HW 2 – Evaluating the application of a function

CS 421 – Fall 2014
Revision 1.0

Assigned September 2, 2014
Due September 9, 2014, 23:59 pm
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Answer the problems below, then using your favorite tool(s), save your work as a PDF (either scanned if handwritten or converted from a program), and commit the PDF in your your repositiory in the directory `assignments/hw2`. Your file should be named `hw2-submission.pdf`. The command to commit this file is

```
svn commit -m "Turning in hw2." hw2-submission.pdf
```

3 Objectives and Background

The purpose of this HW is to test your understanding of

- the order of evaluation of expressions in OCaml,
- the scope of variables, and the state of environments used during evaluation, and
- how an application of a function is evaluated.

Another purpose of HWs is to provide you with experience answering non-programming written questions of the kind you may experience on the midterms and final.

4 Problems

1. (41 pts) Below is a fragment of OCaml code, with various program points indicated by numbers with comments.

```
let x = 6;;
let y = 2;;
let skew (x, y) = (2 * x) + (5 * y);;
(* 1 *)
let diff = fun y -> fun z -> y - z;;
(* 2 *)
let from_x = fun y -> diff x y;;
(* 3 *)
```

```
let z = from_x y;;
(* 4 *)
let seven = fun z ->  skew(z, z);;
(* 5 *)
let y = seven y;;
(* 6 *)
let g x = diff (skew(x,1)) (from_x x);;
(* 7 *)
g 4;;
(* 8 *)
```

You need to do:

1. (16 pts) For each of `(* 1 *)` through `(* 7 *)` give the environment in effect at that point. Parts `(* 1 *)` and `(* 7 *)` are worth 3 points each, and the intervening parts are worth 2 points each. You may name your intermediate environments, and use the names in subsequent closures and environment constructions, except that for `(* 7 *)`, where you must not use the + (update) operator, (but you may use environment names in closures) and all top-level bindings must be explicitly written out.
2. (8 pts) Show step by step how the expression `seven y` is evaluated.
3. (17 pts) Show step by step how the expression `g 4` is evaluated.

You may assume that the evaluation begins in an empty environment, and that the environment is cumulative thereafter. The program points are supposed to indicate points at which all complete preceding declarations have been fully evaluated.