

CMSC330 Practice Problems 9 Solutions

1. Operational semantics

Use operational semantics to determine the values of the following OCaml codes:

a. 1

$$\frac{}{\bullet; 1 \Rightarrow 1}$$

b. 3 + 7

$$\frac{\bullet; 3 \Rightarrow 3 \quad \bullet; 7 \Rightarrow 7}{\bullet; 3 + 7 \Rightarrow 10}$$

c. 1 + (2 + 3)

$$\frac{\frac{\bullet; 2 \Rightarrow 2 \quad \bullet; 3 \Rightarrow 3}{\bullet; (2 + 3) \Rightarrow 5}}{\bullet; 1 + (2 + 3) \Rightarrow 6}$$

d. (fun x → 4) 5

$$\begin{array}{l} \bullet; (\text{fun } x \rightarrow 4) \Rightarrow (\bullet, \lambda x.4) \quad // \text{ 1) evaluate function to produce a closure} \\ \bullet; 5 \Rightarrow 5 \quad // \text{ 2) evaluate the argument} \\ \frac{x:5; 4 \Rightarrow 4}{\bullet; (\text{fun } x \rightarrow 4) 5 \Rightarrow 4} \quad // \text{ 3) evaluate body of closure, after extending} \\ \quad \quad \quad // \text{ environment w/ binding for parameter } x \end{array}$$

e. (fun x → x + 6) 7

$$\begin{array}{l} \frac{x:7; x \Rightarrow 7 \quad x:7; 6 \Rightarrow 6}{\bullet; (\text{fun } x \rightarrow x + 6) 7 \Rightarrow 13} \quad // \text{ 4) evaluate } x \text{ \& } 6 \text{ in extended environment} \\ \bullet; (\text{fun } x \rightarrow x + 6) \Rightarrow (\bullet, \lambda x.x + 6) \quad // \text{ 1) evaluate function to produce a closure} \\ \bullet; 7 \Rightarrow 7 \quad // \text{ 2) evaluate the argument} \\ \quad \quad \quad // \text{ 3) evaluate body of closure, after extending} \\ \quad \quad \quad // \text{ environment w/ binding for parameter } x \end{array}$$

f. $(\text{fun } x \rightarrow (\text{fun } y \rightarrow y + x)) 8 9$

Note expression $a b c$ is equivalent to $((a b) c)$, so we first evaluate $(a b)$ to find the value of the function, then apply the result to c .

$x:8, y:9; y \Rightarrow 9$	// 7) evaluate y & x
$x:8, y:9; x \Rightarrow 8$	// in extended env
$\bullet; (\text{fun } x \rightarrow (\text{fun } y \rightarrow y + x)) \Rightarrow (\bullet, \lambda x. (\text{fun } y \rightarrow y + x))$	// 2) eval func
$\bullet; 8 \Rightarrow 8$	// 3) eval arg
	// 4) eval body after
$x:8; (\text{fun } y \rightarrow y + x) \Rightarrow (x:8, \lambda y. (y + x))$	// extending env for x
$\bullet; (\text{fun } x \rightarrow (\text{fun } y \rightarrow y + x)) 8 \Rightarrow (x:8, \lambda y. (y + x))$	// 1) eval func
$\bullet; 9 \Rightarrow 9$	// 5) eval arg
	// 6) eval body after
$x:8, y:9; (y + x) \Rightarrow 17$	// extending env for y
$\bullet; (\text{fun } x \rightarrow (\text{fun } y \rightarrow y + x)) 8 9 \Rightarrow 17$	

g. let $x = 5$ in let $y = 7$ in $x+y$

	$\bullet, x:5, y:7; x \Rightarrow 5$	$\bullet, x:5, y:7; y \Rightarrow 7$
	$\bullet, x:5; 7 \Rightarrow 7$	$\bullet, x:5, y:7; x+y \Rightarrow 12$
$\bullet; 5 \Rightarrow 5$	$\bullet, x:5; \text{let } y = 7 \text{ in } x+y \Rightarrow 12$	
$\bullet; \text{let } x = 5 \text{ in let } y = 7 \text{ in } x+y \Rightarrow 12$		

