

Solutions for Exercise Sheet 7

In this exercise, we want you to evaluate terms using binding environments, each environment being written as a linked list frames, using this notation:

$$E = F_0 \leftarrow F_1 \leftarrow \dots$$

Each frame must be a set of variable bindings such as $\{x_1 \mapsto V_1, \dots, x_n \mapsto V_n\}$.

Lambda abstraction terms evaluate to closures, written in the form $\text{cls}(\lambda x. M, E)$ where E is the environment that is in effect for the lambda abstraction.

Exercise 1: Environment frames

a. Evaluate the expression

$$((\lambda x. x + y) 100)$$

starting from an global environment consisting of just one frame, $F_0 = \{y \mapsto 4\}$.

$$\frac{\frac{((\lambda x. x + y), F_0) \Downarrow \text{cls}(\lambda x. x + y, F_0) \quad (100, F_0) \Downarrow 100}{((\lambda x. x + y) 100, F_0) \Downarrow 104} \quad \frac{(x, F_0 \leftarrow F_1) \Downarrow 100 \quad (y, F_0 \leftarrow F_1) \Downarrow 4 \quad (100 + 4, F_0 \leftarrow F_1) \Downarrow 104}{(x + y, F_0 \leftarrow F_1) \Downarrow 104}}{(x + y, F_0 \leftarrow F_1) \Downarrow 104}}{(x + y, F_0 \leftarrow F_1) \Downarrow 104}$$

where

$$F_1 = \{x \mapsto 100\}$$

b. Evaluate the expression

$$(\lambda f. \lambda y. f y) (\lambda x. x + y) 100$$

starting from the same environment with F_0 .

Here, to save space, we omit the evaluation of constants, and give a name to whole environments rather than individual frames:

$$\frac{\frac{(\lambda f. \lambda y. f y, E_0) \Downarrow \text{cls}(\lambda f. \lambda y. f y, E_0) \quad (\lambda x. x + y, E_0) \Downarrow \text{cls}(\lambda x. x + y, E_0) \quad (\lambda y. f y, E_1) \Downarrow \text{cls}(\lambda y. f y, E_1) \quad \dots}{((\lambda f. \lambda y. f y) (\lambda x. x + y), E_0) \Downarrow \text{cls}(\lambda y. f y, E_1)} \quad \frac{\dots}{(f y, E_2) \Downarrow 104}}{((\lambda f. \lambda y. f y) (\lambda x. x + y) 100, E_0) \Downarrow 104}$$

where the part of the evaluation indicated by “...” is continued here:

$$\frac{(f, E_2) \Downarrow \text{cls}(\lambda x. x + y, E_0) \quad (y, E_2) \Downarrow 100 \quad \frac{(x, E_3) \Downarrow 100 \quad (y, E_3) \Downarrow 4 \quad (100 + 4, E_3) \Downarrow 104}{(x + y, E_3) \Downarrow 104}}{(f y, E_2) \Downarrow 104}$$

where

$$\begin{aligned} E_0 &= \{y \mapsto 4\} \\ E_1 &= E_0 \leftarrow \{f \mapsto \text{cls}(\lambda x. x + y, E_0)\} \\ E_2 &= E_1 \leftarrow \{y \mapsto 100\} \\ E_3 &= E_0 \leftarrow \{x \mapsto 100\} \end{aligned}$$

c. Draw a diagram in the style of Abelson and Sussman, showing all the environment frames and static links. Label the frames with the procedures whose activations (calls) created them.

Exercise 2: Environment frames, again

Recall the times function:

$$\begin{aligned} \text{times} = & (\lambda n. \lambda f. \text{if } (n = 0) (\lambda x. x) \\ & (\lambda x. \text{times } (n - 1) f (f x))) \end{aligned}$$

Assume that it is defined at the top level and that the global frame is

$$F_0 = \{ \text{times} \mapsto \text{cls}(\lambda n. \lambda f. \dots, F_0), x \mapsto 20 \}$$

Evaluate the term

$$\begin{aligned} & \text{let } \text{inc} = \lambda z. z + 1 \\ & \text{in } \text{times } 3 \text{ inc } x \end{aligned}$$

using the environment model. Show all the environments and frames that arise during the evaluation.

