THE λ-CALCULUS!

ALSO:
OFFICE HOURS
2-3 IN 613 (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

$$
\begin{aligned}
e ::= \; & x && \text{(VARIABLE)} \\
\mid \; & \lambda x . e && \text{(ABSTRACTION)} \\
\mid \; & e_1 \; e_2 && \text{(APPLICATION)}
\end{aligned}
$$

"APPLIED" λ-CALCULUS

$$
\begin{aligned}
\mid \; & n \\
\mid \; & \text{"string"} \\
\mid \; & e_1 + e_2
\end{aligned}
$$

SEMANTICS (INFORMAL)

① $\lambda x . e$

BODY (USE x)

PARAMETER

PYTHON:

```
def f(x):
    return e
```

(EXCEPT NO NAME)

② $e_1 \; e_2$

e_1 ← FUNCTION; e_2 ← ARGUMENT

PYTHON:

$e_1(e_2)$

FOR EXAMPLE:

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

$(\lambda x.\ x^2)$ — ABSTRACTION

$(\lambda x.\ x^2)\ 5$ — 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 IDENTITY FUNCTION

ABOUT PARENTHESES:

- λ BODIES ARE AS "BIG" AS POSSIBLE

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

YES — IMPLICITLY

- APPLICATIONS GO L-R

$$(e_1\ e_2)\ e_3$$

YES

BINDING (AND α-EQUIVALENCE)

"x" IS **BOUND** IF ∃ AN ENCLOSING

$$\lambda x.\ \_\_\_\_$$

OTHERWISE, IT'S **FREE**.

$$\lambda x.\ (x\ (\lambda y.\ y\ a)\ x)\ y$$

y LIVES HERE

BOUND

FREE

CLOSED TERM: ALL VARIABLE OCCURRENCES ARE BOUND

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

$$\lambda x.\ x \qquad \lambda y.\ y$$

α - EQUIVALENT

α - RENAMING IS SAFELY 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: $e_1 \{ e_2 / x \}$

$$(\lambda x.\, e_1)\, e_2 \qquad \longleftrightarrow \qquad e_1 \{ e_2 / x \}$$

β - EQUIVALENT

β-REDUCTION IS REWRITING w/ SUBSTITUTION

MULTIPLE WAYS TO β-REDUCE:

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

↙ ↘

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

CALL BY VALUE

FUNCTIONS ONLY CALLED ON VALUES

$$(\lambda x.\, e_1)\, e_2$$

e_2 MUST BE A VALUE BEFORE β-REDUCTION

VALUE

- NOT β-REDUCIBLE
- $\lambda x.\, e$

FORMAL SEMANTICS

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

$$\frac{e \rightarrow e'}{v\, e \rightarrow v\, e'}$$

$$\beta \quad \frac{}{(\lambda x.e)\,v \to e\{v/x\}}$$

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

$\to$ 7

$\to$ $(\lambda y.\ y\ 7)\ (\lambda x.x+1)$

$\to$ $(\lambda x.\ x+1)\ 7$

$\to$ $7+1$

$\to$ 8

CALL BY NAME

APPLY A.S.A.P.

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

$$\beta \quad \frac{}{(\lambda x.e_1)\,e_2 \to e_1\{e_2/x\}}$$

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

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

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

$$\rightarrow\ (5+2)+1$$

$$\rightarrow\ 7+1$$

$$\rightarrow\ 8$$