An example of the operational semantics OS_{clo}

Prof. Kamin, CS 421, Spring 2009

The application rule is a confusing part of the closure-based operational semantics. In particular, it is not clear how the environment contained within a closure can be different from the environment of the application itself. That is, in this rule:

$$\frac{\eta, e_1 \Downarrow < \text{fun x->e}, \eta' > \quad \eta, e_2 \Downarrow v \quad \eta[x \rightarrow v], e \Downarrow v'}{\eta, e_1 e_2 \Downarrow v'}$$

how can η and η' be different?

In fact, these two environments can be completely different. η' comes from the place where the function was created, while η comes from the place where it is applied. We now give an example that illustrates this.

To give this example, it is easier if we use let expressions. So we add a new rule for those. Note that this rules gives identical results to what we would get if we translated the let expression to an application of a function.

$$\frac{\eta, e_2 \Downarrow v \quad \eta[x \rightarrow v], e_1 \Downarrow v'}{\eta, \text{ let x=} e_2 \text{ in } e_1 \Downarrow v'}$$

Consider this expression:

```
let g = let a=1
        in let b=2
           in fun z -> a + b*z
in let c=3
   in g c
```

We give the proof tree below, but the key is this: The closure assigned to g is <fun z->a+b*z, {a→1,b→2}>. When g is applied in expression g c, the environment in effect is {g→v_g, c→3},

where v_g is the closure named above. Thus, in the application rule, the second environment is "η", while the first is "η'". Thus, these environments are indeed completely different.

We now she the proof tree showing that the expression above evaluates to 7. It uses a number of abbreviations just to make the tree easier to draw:

E_1 = fun z -> a + b*z
E_2 = let a=1 in let b=2 in E_1
E_3 = let c=3 in g c

$\eta_0 = \{a \rightarrow 1\}$
$\eta_1 = \{a \rightarrow 1, b \rightarrow 2\}$
$v_g = \langle E_1, \eta_1 \rangle$
$\eta_2 = \{g \rightarrow v_g, c \rightarrow 3\}$

Thus, what we want to prove is $\emptyset$, let g=E_2 in $E_3 \Downarrow 7$.

$$\dfrac{\dfrac{\dfrac{}{\emptyset, 1 \Downarrow 1} \quad \dfrac{\dfrac{}{\eta_0, 2 \Downarrow 2} \quad \dfrac{}{\eta_1, E_1 \Downarrow v_g}}{\eta_0, \text{let b=2 in } E_1 \Downarrow v_g}}{\emptyset,\ E_2 \Downarrow v_g} \quad \dfrac{\dfrac{}{\{g \rightarrow v_g\}, 3 \Downarrow 3} \quad \dfrac{\textit{see below}}{\eta_2,\ g\ c \Downarrow 7}}{\{g \rightarrow v_g\},\ E_3 \Downarrow 7}}{\emptyset,\ \text{let g=} E_2 \text{ in } E_3 \Downarrow 7}$$

To complete the proof, we finish the tree at the upper right:

$$\dfrac{\dfrac{}{\eta_2, g \Downarrow v_g} \quad \dfrac{}{\eta_2, c \Downarrow 3} \quad \dfrac{\textit{(left to reader)}}{\eta_1[z \rightarrow 3],\ a+b*z \Downarrow 7}}{\eta_2, g\ c \Downarrow 7}$$

Again, in the application rule (the last line above), the environment of the application is η_2, which gives bindings for g and c, while the environment of the closure – which is used to evaluate the body of the function – is η_1, which has bindings for a and b.