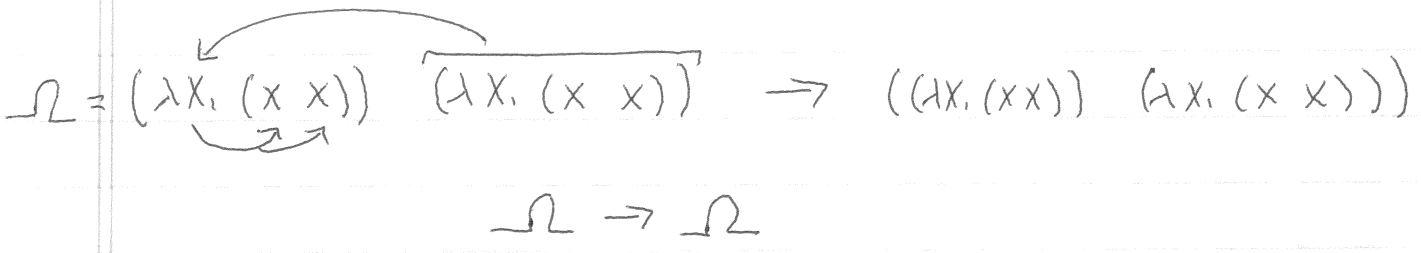
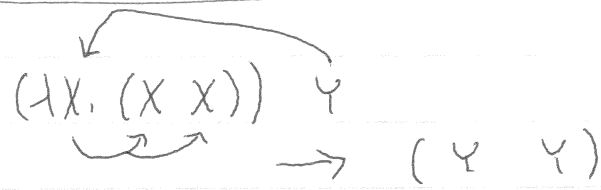
Booleans

$true := \lambda X. \lambda Y. X$ if C T F := ((C T) F)
 $false := \lambda X. \lambda Y. Y$

$true M N \mapsto M$
 $(\lambda X. \lambda Y. X) M N \mapsto (\lambda Y. M) N \mapsto M$
 $false M N \mapsto N$

Pairs : $fst, snd, pair$
 $fst (pair M N) \mapsto M$
 $snd (pair M N) \mapsto N$
 $pair := \lambda X. \lambda Y. \lambda Z. (Z X) Y$
 $fst := \lambda P. P true$
 $snd := \lambda P. P false$

Numbers
 $0 = \lambda F. \lambda Z. Z$
 $1 = \lambda F. \lambda Z. F Z$
 $2 = \lambda F. \lambda Z. F(F Z)$
 $add 1 = \lambda N. \lambda F. \lambda Z. N F(F Z)$
 $plus = \lambda N. \lambda M. \lambda F. \lambda Z. N F (M F Z)$



$(\lambda X. F (X X)) (\lambda X. F (X X)) = ?$ Y-combinator
 $\rightarrow F ((\lambda X. F (X X)) (\lambda X. F (X X)))$
 $\rightarrow F (F (?))$
 $\rightarrow F (F (F ?))$ $F := \lambda N. if \lfloor zero N$

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

