

## Programming constructs in $\lambda$ calc.

$\lambda x. M$

$((\lambda x. \lambda y. M) a) b \quad f(a, b)$

"Currying"

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

function specialization

### Church Boolean

$\left. \begin{array}{l} \text{tru} = \lambda t. \lambda f. t \\ \text{fls} = \lambda t. \lambda f. f \end{array} \right\}$

$\text{NOT} = \lambda x. (x \text{ fls}) \text{ tru}$

$\uparrow \quad \uparrow \quad \uparrow$   
 $\text{①} \quad \text{②}$

tru returns ①

fls returns ②

conditional

if (b) then v else w

$$\text{cond} = \lambda l. \lambda m. \lambda n. \underbrace{l \ m \ n}$$

$$\begin{aligned} & \xrightarrow{\text{cond} \ \text{tru} \ v \ w} (\lambda m. \lambda n. \text{tru} \ m \ n) \ v \ w \\ & \rightarrow \rightarrow \text{tru} \ v \ w \\ & \rightarrow v \\ & \rightarrow w \end{aligned}$$

$$\text{AND} = \lambda p. \lambda q. (p \ q) \ \text{fls}$$

$(p \ q) \ p$

$$\text{OR} = \lambda p. \lambda q. (p \ \text{tru}) \ q$$

$(p \ p) \ q$

$$\text{pair} = \lambda f. \lambda s. \lambda b. b \ f \ s$$

$\uparrow$   
getter

$$\text{fst} = \lambda p. p \ \text{tru}$$

$$\text{snd} = \lambda p. p \ \text{fls}$$

ex  
(v, w)

$$\text{pair} \ v \ w \rightarrow \lambda b. b \ v \ w$$

$$\begin{aligned} \text{fst} (\text{pair} \ v \ w) & \rightarrow (\lambda p. p \ \text{tru}) \ (\lambda b. b \ v \ w) \\ & \rightarrow (\lambda b. b \ v \ w) \ \text{tru} \end{aligned}$$

$\rightarrow \text{tr} \quad \vee \quad \omega$   
 $\rightarrow \vee$

### Church numerals

$$c_0 = \lambda s. \lambda z. z \quad \xrightarrow{\text{zero}}$$

$$c_1 = \lambda s. \lambda z. s \ z \quad \xrightarrow{\text{successor}}$$

$$c_2 = \lambda s. \lambda z. s(s \ z)$$

$$c_3 = \lambda s. \lambda z. s(s(s \ z))$$

0000  
 0001  
 0010

$$f1s = \lambda x. \lambda y. y$$

$$\lambda s. \lambda z. z$$

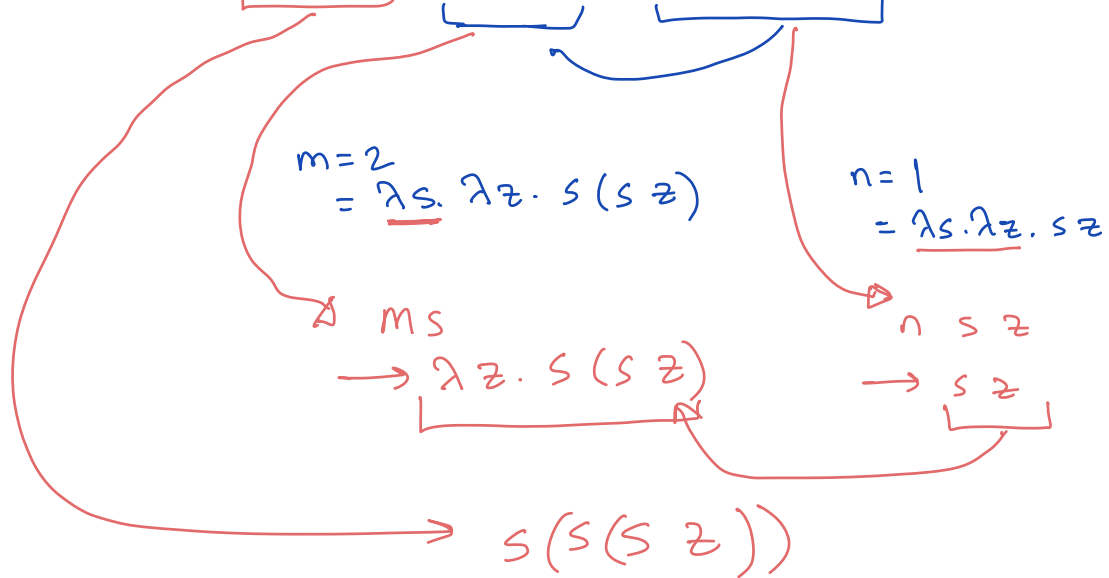
$$\cancel{\lambda s. \lambda z.} s(s \ z)$$

$$inc = \lambda n. \lambda s. \lambda z. s(n \ s \ z)$$

rebuild  $\lambda s. \lambda z.$   
 demolish  $\lambda s. \lambda z.$   
 adds an additional  $s$  in  $n$

$inc \ (\lambda s'. \lambda z'. z')$   
 $\rightarrow \lambda s. \lambda z. s \ ((\cancel{\lambda s'. \lambda z'.} z') \ s \ z)$   
 $\rightarrow \lambda s. \lambda z. s \ z$

$$\text{plus} = \lambda m. \lambda n. \lambda s. \lambda z. (m \ s) \ (n \ s \ z)$$

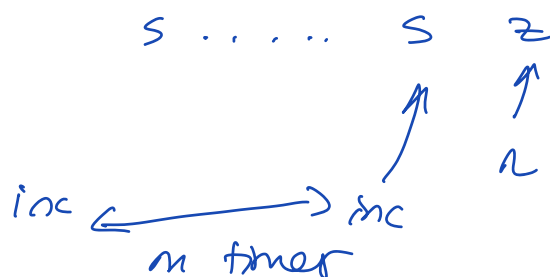


$$\text{times} = \lambda m. \lambda n. m \ (\text{plus } n) \ c_0$$

$$s(s(s \dots \dots z))$$

$$\text{plus2} = \lambda m. \lambda n. m \ \text{inc} \ n$$

$$\text{inc}(\text{inc}(\text{inc} \dots \dots n)))$$



$$\text{isZero} = \lambda n. n \underbrace{(\text{AND fls})}_{\substack{\lambda x. \text{fls} \\ \uparrow}} \text{tr.}$$


---

$$c_0 = \lambda s. \lambda z. z$$

$$c_0 \underbrace{(\lambda x. x)}_{\uparrow} \text{tr}$$

$$c_1 = \lambda s. \lambda z. s \ z$$

$$c_1 \underbrace{(\lambda x. x)}_{\uparrow} \text{tr}$$

$$(\text{AND fls}) \text{tr} \rightarrow \text{fls}$$

$$\underbrace{(\text{AND fls}) \underbrace{((\text{AND fls}) \text{tr})}_{\text{fls}}}_{\text{fls}}$$

Exponentiation

$$M^n$$

$$\text{exp} = \lambda m. \lambda n. \underbrace{n \quad m}_{\text{curried}}$$

de Bruijn  
indices

OCaml