Exercise 3: SECD Machine

a. Show the evaluation of the expression $((\lambda x. x) 2)$ using the SECD machine. The initial configuration is:

$$(\top, \{ \}, [(\lambda x. x) 2, \top])$$

At each step specify the transition rule being employed.

	<u>stack</u>	<u>environment</u>	<u>control</u>	<u>dump</u>	<u>rule used</u>
		{ }	$((\lambda x. x) 2)$	\top	
1.		{ }	$(\lambda x. x), 2, \text{app}$	\top	APPLICATION
2.	$\text{cls}(\lambda x. x, \{ \})$	{ }	$2, \text{app}$	\top	LAMBDA
3.	$\text{cls}(\lambda x. x, \{ \}), 2$	{ }	app	\top	PUSH
4.		$\{x \mapsto 2\}$	x, rtn	$\{\top\}$	APPLY
5.	2	$\{x \mapsto 2\}$	rtn	$\{\top\}$	LOOKUP
6.	2	{ }		\top	RETURN

b. Show that there is a multiple step transition sequence of the following form:

$$\text{DBL-APPLICATION } (S, E, [(M N_1 N_2) | \overline{C}, D) \longrightarrow^* (S, E, [M, N_1, \text{app}, N_2, \text{app} | \overline{C}, D)$$

Recall that $(M N_1 N_2)$ is to be parsed as $((M N_1) N_2)$ by our bracketing convention. So, the application of $(M N_1)$ to N_2 is at the outer level and gets processed first.

$$\begin{aligned} & (S, E, [(M N_1 N_2) | \overline{C}, D) \\ & \longrightarrow (S, E, [(M N_1), N_2, \text{app} | \overline{C}, D) \\ & \longrightarrow (S, E, [M, N_1, \text{app}, N_2, \text{app} | \overline{C}, D) \end{aligned}$$

Both the transitions are instances of the APPLICATION rule.

c. Shown below are the first few steps of the evaluation for the expression $((\lambda f. \lambda y. f y) (\lambda x. + x y) 1)$. Mention the transition rules used for each step and verify that the rules have been applied correctly.

See Handout 6, paragraph 8, for the rules used.

d. Continue the evaluation of part (c).

	<u>stack</u>	<u>environment</u>	<u>control</u>	<u>dump</u>	<u>rule used</u>
.		E_3	$f y, \text{rtn}$	$\overline{E_1}$	
10.		E_3	$f, y, \text{app}, \text{rtn}$	$\overline{E_1}$	APPLICATION
11.	$\text{cls}(\lambda x. + x y, E_1)$	E_3	$y, \text{app}, \text{rtn}$	$\overline{E_1}$	LOOKUP
12.	$\text{cls}(\lambda x. + x y, E_1), 1$	E_3	app, rtn	$\overline{E_1}$	LOOKUP
13.		E_4	$+ x y, \text{rtn}, \text{rtn}$	$\overline{E_1}, \overline{E_3}$	APPLY
	\vdots				
?.	5	E_1		\top	

where $E_4 = \{x \mapsto 1, y \mapsto 4, z \mapsto 5\}$. In the elided steps, the computation of $+ x y$ now puts the value 5 on the stack and the two `ret` instructions restore the environment to E_1 . This leads to the final configuration.

Exercise 4: [Optional question] SECD transition rules

Recall that the expression form **let** $x = M$ **in** N can be desugared into lambda calculus. Define a transition rule for the SECD machine that allows it to directly reduce **let** expressions without translating it into lambda calculus.

The **let** expression is desugared into lambda calculus as $(\lambda x. N) M$. If this expression is encountered by the SECD machine, it immediately translates it to the sequence $(\lambda x. M), N, \text{app}$. So, one possibility is to use the transition rule:

$$(S, E, [\mathbf{let} \ x = M \ \mathbf{in} \ N \mid \overline{C}, D) \longrightarrow (S, E, [(\lambda x. N), M, \text{app} \mid \overline{C}, D)$$

You might also notice that there is enough detail available in the first expression in the control to carry out a further step. If you incorporate it into the transition rule for **let**, you can use the rule:

$$(S, E, [\mathbf{let} \ x = M \ \mathbf{in} \ N \mid \overline{C}, D) \longrightarrow ([S \mid \mathbf{cls}(\lambda x. N, E), E, [M, \text{app} \mid \overline{C}, D)$$