

THE λ -CALCULUS!

ALSO:
OFFICE HOURS
2-3 IN 413 (ANDREW)
3-4 IN 411A (ADRIAN)

A PROGRAMMING LANGUAGE WITH JUST:

- VARIABLES
- FUNCTIONS ("ABSTRACTION")
- CALLS ("APPLICATION")

(IT'S HARD TO GET TINIER THAN THAT. (BUT IT'S POSSIBLE...))

SYNTAX

$e ::= x$ (VARIABLE)
| $\lambda x.e$ (ABSTRACTION)
| $e_1 e_2$ (APPLICATION)

"APPLIED" λ -CALCULUS

| n
| "string"
| $e_1 + e_2$

SEMANTICS (INFORMAL)

① $\lambda x.e$
↑ ↑ ↖

PYTHON:
`def f(x):` (EXCEPT NO NAME)
 `return e`

② e_1, e_2 ARGUMENT
 ↑
 FUNCTION

PARAMETER ONLY (USE x)

PYTHON:
 $e_1 (e_2)$

FOR EXAMPLE :

$$(\lambda x. x^2) 5 = 25$$

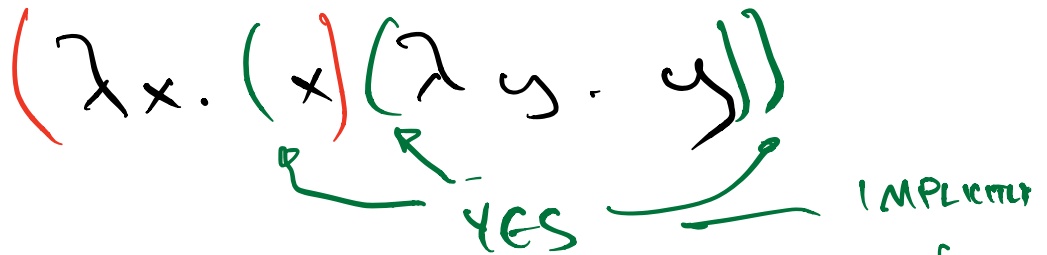
ABSTRACTION

APPLICATION

$\lambda x. x$ IDENTITY FUNCTION
 $\lambda x. (f (g x))$
 $(\lambda x. x) (\lambda y. y)$
 $\lambda y. \lambda x. x$
 IGNORE y AND RETURN

ABOUT PARENTHESES:

◦ λ BODIES ARE AS "BIG" AS POSSIBLE



◦ APPLICATIONS AS L-R

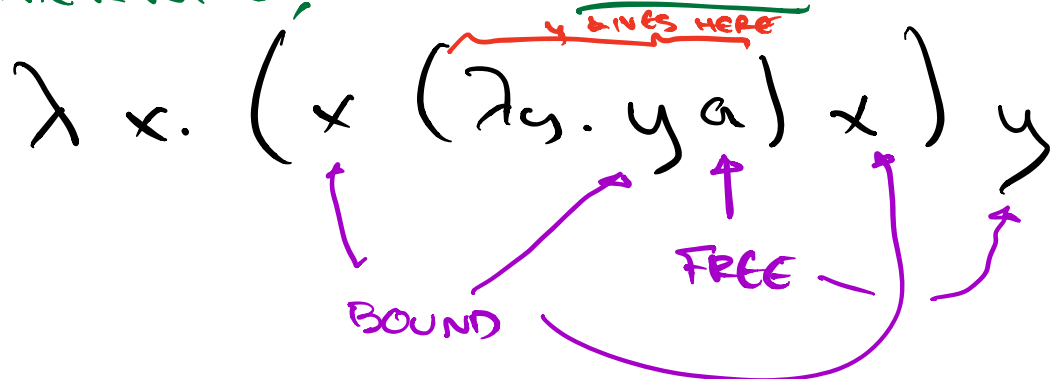


BINDING (AND α -EQUIVALENCE)

"x" IS BOUND IF \exists AN ENCLOSING

$\lambda x. \dots$

OTHERWISE, IT'S FREE.



CLOSED TERM : ALL VARIABLE OCCURRENCES ARE BOUND

$\lambda x. (\lambda x. x + x)$

$\lambda x. x$ $\lambda y. y$
↑ ↓
α-EQUIVALENT

α-RENAMING IS SAFE
CHANGING THE NAMES OF
BOUND VARIABLES

$\lambda x. \lambda y. x$

↓
 $\lambda y. \lambda y. y$

NOT
α-EQUIVALENT

HIGHER-ORDER FUNCTIONS

... TAKE AND RETURN FUNCTIONS

$\lambda f. f 42$

$\lambda v. \lambda f. f v$

"REVERSE APPLICATION"

ACTUAL SEMANTICS

$(\lambda x. e_1) e_2$
SHOULD REPLACE ^{FREE} x IN
 e_1 w/ e_2

NOTATION: $e_1 \{e_2/x\}$

$(\lambda x. e_1) e_2 \quad e_1 \{e_2/x\}$

\curvearrowright
 β -EQUIVALENT

β -REDUCTION IS
REWRITING w/ SUBSTITUTION

MULTIPLE WAYS TO β -REDUCE:

$(\lambda x. x+x) \quad ((\lambda y. y) 5)$
 $\swarrow \quad \searrow$

$(\lambda y. y) 5 + (\lambda y. y) 5$ $(\lambda x. x+x) 5$

CALL BY VALUE

FUNCTIONS ONLY CALLED
ON VALUES

$(\lambda x. e_1) e_2$

↑

MUST BE A VALUE
BEFORE β -REDUCTION

VALUE

- Not β -REDUCIBLE

- $\lambda x. e$

FORMAL SEMANTICS

$$\frac{e_1 \rightarrow e_1'}{e_1 e_2 \rightarrow e_1' e_2}$$

$$\frac{e \rightarrow e'}{\forall e \rightarrow \forall e'}$$

$$\beta \frac{}{(\lambda x. e) v \rightarrow e \{v/x\}}$$

$$((\lambda x. \lambda y. y x) (5+2)) (\lambda x. x+1)$$

$$\begin{aligned} & \rightarrow \underbrace{\hspace{10em}}_7 \\ & \rightarrow (\lambda y. y \ 7) \ (\lambda x. x+1) \\ & \rightarrow (\lambda x. x+1) \ 7 \\ & \rightarrow 7+1 \\ & \rightarrow 8 \end{aligned}$$

CALL BY NAME

APPLY A.S.A.P.

$$\frac{e_1 \rightarrow e_1'}{e_1 e_2 \rightarrow e_1' e_2}$$

$$\beta \frac{}{(\lambda x. e_1) e_2 \rightarrow e_1 \{e_2/x\}}$$

$$\begin{aligned}
& (\lambda x. \lambda y. y x) (S+2) (\lambda x. x+1) \\
\rightarrow & (\lambda y. y (S+2)) (\lambda x. x+1) \\
\rightarrow & (\lambda x. x+1) (S+2) \\
\rightarrow & (S+2) + 1 \\
\rightarrow & 7 + 1 \\
\rightarrow & 8
\end{aligned}$$