PL Practice, Applications, grab bag, etc.

Garbage Collection via environments & resource mgmt.

Environment-based interpreter

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

$e ::= \dots$

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

Relation $E \vdash e \Rightarrow v$

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

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

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

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

input/start state

output/finish state

```
Let f =
    Let a=1 In
    Function q -> a
In
Let a = 7 In
f 2
```

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

C op sems (sort of)

* ! + ‡

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

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

$v ::= \dots \mid a$

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

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

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

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

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

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

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

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

(int *) 5

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

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

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

Garbage Collection?

FbS

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

GFbS

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

$$\frac{c \mapsto v \in S \qquad \begin{array}{c} c \text{ does not appear in any value of } S \\ c \text{ does not appear in } e \end{array} \qquad \langle S_0 \backslash c, e\rangle \Rightarrow \langle S_1, v\rangle}{\langle S_0, e\rangle \Rightarrow \langle S_1, v\rangle}$$

Lambda Calculus

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

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

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

$0 \equiv \lambda f.\lambda x.\, x$ Function f → Function x → x

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

$2 \equiv \lambda f.\lambda x.\, f\,(f\, x)$

$$\lambda f.\lambda x.\; N \; \overbrace{(M \;\; f)}^{\text{calls f 3 times}} \;\; x$$

$$\boxed{\lambda N.\lambda M.\;\lambda f.\lambda x.\; N\,(M \;\; f) \;\; x}$$ Multiplication

Booleans

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

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

if e_1 then e_2 else e_3 $\equiv e_1 \; (\lambda x.e_2) \; (\lambda x.e_3) \;\; 0$

e_1 and e_2 $\equiv$ if e_1 then e_2 else false

not e_1 $\equiv$ if e_1 then false else true

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