Lecture 11. Operational Equivalence

Logistics:

March	10th	Tue	Midterm Review
March	12th	Thu	Midterm
March	6th	Fri	Assignment 5

F^b $\quad e ::= i \mid b \mid x \mid \ldots$

$\quad v ::= i \mid b \mid \text{Fun } x \to e$

evaluation $\quad e \Rightarrow v$

equivalence $\quad e \cong e'$

$1 \cong 1$

$\text{Fun } x \to x \cong \text{Let Rec } f\ x = x \text{ In } f$

Correctness:

Suppose for a task, there is a program e^* that is ground truth. You are generating e. Then we say e is correct iff

$$e \cong e^*$$

Optimizations

Transformations $\quad T : e \to e$

$$e \cong T(e)$$

Operational Equivalence

$$e \cong e'$$

$$2+3 \cong 3+2 \qquad \text{True}$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$5 \cong 5$$

$$\boxed{a+b} \cong \boxed{b+a} \qquad \text{True}$$

Let $x = \boxed{a+b}$ In $x+1$

Fun $a \to$ Fun $b \to \boxed{a+b}$

$$\text{Fun } x \to \text{Fun } y \to \underline{\boxed{x}+\boxed{y}} \cong \text{Fun } a \to \text{Fun } b \to \underline{\boxed{b}+\boxed{a}}$$

$$\text{Fun } a \to \text{Fun } b \to \underline{a+b}$$

α- Equivalence

Δ

$$\boxed{a} \cong \boxed{b}$$

Let $a = 3$ In

Let $b = 5$ In

$\boxed{}$

3

5

λ-Calculus

α-renaming

$$\lambda x.e \stackrel{\alpha}{\cong} \lambda y.\, e[y/x]$$

for all x whose binder is the argument

$\text{Fun } a \to \text{Fun } a \to \boxed{a} + \boxed{a}$

(1) (2)

$a \to x$

$\overset{\alpha}{\equiv} \quad \text{Fun } x \to \text{Fun } a \to a + a$

$\text{Fun } a \to \text{Fun } a \to a + a$

$a \to x$

$\overset{\alpha}{\equiv} \quad \text{Fun } a \to \text{Fun } \boxed{x} \to \boxed{x} + \boxed{x}$

$\boxed{\text{add}} \cong \boxed{\text{Fun } x \to \text{add } x}$

$\text{Let add} = \text{Fun } x \to \text{Fun } y \to x + y \text{ In } \boxed{}$

$\text{Let add} = 3 \text{ In } \boxed{}$

$\lambda x.e \overset{\eta}{\equiv} \lambda z.(\lambda x.e)\, z$ eta-equivalence

$$\text{Fun } x \to e \overset{\eta}{\equiv} \text{Fun } z \to (\text{Fun } x \to e)\ z$$

$$\cong$$

$$\boxed{\text{Let Rec } f\ x = f\ (x+1) \text{ In } f\ 0} \equiv e$$

$$\cong$$

$$e' \equiv \boxed{(0, 0)}$$

App

Int Int

0 0

Definition of diverging or invalid programs

For e_1 and e_2 if

$\nexists\, v_1$ s.t. $e_1 \Rightarrow v_1$ And

$\nexists\, v_2$ s.t $e_2 \Rightarrow v_2$ Then

$e_1 \cong e_2$

$$\text{Let } x = \boxed{\Omega} \text{ In } \underline{0} \cong \underline{0}$$

$$\frac{\boxed{e_1} \Rightarrow v_1 \qquad e_2[v_1/x] \Rightarrow v_2}{\text{Let } x = e_1 \text{ In } e_2 \Rightarrow v_2}$$

$$\frac{\cancel{\nexists v \text{ s.t. } e_1 \Rightarrow v} \qquad x \notin FV(e_2) \qquad e_2 \Rightarrow v_2}{\text{Let } x = e_1 \text{ In } e_2 \Rightarrow v_2}$$

(Hypothetical)

For any $\boxed{e}$, If True Then $\boxed{e}$ Else $\boxed{\Omega}$ $\cong$ $\boxed{e}$

Meta Variable

True

$$[\text{If-True}]\ \frac{e_1 \Rightarrow \text{True} \qquad e_2 \Rightarrow v_2}{\text{If } e_1 \text{ Then } e_2 \text{ Else } e_3 \Rightarrow v_2}$$

Dead Code Elimination

C

```
#define MACOS 1
if (MACOS) {
        do_mac_thing();
} else if (WIN) {
        do_win_thing();
}
```

```
if ( 1 ) {
    do_mac_thing();
} else if ( 0 ) {
    do_win_thing();
}
```

For any e, $\text{Let } x = e \text{ In } x + x \overset{\sim}{=} e + e$

True

β-reduction

(Grammar of Context)

$C ::= [\,]$ Hole

$\mid C\ e$

$\mid e\ C$

$\mid C + e \mid e + C$

$\mid \text{Let } x = C \text{ In } e$

$\mid \text{Let } x = e \text{ In } C$

$\mid \ldots$

(Syntax of F6)

$e ::=$

$\mid e\ e$

$\mid e + e$

$\mid \text{Let } x = e \text{ In } e$

$\mid \ldots$

$C[e]$

$C' \overset{\text{def}}{=} \text{let } x = [\,] \text{ In } x + x$

$C'[3] = \text{Let } x = 3 \text{ In } x + x$

$C'[y] = \text{Let } x = y \text{ In } x + x$

Formal Def: Op Equiv.

We say that $e_1 \cong e_2$ iff

$\forall C.$ $C[e_1]$ and $C[e_2]$ are closed exprs

if $\exists v_1$ $C[e_1] \Rightarrow v_1$. then

$\exists v_2$ $C[e_2] \Rightarrow v_2$ and

$\boxed{v_1 \cong v_2}$

And

if $\exists v_2$ $C[e_2] \Rightarrow v_2$ then

$\exists v_1$ $C[e_1] \Rightarrow v_1$ and

$\boxed{v_1 \cong v_2}$

$v ::= \boxed{\bar{v}} \mid \boxed{b} \mid \text{Fun } x \to e$

$\text{Fun } x \to e = \text{Fun } x' \to e'$

$x' = x$

$e' = e$