

PL Practice, Applications, grab bag, etc.

Garbage Collection via environments & resource mgmt.

Environment-based interpreter

$$E ::= \{ x \mapsto v, \dots \}$$

$$e ::= \dots$$

$$v ::= \dots \mid \underbrace{\langle \text{Function } x \mapsto e, E \rangle}_{\text{closure}}$$

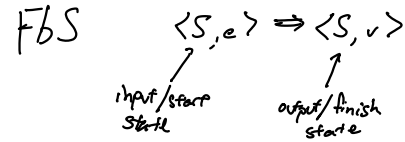
Relation $E \vdash e \Rightarrow v$

$$\frac{E \vdash e_1 \Rightarrow v_1 \quad E, x \mapsto v_1 \vdash e_2 \Rightarrow v_2}{E \vdash \text{Let } x = e_1 \text{ In } e_2 \Rightarrow v_2}$$

$$\frac{x \mapsto v \in E}{E \vdash x \Rightarrow v}$$

$$\frac{E \vdash e_1 \Rightarrow \langle \text{Function } x \mapsto e', E' \rangle \quad E \vdash e_2 \Rightarrow v_2 \quad E', x \mapsto v_2 \vdash e' \Rightarrow v}{E \vdash e_1 \quad e_2 \Rightarrow v}$$

$$\frac{\{ a \mapsto 1 \} \vdash \text{Fun } q \mapsto a \Rightarrow \dots}{\dots}$$



Let $f =$
 Let $a = 1$ In
 Function $q \mapsto a$
 In
 Let $a = 7$ In
 $f \quad a$

C op sems (sort of) *! +

$$M ::= \{ a \mapsto v, \dots \}$$

$$S ::= \{ x \mapsto v \}$$

$$v ::= \dots \mid a$$

$$s ::= e \mid \text{int } x; \mid \dots$$

$$S, M \vdash e \Rightarrow v, M$$

$$\frac{S, M \vdash \mathbb{Z} \Rightarrow \mathbb{Z}, M}{a \notin M}$$

$$S, M \vdash \text{malloc}(\text{sizeof}(\text{int})) \Rightarrow a, M \cup \{ a \mapsto \mathbb{Z} \}$$

$$S, M \vdash s \Rightarrow S, M$$

$$S, M \vdash \text{int } x; \Rightarrow S \cup \{ x \mapsto v \}, M$$

$$S, M \vdash e \Rightarrow v, M'$$

$$S, M \vdash \text{int } x = e; \Rightarrow S \cup \{ x \mapsto v \}, M'$$

$$\frac{S, M \vdash e \Rightarrow v, M'}{S, M \vdash e \Rightarrow S, M'}$$

(int *) 5

$$S, M \vdash \text{malloc}(\text{sizeof}(\text{int})) \Rightarrow a_0, M \cup \{ a_0 \mapsto v \}$$

$$S, M \vdash \text{int }^* a = \text{malloc}(\text{sizeof}(\text{int})); \Rightarrow S \cup \{ x \mapsto a_0 \}, M \cup \{ a_0 \mapsto v \}$$

$$\frac{\Gamma \vdash e : \tau'}{\Gamma \vdash (\tau) e : \tau}$$

Can we determine automatically when memory/cells/etc. are no longer used?

Garbage Collection?

FBS

$$\langle S, e \rangle \Rightarrow \langle S, v \rangle$$

CFBS

$$\langle S, e \rangle \Rightarrow \langle S, v \rangle$$

c does not appear in any value of S

$$c \mapsto v \in S \quad c \text{ does not appear in } e \quad \langle S_0 \setminus c, e \rangle \Rightarrow \langle S_1, v \rangle$$

$$\langle S_0, e \rangle \Rightarrow \langle S_1, v \rangle$$

Lambda Calculus

$$e ::= \lambda x. e \mid x \mid e e$$

$$\lambda x. e \Rightarrow \lambda x. e$$

$$\frac{e_1 \Rightarrow \lambda x. e' \quad e'[e_2/x] \Rightarrow e}{e_1 e_2 \Rightarrow e}$$

Alonzo Church

Church Encodings

Natural numbers: a function that takes another function and an argument and applies the function to the argument N times is an encoding of N

$$0 \equiv \lambda f. \lambda x. x \quad \text{Function } f \rightarrow \text{Function } x \rightarrow x$$

$$1 \equiv \lambda f. \lambda x. f x$$

$$2 \equiv \lambda f. \lambda x. f (f x) \quad \text{calls } f \text{ 3 times}$$

$$\lambda f. \lambda x. N (M f) x$$

$$\boxed{\lambda N. \lambda M. \lambda f. \lambda x. N (M f) x} \quad \text{Multiplication}$$

Booleans

$$\text{true} \equiv \lambda x. \lambda y. x$$

$$\text{false} \equiv \lambda x. \lambda y. y$$

$$\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \equiv e_1 (\lambda x. e_2) (\lambda x. e_3) \circ$$

$$e_1 \text{ and } e_2 \equiv \text{if } e_1 \text{ then } e_2 \text{ else false}$$

$$\text{not } e_1 \equiv \text{if } e_1 \text{ then false else true}$$

$$\equiv \lambda x. \lambda y. e_1 y x$$