h. let $x = \text{let } x = 5 \text{ in } x+2$ in $x+4$

	$\bullet, x:5; x \Rightarrow 5$	$\bullet, x:5; 2 \Rightarrow 2$
	$\bullet, 5 \Rightarrow 5$	$\bullet, x:5; x+2 \Rightarrow 7$
	$\bullet; \text{let } x = 5 \text{ in } x+2 \Rightarrow 7$	$\bullet, x:7; x \Rightarrow 7$
	$\bullet; \text{let } x = \text{let } x = 5 \text{ in } x+2 \text{ in } x+4 \Rightarrow 11$	$\bullet, x:7; 4 \Rightarrow 4$

i. let $f = \text{fun } x \rightarrow x+5$ in $f 7$

	$\bullet, x:7; x \Rightarrow 7$
	$\bullet, x:7; 5 \Rightarrow 5$
$\bullet; f: (\bullet, \lambda x. x+5); f \Rightarrow (\bullet, \lambda x. x+5)$	
$\bullet; f: (\bullet, \lambda x. x+5); 7 \Rightarrow 7$	
	$\bullet, x:7; x+5 \Rightarrow 12$
$\bullet; \text{fun } x \rightarrow x+5 \Rightarrow (\bullet, \lambda x. x+5)$	$\bullet; f: (\bullet, \lambda x. x+5); f 7 \Rightarrow 12$
$\text{let } f = \text{fun } x \rightarrow x+5 \text{ in } f 7 \Rightarrow 12$	

j. let $y = 5$ in let $f = \text{fun } x \rightarrow x+y$ in let $y = 6$ in $f\ 7$ (for static scoping)

$y:5, x:7; x \Rightarrow 7$
 $y:5, x:7; y \Rightarrow 5$
 $y:5, f:(y:5, \lambda x.x+y), y:6; f \Rightarrow (y:5, \lambda x.x+y)$
 $y:5, f:(y:5, \lambda x.x+y), y:6; 7 \Rightarrow 7$
 $y:5, x:7; x+y \Rightarrow 12$
 $y:5, f:(y:5, \lambda x.x+y); 6 \Rightarrow 6$ $y:5, f:(y:5, \lambda x.x+y), y:6; f\ 7 \Rightarrow 12$
 $y:5; \text{fun } x \rightarrow x+y \Rightarrow (y:5, \lambda x.x+y)$ $y:5, f:(y:5, \lambda x.x+y); \text{let } y = 6 \text{ in } f\ 7 \Rightarrow 12$
 $\bullet, 5 \Rightarrow 5$ $y:5; \text{let } f = \text{fun } x \rightarrow x+y \text{ in let } y = 6 \text{ in } f\ 7 \Rightarrow 12$
 $\bullet; \text{let } y = 5 \text{ in let } f = \text{fun } x \rightarrow x+y \text{ in let } y = 6 \text{ in } f\ 7 \Rightarrow 12$

k. let $y = 5$ in let $f = \text{fun } x \rightarrow x+y$ in let $y = 6$ in $f\ 7$ (for dynamic scoping)

$y:5, f:(y:5, \lambda x.x+y), y:6, x:7; x \Rightarrow 7$
 $y:5, f:(y:5, \lambda x.x+y), y:6, x:7; y \Rightarrow 6$
 $y:5, f:(y:5, \lambda x.x+y), y:6; f \Rightarrow (y:5, \lambda x.x+y)$
 $y:5, f:(y:5, \lambda x.x+y), y:6; 7 \Rightarrow 7$
 $y:5, f:(y:5, \lambda x.x+y), y:6, x:7; x+y \Rightarrow 13$
 $y:5, f:(y:5, \lambda x.x+y); 6 \Rightarrow 6$ $y:5, f:(y:5, \lambda x.x+y), y:6; f\ 7 \Rightarrow 13$
 $y:5; \text{fun } x \rightarrow x+y \Rightarrow (y:5, \lambda x.x+y)$ $y:5, f:(y:5, \lambda x.x+y); \text{let } y = 6 \text{ in } f\ 7 \Rightarrow 13$
 $\bullet, 5 \Rightarrow 5$ $y:5; \text{let } f = \text{fun } x \rightarrow x+y \text{ in let } y = 6 \text{ in } f\ 7 \Rightarrow 13$
 $\bullet; \text{let } y = 5 \text{ in let } f = \text{fun } x \rightarrow x+y \text{ in let } y = 6 \text{ in } f\ 7 \Rightarrow 13$