CMSC330 Spring 2014 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

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

   c. 1 + (2 + 3)

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

   d. (fun x → 4) 5

   • ; (fun x → 4) ⇒ (• , λx.4) // 1) evaluate function to produce a closure
   • ; 5 ⇒ 5 // 2) evaluate the argument
   // 3) evaluate body of closure, after extending
   x:5; 4 ⇒ 4 // environment w/ binding for parameter x)
   • ; (fun x → 4) 5 ⇒ 4

   e. (fun x → x + 6) 7

   x:7; x ⇒ 7 x:7; 6 ⇒ 6 // 4) evaluate x & 6 in extended environment

   • ; (fun x → x + 6) ⇒ (• , λx.x + 6) // 1) evaluate function to produce a closure
   • ; 7 ⇒ 7 // 2) evaluate the argument
   // 3) evaluate body of closure, after extending
   x:7; x + 6 ⇒ 13 // environment w/ binding for parameter x)
   • ; (fun x → x + 6) 7 ⇒ 13

f. (fun x → (fun y → 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 ⇒ 9 x:8, y:9 ; x ⇒ 8 // 7) evaluate y & x // in extended env

• ; (fun x → (fun y → y + x)) ⇒ (• , λx.(fun y → y + x)) // 2) eval func
• ; 8 ⇒ 8 // 3) eval arg
// 4) eval body after
x:8 ; (fun y → y + x) ⇒ (x:8, λy.(y + x)) // extending env for x
• ; (fun x → (fun y → y + x)) 8 ⇒ (x:8, λy.(y + x)) // 1) eval func
• ; 9 ⇒ 9 // 5) eval arg
// 6) eval body after
x:8, y:9 ; (y + x) ⇒ 17 // extending env for y
• ; (fun x → (fun y → y + x)) 8 9 ⇒ 17

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

•,x:5, y:7; x ⇒ 5 •,x:5, y:7; y ⇒ 7
•,x:5; 7 ⇒ 7 •,x:5, y:7; x+y ⇒ 12
•; 5 ⇒ 5 •,x:5; let y = 7 in x+y ⇒ 12
•; let x = 5 in let y = 7 in x+y ⇒ 12

h. let x = let x = 5 in x+2 in x+4

•,x:5; x ⇒ 5 •,x:5; 2 ⇒ 2
•,5 ⇒ 5 •,x:5; x+2 ⇒ 7 •,x:7; x ⇒ 7 •,x:7; 4 ⇒ 4
•; let x = 5 in x+2 ⇒ 7 •,x:7; x+4 ⇒ 11
•; let x = let x = 5 in x+2 in x+4 ⇒ 11

i. let f = fun x → x+5 in f 7

•, x:7; x ⇒ 7 •, x:7; 5 ⇒ 5
•,f: (•, λx.x+5) ; f ⇒ (•, λx.x+5)
•,f: (•, λx.x+5); 7 ⇒ 7
•, x:7; x+5 ⇒ 12
•; fun x → x+5 ⇒ (•, λx.x+5) •,f: (•, λx.x+5); f 7 ⇒ 12
let f = fun x → x+5 in f 7 ⇒ 12

j. let y = 5 in let f = fun x → x+y in let y = 6 in f 7 (for static scoping)

y:5, x:7; x ⇒ 7
y:5, x:7; y ⇒ 5
y:5, f:(y:5, λx.x+y), y:6; f ⇒ (y:5, λx.x+y)
y:5, f:(y:5, λx.x+y), y:6; 7 ⇒ 7
y:5, x:7; x+y ⇒ 12
y:5, f:(y:5, λx.x+y); 6 ⇒ 6 y:5, f:(y:5, λx.x+y), y:6; f 7 ⇒ 12
y:5; fun x → x+y ⇒ (y:5, λx.x+y) y:5, f:(y:5, λx.x+y); let y = 6 in f 7 ⇒ 12
•,5 ⇒ 5 y:5; let f = fun x → x+y in let y = 6 in f 7 ⇒ 12
•; let y = 5 in let f = fun x → x+y in let y = 6 in f 7 ⇒ 12

k. let y = 5 in let f = fun x → x+y in let y = 6 in f 7 (for dynamic scoping)

y:5, f:(y:5, λx.x+y), y:6, x:7; x ⇒ 7
y:5, f:(y:5, λx.x+y), y:6, x:7; y ⇒ 6
y:5, f:(y:5, λx.x+y), y:6; f ⇒ (y:5, λx.x+y)
y:5, f:(y:5, λx.x+y), y:6; 7 ⇒ 7
y:5, f:(y:5, λx.x+y), y:6, x:7; x+y ⇒ 13
y:5, f:(y:5, λx.x+y); 6 ⇒ 6 y:5, f:(y:5, λx.x+y), y:6; f 7 ⇒ 13
y:5; fun x → x+y ⇒ (y:5, λx.x+y) y:5, f:(y:5, λx.x+y); let y = 6 in f 7 ⇒ 13
•,5 ⇒ 5 y:5; let f = fun x → x+y in let y = 6 in f 7 ⇒ 13
•; let y = 5 in let f = fun x → x+y in let y = 6 in f 7 ⇒ 13

2. Programming languages

a. Describe the difference between HTML and XML.

HTML tags are predefined and presentation-oriented, whereas XML tags are user defined and are intended for describing data and metadata.

b. Describe the difference between query languages and programming languages.

Query languages are designed to make requests to a database or information system, whereas programming languages are designed to express computations that can be performed by a machine.

3. Markup languages
    a. Creating your own XML tags, write an XML document that organizes the following information: 1-hour test on Spanish Monday in Jiménez worth 15%. 1-hour test on Computers Tuesday in CSIC worth 10%. 30-minute test on Computers Friday in AVW worth 5%.

```
<testList>
    <test>
        <length>1 hour</length>
        <subject>Spanish</subject>
        <date>Monday</date>
        <location>Jiménez<location>
        <value>15%</value>
    </test>
    <test>
        <length>1 hour</length>
        <subject>Computers</subject>
        <date>Tuesday</date>
        <location>CSIC<location>
        <value>10%</value>
    </test>
    <test>
        <length>30 minute</length>
        <subject>Computers</subject>
        <date>Friday</date>
        <location>AVW<location>
        <value>5%</value>
    </test>
</testList>
```

OR

```
<testList>
    <test subject=”Spanish”>
        <length unit=”hour”>1</length>
        …
    </test>
    <test subject=”Computers” >
        <length unit=”hour”>1</length>
        …
    </test>
    <test subject=”Computers” >
        <length unit=”minute”>30</length>
        …
    </test>
</testList>
```